

# An unstructured grid, three-dimensional model based on the shallow water equations

Vincenzo Casulli<sup>a,\*</sup> and Roy A. Walters<sup>b</sup>

<sup>a</sup> *Department of Civil and Environmental Engineering, University of Trento, 38050 Mesiano di Povo, Trento, Italy*

<sup>b</sup> *US Geological Survey, PO Box 25046, MS 413, Lakewood, CO 80225, USA*

## SUMMARY

A semi-implicit finite difference model based on the three-dimensional shallow water equations is modified to use unstructured grids. There are obvious advantages in using unstructured grids in problems with a complicated geometry. In this development, the concept of unstructured orthogonal grids is introduced and applied to this model. The governing differential equations are discretized by means of a semi-implicit algorithm that is robust, stable and very efficient. The resulting model is relatively simple, conserves mass, can fit complicated boundaries and yet is sufficiently flexible to permit local mesh refinements in areas of interest. Moreover, the simulation of the flooding and drying is included in a natural and straightforward manner. These features are illustrated by a test case for studies of convergence rates and by examples of flooding on a river plain and flow in a shallow estuary. Copyright © 2000 John Wiley & Sons, Ltd.

**KEY WORDS:** finite difference; finite volume; semi-implicit; shallow water; three-dimensional; unstructured grid

## 1. INTRODUCTION

A practical numerical method to solve the two-dimensional shallow water equations has been formulated using a semi-implicit time integration, a finite difference spatial discretization and an Eulerian–Lagrangian approximation to advection [1]. The resulting model has been shown to be numerically stable, very efficient and has been successfully used in a number of applications, e.g. [2,3]. Recently, the two-dimensional model has been extended to the three-dimensional shallow water equations [4,5]. In the three-dimensional model, the pressure gradient term in the momentum equations and the divergence term in the vertically integrated continuity equation are treated implicitly to remove stability constraints. Numerical experiments with the three-dimensional model have shown that the algorithm is also stable and

---

\* Correspondence to: Department of Civil and Environmental Engineering, University of Trento, 38050 Mesiano di Povo, Trento, Italy.

highly efficient. Moreover, when only one vertical layer is specified, this method reduces to the semi-implicit method for the two-dimensional vertically integrated shallow water equations. The resulting two- and three-dimensional models, however, have been restricted to using uniform rectangular grids in the horizontal plane. Our purpose here is to extend this model to use a more flexible unstructured grid that can be used to discretize arbitrary geometries.

Numerical methods using orthogonal curvilinear co-ordinates are relatively simple because no new terms are introduced by the curvilinear transformation, e.g. [6]. However, orthogonal structured grids are not sufficiently flexible to fit an arbitrary geometry. On the other hand, non-orthogonal curvilinear co-ordinates allow for higher flexibility but introduce new terms that often result in larger discretization errors and/or instability, see [7] and [8, pp. 200–221]. The importance of boundary fitting has also been investigated in terms of finite difference methods derived for unstructured triangular grids [9]. In this case, the two-dimensional shallow water equations have been discretized by explicit schemes. Thus, the time step was restricted to comply with the Courant–Friedrichs–Lewy (CFL) stability condition. As an alternative to finite difference methods, several finite element algorithms using triangular elements are currently being studied and applied [10,11]. The two-dimensional model described in [10] is very similar in concept to the development presented here, except that it uses a finite element rather than a finite difference spatial discretization. The elements used are the Raviart–Thomas elements of lowest order with the result that the models differ in the formulation of the pressure gradient terms.

In order to obtain an efficient finite difference model for the three-dimensional shallow water equations, an unstructured grid with special properties is introduced. The name *unstructured orthogonal grid* is used to denote this special grid. Then, a semi-implicit finite difference scheme is modified and adapted to this grid. The advection, Coriolis and horizontal friction terms are discretized by using an Eulerian–Lagrangian approach. The resulting algorithm conserves mass and is computationally competitive with the corresponding semi-implicit finite difference scheme for uniform rectangular grids. Moreover, this algorithm has more flexibility in fitting complicated boundaries. In the particular case that only one vertical layer is specified, this algorithm is consistent with the vertically integrated, two-dimensional shallow water equations in the same manner as the corresponding structured grid model [5]. Preliminary results that were reported during the development of the model are presented in [12]. The resulting scheme is suitable for the simulation of complex three-dimensional flows using a fine spatial resolution and relatively large time steps. The present formulation allows for the simulation of flooding and drying of low-lying areas in a straightforward manner.

## 2. GOVERNING EQUATIONS

The governing three-dimensional, primitive variable equations describing constant density, free surface flows can be derived from the Navier–Stokes equations after averaging over turbulent time scales and under the simplifying assumption that the pressure is hydrostatic [4]. The resulting equations, the three-dimensional version of the shallow water equations, have the following form:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} - fv = -g \frac{\partial \eta}{\partial x} + \nu^h \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{\partial}{\partial z} \left( \nu^v \frac{\partial u}{\partial z} \right), \quad (1)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + fu = -g \frac{\partial \eta}{\partial y} + \nu^h \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \frac{\partial}{\partial z} \left( \nu^v \frac{\partial v}{\partial z} \right), \quad (2)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \quad (3)$$

where  $u(x, y, z, t)$ ,  $v(x, y, z, t)$  and  $w(x, y, z, t)$  are the velocity components in the horizontal  $x$ -,  $y$ - and vertical  $z$ -directions respectively;  $t$  is the time;  $\eta(x, y, t)$  is the water surface elevation measured from the undisturbed water surface;  $f$  is the Coriolis parameter,  $g$  is the gravitational acceleration and  $\nu^h$  and  $\nu^v$  are the coefficients of horizontal and vertical eddy viscosity respectively. These coefficients can be derived from an appropriate turbulence closure model. Here, it will be assumed that  $\nu^h$  and  $\nu^v$  are prescribed non-negative functions of space and time, and no specific closure model is used.

Integrating the continuity equation over depth and using a kinematic condition at the free surface leads to the following free surface equation:

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x} \left[ \int_{-h}^{\eta} u \, dz \right] + \frac{\partial}{\partial y} \left[ \int_{-h}^{\eta} v \, dz \right] = 0, \quad (4)$$

where  $h(x, y)$  is the water depth measured from the undisturbed water surface so that  $H(x, y, t) = h(x, y) + \eta(x, y, t)$  is the total water depth.

The boundary conditions at the free surface are specified by the prescribed wind stresses as

$$\nu^v \frac{\partial u}{\partial z} = \gamma_T (u_a - u), \quad \nu^v \frac{\partial v}{\partial z} = \gamma_T (v_a - v), \quad \text{at } z = \eta, \quad (5)$$

where  $u_a$  and  $v_a$  are the wind velocity components in the  $x$ - and  $y$ -directions respectively, and  $\gamma_T$  is a non-negative wind stress coefficient that depends on the wind speed. At the sediment–water interface, the bottom friction is specified by

$$\nu^v \frac{\partial u}{\partial z} = \gamma_B u, \quad \nu^v \frac{\partial v}{\partial z} = \gamma_B v, \quad \text{at } z = -h, \quad (6)$$

where  $\gamma_B$  is a non-negative bottom friction coefficient. Typically,  $\gamma_B$  can be given by the Manning–Chezy formula, or by fitting it to a turbulent boundary layer.

### 3. ORTHOGONAL UNSTRUCTURED GRID

Before discretizing Equations (1)–(6), the horizontal  $(x, y)$  domain is covered by a grid consisting of a set of non-overlapping convex polygons, usually either triangles or quadrilaterals. Each side of a polygon is either a boundary line or a side of an adjacent polygon.

### Definition

A grid is said to be an *unstructured orthogonal grid* if within each polygon a point (hereafter called *center*) can be identified in such a way that the segment joining the centers of two adjacent polygons and the side shared by the two polygons, have a non-empty intersection and are *orthogonal* to each other (see Figure 1). A *structured orthogonal grid* has the same definition except that it applies to structured grids.

The center of a polygon does not necessarily coincide with its geometrical center. Examples of structured orthogonal grids are, of course, the rectangular finite difference grids, as well as a grid of uniform equilateral triangles. In these particular cases, the center of each polygon can be identified with its geometrical center. A non-uniform structured orthogonal grid can be obtained by projecting in the  $(x, y)$  domain a uniform grid derived from an appropriate orthogonal curvilinear transformation. In general, however, a structured orthogonal grid derived from curvilinear transformation is not sufficiently flexible to fit an arbitrary spatial domain.

An example of an unstructured orthogonal grid is the Delaunay triangulation of a set of points  $Q_i, i = 1, 2, \dots, N$ , provided that the triangulation includes only acute triangles. In this case, the centers of the triangles are identified by the vertices of the corresponding Dirichlet tessellation [13]. Similarly, the Voronoi regions of a Dirichlet tessellation determined by  $Q_i$  also form an orthogonal grid, with  $Q_i$  being the centers of the polygons. Using co-volumes defined by Voronoi–Delaunay triangulation, a similar orthogonality condition has been proposed for the solution of the incompressible Navier–Stokes equations [14]. With the definition given above, an irregular discretization with quadrilaterals can also be used to form an unstructured orthogonal grid.

Once the  $(x, y)$  domain has been covered with a structured or unstructured orthogonal grid, one has  $N_p$  polygons, each having an arbitrary number of sides  $S_i \geq 3, i = 1, 2, \dots, N_p$ . Let  $N_s$  be the total number of sides in the grid, and let  $\lambda_j, j = 1, 2, \dots, N_s$  be the length of each side. The sides of the  $i$ th polygon are identified by an index  $j(i, l), l = 1, 2, \dots, S_i$ , so that

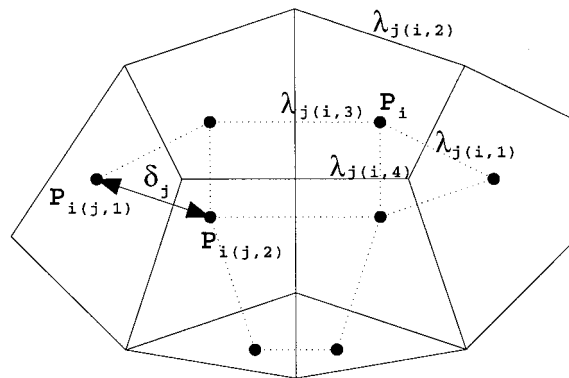


Figure 1. Orthogonal unstructured grid.

$1 \leq j(i, l) \leq N_s$ ,  $i = 1, 2, \dots, N_p$ . Similarly, the two polygons that share the  $j$ th side of the grid are identified by the indices  $i(j, 1)$  and  $i(j, 2)$  so that  $1 \leq i(j, 1) \leq N_p$  and  $1 \leq i(j, 2) \leq N_p$ ,  $j = 1, 2, \dots, N_s$ . The non-zero distance between the centers of two adjacent polygons that share the  $j$ th side is denoted by  $\delta_j$  (see Figure 1).

Along the vertical direction a simple finite difference discretization, not necessarily uniform, is adopted. By denoting a given level surface with  $z_{k+1/2}$ , the vertical discretization step is defined by  $\Delta z_k = z_{k+1/2} - z_{k-1/2}$ ,  $k = 1, 2, \dots, N_z$ .

The three-dimensional space discretization consists of prisms whose horizontal faces are the polygons of a given orthogonal grid and whose height are  $\Delta z_k$ . The discrete velocities and water surface elevation are defined at staggered locations as follows. The water surface elevation  $\eta_i^n$ , assumed to be constant within each polygon, is located at the center of the  $i$ th polygon; the velocity component normal to each face of a prism, assumed to be constant over the face, is defined at the point of intersection between the face and the segment joining the centers of the two prisms that share the face. Finally, the water depth  $h_j$  is specified, and assumed constant, on each side of a polygon.

#### 4. A SEMI-IMPLICIT NUMERICAL METHOD

A semi-implicit scheme is used in order to obtain an efficient numerical algorithm whose stability is independent of the free surface wave speed, wind stress, vertical viscosity and bottom friction. The water surface elevation in the horizontal momentum equations (1) and (2), and the velocity in the free surface equation (4), are discretized by the  $\theta$  method [5]. In addition, the wind stress term, the vertical friction term, and the bottom friction term are discretized implicitly for stability.

Of course, Equations (1)–(6) are invariant under solid rotation of the  $x$ - and  $y$ -axis on the horizontal plane. Using an unstructured orthogonal grid, a consistent semi-implicit finite difference discretization for the velocity component normal to each vertical face of a prism can be derived from Equations (1) and (2) and takes the following form:

$$u_{j,k}^{n+1} = Fu_{j,k}^n - g \frac{\Delta t}{\delta_j} [\theta(\eta_{i(j,2)}^{n+1} - \eta_{i(j,1)}^{n+1}) + (1 - \theta)(\eta_{i(j,2)}^n - \eta_{i(j,1)}^n)] \\ + \frac{\Delta t}{\Delta z_{j,k}^n} \left[ v_{j,k+1/2}^v \frac{u_{j,k+1}^{n+1} - u_{j,k}^{n+1}}{\Delta z_{j,k+1/2}^n} - v_{j,k-1/2}^v \frac{u_{j,k}^{n+1} - u_{j,k-1}^{n+1}}{\Delta z_{j,k-1/2}^n} \right], \quad k = m_j, m_j + 1, \dots, M_j^n, \quad (7)$$

where  $u_{j,k}^n$  denotes the horizontal velocity component normal to the  $j$ th side of the grid, at vertical level  $k$  and time step  $n$ . The positive direction for  $u_{j,k}^n$  has been chosen to be from  $i(j, 1)$  to  $i(j, 2)$ .  $F$  is an explicit finite difference operator, which accounts for the contributions from the discretization of the Coriolis, advection and horizontal friction terms. A particular form for  $F$  can be given in several ways, such as by using an Eulerian–Lagrangian scheme, e.g. [1,4,15]. When the Coriolis, advection and horizontal friction terms are neglected in the above scheme,  $F$  reduces to the identity operator, i.e.  $Fu_{j,k}^n = u_{j,k}^n$ . Including all the above terms, and by using an Eulerian–Lagrangian discretization, a form for  $F$  can be chosen as

$$Fu_{j,k}^n = \frac{[1 - \theta(1 - \theta)f^2\Delta t^2]u_{j,k}^* + f\Delta tv_{j,k}^*}{1 + \theta^2 f^2 \Delta t^2} + \Delta tv^h \Delta_h u_{j,k}^*, \quad (8)$$

where  $u_{j,k}^*$  denotes the horizontal velocity component normal to the  $j$ th side of the grid and  $v_{j,k}^*$  is the tangential velocity component in a right-hand co-ordinate system. Both components are interpolated at time  $t_n$  at the end of the Lagrangian trajectory based on the values at adjacent grid points. The Lagrangian trajectory is calculated by integrating the velocity backwards in time from node  $(j, k)$  at  $t^{n+1}$  to its location at time  $t^n$ .  $\Delta_h$  is the horizontal Laplacian discretization.

For stability, the implicitness factor  $\theta$  has to be chosen in the range  $1/2 \leq \theta \leq 1$  [5]. Moreover, the vertical space increment  $\Delta z_{j,k}^n$  is defined as the distance between two consecutive level surfaces, except near the bottom and near the free surface, where  $\Delta z_{j,k}^n$  is the distance between a level surface and the bottom or free surface respectively. Thus, in general,  $\Delta z_{j,k}^n$  depends on the spatial location, and near the free surface it also depends on the time step. The vertical space increment  $\Delta z_{j,k}^n$  is also allowed to vanish in order to account for wetting and drying. Of course, the momentum equation (7) is not defined at the grid points characterized by  $\Delta z_{j,k}^n = 0$ . Finally,  $m_j$  and  $M_j^n$ ,  $1 \leq m_j \leq M_j^n \leq N_z$ , denote the lower and upper limit for the  $k$  index representing the bottom and the top finite difference stencil respectively. As indicated,  $m$  and  $M$  depend on their spatial location and  $M$  may also change with the time level to account for the free surface dynamics. For notational simplicity, the subscript and the superscript to  $m_j$  and  $M_j^n$  will be omitted in the following development.

The values of  $u_{j,k}^{n+1}$  above the free surface and below the bottom in Equation (7) are eliminated by means of the boundary conditions (5) and (6), which yield the following difference formulae:

$$v_{j,M+1/2}^v \frac{u_{j,M+1}^{n+1} - u_{j,M}^{n+1}}{\Delta z_{j,M+1/2}^n} = \gamma_T^{n+1} (u_a^{n+1} - u_{j,M}^{n+1}) \quad (9)$$

and

$$v_{j,m-1/2}^v \frac{u_{j,m}^{n+1} - u_{j,m-1}^{n+1}}{\Delta z_{j,m-1/2}^n} = \gamma_B^{n+1} u_{j,m}^{n+1}. \quad (10)$$

A semi-implicit finite volume discretization for the free surface equation (4) at the center of each polygon is taken to be

$$P_i \eta_i^{n+1} = P_i \eta_i - \theta \Delta t \sum_{l=1}^{S_i} \left[ s_{i,l} \lambda_{j(i,l)} \sum_{k=m}^M \Delta z_{j(i,l),k}^n u_{j(i,l),k}^{n+1} \right] - (1 - \theta) \Delta t \sum_{l=1}^{S_i} \left[ s_{i,l} \lambda_{j(i,l)} \sum_{k=m}^M \Delta z_{j(i,l),k}^n u_{j(i,l),k}^n \right], \quad (11)$$

where  $P_i$  denotes the area of the  $i$ th polygon;  $s_{i,l}$  is a sign function associated with the orientation of the normal velocity defined on the  $l$ th side of the polygon  $i$ . Specifically,  $s_{i,l} = 1$  if a positive velocity on the  $l$ th side corresponds to outflow,  $s_{i,l} = -1$  if a positive velocity on the  $l$ th side corresponds to inflow to the  $i$ th water column. Thus,  $s_{i,l}$  can be written as

$$s_{i,l} = \frac{i[j(i,l), 2] - 2i + i[j(i,l), 1]}{i[j(i,l), 2] - i[j(i,l), 1]}. \quad (12)$$

Equations (7) and (11) constitute a linear system of at most  $N_z N_s + N_p$  equations. This system has to be solved at each time step in order to calculate the new field variables  $u_{j,k}^{n+1}$  and  $\eta_i^{n+1}$  throughout the flow domain.

## 5. SOLUTION ALGORITHM

Since a linear system of  $N_z N_s + N_p$  equations can be quite large, even for modest values of  $N_z$ ,  $N_s$  and  $N_p$ , the system of Equations (7) and (11) is first decomposed into a set of  $N_s$  independent tridiagonal systems of  $N_z$  equations and one linear system of  $N_p$  equations. Specifically, upon multiplication by  $\Delta z_{j,k}^n$ , and after including the boundary conditions (9) and (10), Equations (7) and (11) are first written in matrix form as

$$\mathbf{A}_j^n \mathbf{U}_j^{n+1} = \mathbf{G}_j^n - \theta g \frac{\Delta t}{\delta_j} [\eta_{i(j,2)}^{n+1} - \eta_{i(j,1)}^{n+1}] \Delta \mathbf{Z}_j^n, \quad (13)$$

$$P_i \eta_i^{n+1} = P_i \eta_i^n - \theta \Delta t \sum_{l=1}^{S_i} s_{i,l} \lambda_{j(i,l)} [\Delta \mathbf{Z}_{j(i,l)}^n]^\top \mathbf{U}_{j(i,l)}^{n+1} - (1 - \theta) \Delta t \sum_{l=1}^{S_i} s_{i,l} \lambda_{j(i,l)} [\Delta \mathbf{Z}_{j(i,l)}^n]^\top \mathbf{U}_{j(i,l)}^n, \quad (14)$$

where  $\mathbf{U}$ ,  $\Delta \mathbf{Z}$ ,  $\mathbf{G}$  and  $\mathbf{A}$  are defined as follows:

$$\mathbf{U}_j^{n+1} = \begin{bmatrix} u_{j,M}^{n+1} \\ u_{j,M-1}^{n+1} \\ \vdots \\ u_{j,m}^{n+1} \end{bmatrix}, \quad \Delta \mathbf{Z}_j^n = \begin{bmatrix} \Delta z_{j,M}^n \\ \Delta z_{j,M-1}^n \\ \vdots \\ \Delta z_{j,m}^n \end{bmatrix},$$

$$\mathbf{G}_j^n = \begin{bmatrix} \Delta z_{j,M}^n \left[ F u_{j,M}^n - g \frac{\Delta t}{\delta_j} (1 - \theta) (\eta_{i(j,2)}^n - \eta_{i(j,1)}^n) \right] + \Delta t \gamma_T^{n+1} u_a^{n+1} \\ \Delta z_{j,M-1}^n \left[ F u_{j,M-1}^n - g \frac{\Delta t}{\delta_j} (1 - \theta) (\eta_{i(j,2)}^n - \eta_{i(j,1)}^n) \right] \\ \vdots \\ \Delta z_{j,m}^n \left[ F u_{j,m}^n - g \frac{\Delta t}{\delta_j} (1 - \theta) (\eta_{i(j,2)}^n - \eta_{i(j,1)}^n) \right] \end{bmatrix},$$

$$\mathbf{A}_j^n = \begin{bmatrix} \Delta z_{j,M}^n + a_{j,M-1/2}^n + \gamma_T^{n+1} \Delta t & -a_{j,M-1/2}^n & & 0 \\ -a_{j,M-1/2}^n & \Delta z_{j,M-1}^n + a_{j,M-1/2}^n + a_{j,M-3/2}^n & -a_{j,M-3/2}^n & \\ & \ddots & \ddots & \\ 0 & & -a_{j,m+1/2}^n & \Delta z_{j,m}^n + a_{j,m+1/2}^n + \gamma_B^{n+1} \Delta t \end{bmatrix},$$

with  $a_{j,k \pm 1/2}^n = v_{j,k \pm 1/2}^v \Delta t / \Delta z_{j,k \pm 1/2}^n$ .

Formal substitution of the expressions for  $\mathbf{U}_j^{n+1}$  from Equation (13) into (14) yields a discrete wave equation for  $\eta^{n+1}$ , which is given by

$$\begin{aligned} P_i \eta_i^{n+1} - g \Delta t^2 \theta^2 \sum_{l=1}^{S_i} \frac{s_{i,l} \lambda_{j(i,l)}}{\delta_{j(i,l)}} [(\Delta \mathbf{Z})^\top \mathbf{A}^{-1} \Delta \mathbf{Z}]_{j(i,l)}^n (\eta_{j(i,l),2}^{n+1} - \eta_{j(i,l),1}^{n+1}) \\ = P_i \eta_i^n - (1 - \theta) \Delta t \sum_{l=1}^{S_i} s_{i,l} \lambda_{j(i,l)} [(\Delta \mathbf{Z})^\top \mathbf{U}]_{j(i,l)}^n - \theta \Delta t \sum_{l=1}^{S_i} s_{i,l} \lambda_{j(i,l)} [(\Delta \mathbf{Z})^\top \mathbf{A}^{-1} \mathbf{G}]_{j(i,l)}^n. \end{aligned} \quad (15)$$

Since matrix  $\mathbf{A}_j^n$  is positive definite, its inverse is also positive definite and therefore  $[(\Delta \mathbf{Z})^\top \mathbf{A}^{-1} \Delta \mathbf{Z}]_j^n$  is a non-negative number. Hence, Equation (15) constitutes a linear, sparse system of  $N_p$  equations for  $\eta_i^{n+1}$ . This system is strongly diagonally dominant, symmetric and positive definite. Thus, it has a unique solution that can be efficiently determined by a preconditioned conjugate gradient method, e.g. [17].

Once the new free surface location has been calculated, Equation (13) constitutes a linear, tridiagonal system for  $\mathbf{U}_j^{n+1}$ . Each of these tridiagonal systems is independent of the others and is symmetric and positive definite. Thus, they can be conveniently solved by a direct method to determine  $\mathbf{U}_j^{n+1}$  throughout the computational domain.

Finally, a finite volume discretization of the continuity equation (3) yields the vertical component of the velocity at the new time level. Specifically, by setting  $w_{i,m-1/2}^{n+1} = 0$ , the vertical velocity component on each water column is determined recursively by

$$w_{i,k+1/2}^{n+1} = w_{i,k-1/2}^{n+1} - \frac{1}{P_i} \sum_{l=1}^{S_i} s_{i,l} \lambda_{j(i,l)} \Delta z_{j(i,l)}^n u_{j(i,l),k}^{n+1}, \quad k = m, m+1, \dots, M-1. \quad (16)$$

The numerical algorithm presented above includes the simulation of flooding and drying. At each time step the new water depths  $H_j^{n+1}$  at the polygon's sides are defined as

$$H_j^{n+1} = \max[0, h_j + \eta_{i(j,1)}^{n+1}, h_j + \eta_{i(j,2)}^{n+1}]. \quad (17)$$

The vertical grid spacings  $\Delta z_j^{n+1}$  are updated accordingly. Thus, an occurrence of a zero value for the total depth  $H_j^{n+1}$  implies that all the vertical faces separating prisms between the water column  $i(j, 1)$  and  $i(j, 2)$  are dry and may become wet at a later time when  $H_j^{n+1}$  becomes positive. The height of a dry face and the corresponding normal velocity are taken to be zero.



An interesting aspect of the present algorithm is given by the fact that the size and the structure of the linear system given by the discrete wave equation (15) is independent from the vertical resolution. The vertical resolution affects the assembly of Equation (15) and the number of equations for normal velocity. Moreover, since a major part of the computational effort is required to determine the free surface elevation from Equation (15), an arbitrarily fine vertical resolution can be used with an acceptable increase of the corresponding computing time. On the other hand, in the case where only one vertical layer is used, one has  $N_z = 1$  and  $\Delta z_{j,1}^n = H_j^n$ . The equations then reduce to consistent discretization of the two-dimensional shallow water equations in the same manner as the model with structured grids [4]. Thus, a two-dimensional semi-implicit algorithm for the vertically averaged shallow water equations is obtained, as a particular case, from the general three-dimensional model by specifying only one vertical layer.

## 6. PROPERTIES OF THE METHOD

The algorithm developed above is relatively simple, yet general and robust. A number of interesting properties concerning mass conservation, numerical accuracy, stability and general-ity are discussed below.

- (a) In the present scheme, the local mass conservation is assured by the finite volume form (16) used to discretize the incompressibility condition (3). Also, local two-dimensional and global mass conservation is guaranteed by Equation (11), which is a finite volume discretization of the free surface equation (4).
- (b) When the polygons of the horizontal mesh are uniform rectangles, this algorithm reduces to the semi-implicit finite difference model presented in [4] and further analyzed and improved in [5]. The performance and the usefulness of such a model has been widely documented, e.g. [2,3,16]. The present algorithm represents an extension of the uniform grid finite difference formulation to unstructured grids, which adds considerable flexibility to the model.
- (c) The highest numerical accuracy is obtained when a uniform grid, such as equilateral triangles or uniform rectangles, is used. In these cases, the normal velocity on each side of the polygon is located at the center point of the side and the centers of two adjacent polygons are equally spaced from the common side. Consequently, the discretization error for the gravity wave terms in Equations (13) and (14) is second-order in space. Moreover, these equations are also second-order accurate in time for  $\theta = 1/2$ . Since a uniform grid, in general, does not fit the boundaries and does not allow for local mesh refinements, an unstructured, non-uniform grid can be used with a somewhat larger discretization error. This error can be minimized when the polygon size and shape vary gradually through the flow domain [18].
- (d) The stability analysis of the semi-implicit finite difference method (13) and (14) has been carried out in [5] for the case of a uniform rectangular grid and under the assumptions that the governing differential equations (1)–(4) are linear, with constant coefficients and defined on an infinite horizontal domain, or with periodic boundary conditions on a finite

domain. The analysis shows that the method is stable in the von Neumann sense if  $1/2 \leq \theta \leq 1$ , and if the operator  $F$  used to discretize the Coriolis, advection and horizontal friction terms is itself stable. Computational results on several test cases have indicated that no additional stability restrictions are required when a non-uniform unstructured mesh is used. Thus, the stability of the present algorithm is independent of the celerity, wind stress, vertical viscosity and bottom friction. It does depend on the discretization of the advection and horizontal friction terms. When an Eulerian–Lagrangian method is used for the explicit terms, a mild limitation on the time step depends on the horizontal viscosity coefficient and on the smallest polygon size. This method becomes unconditionally stable when the horizontal friction terms are neglected. Use of Eulerian–Lagrangian discretization is always recommended because a higher accuracy is obtained [1,4,15].

- (e) When only one vertical layer is specified ( $N_z = 1$ ), then  $\Delta z_{j,1}^n = H_j^n$  and the present algorithm reduces to a semi-implicit scheme that is consistent with the two-dimensional, vertically integrated shallow water equations. This property of the present algorithm leads to a general purpose computer code that can be used for one-dimensional channels, two-dimensional vertically averaged problems, and three-dimensional problems. More importantly, when the three-dimensional model is applied to a typical coastal embayment characterized by a mixture of channels and bays, these different geometric features are connected as part of the formulation without any special constraints. An example is a shallow coastal plain estuary with deep channels that are connected to the channels in a river delta.
- (f) The two-dimensional finite element version of this model [10] has a similar structure to the model developed here, with the same finite volume form of the continuity equation, a semi-implicit time integration, and Eulerian–Lagrangian advection. The two models end up with an identical structure to the discretized wave equation; however, the approximation to the pressure gradient term in the momentum equations is different because of the different approximation methods used. As noted below, this difference leads to different convergence rates for a linear solution on a regular grid but similar results for irregular grids.

## 7. COMPUTER APPLICATIONS

The model is applicable to realistic field scale problems with different flow regimes and complexity. There are three test problems presented here. The first is a test case with an analytical solution that is used to assess the optimal convergence rates that can be obtained. The second problem is a simulation of a two-dimensional flood that spreads over the Big Lost River flood plain and is used to assess the performance of the model (particularly the wetting and advection algorithms) in a highly irregular but realistic geometry. Finally, the flow simulation in the Jade–Weser Estuary shows the suitability of the present algorithm for the simulations of complex, three-dimensional environmental flows using an extremely fine spatial resolution and a relatively large time step.

### 7.1. Tidal channel

The geometry for this test case is a one-dimensional bay with length  $L = 4$  km and depth  $h = 10$  m. The water surface elevation is forced at one end with a periodic function with an amplitude  $a = 0.001$  m and period of  $T = 3600$  s. There is no bottom friction, advection or Coriolis effects included. The small amplitude leads to a linear response for comparison with a solution of the linear equations. The analytical solution for this case is one-dimensional and is derived relatively easily (see [19], p. 191). It is

$$u = \frac{ac \sin(kx)}{h \cos(kL)} \sin(\omega t), \quad (18)$$

$$\eta = a \frac{\cos(kx)}{\cos(kL)} \cos(\omega t), \quad (19)$$

subject to the boundary conditions  $u = 0$  at  $x = 0$  and  $\eta = a \cos(\omega t)$  at  $x = L$ , where  $\omega = 2\pi/T$  is the angular frequency,  $c = (gh)^{1/2}$  is the celerity and  $k = \omega/c$  is the wave number.

The standard error between the model solution and this analytical solution is calculated at the computational nodes for various grid refinements to show the convergence rate as the grid is refined and the time step is reduced.

The channel is discretized with equilateral triangles, one triangle wide and a variable number of triangles in length. From the base grid of three triangles (2 degrees of freedom (df) at sea level), the grid was refined to 4, 6 and 8 df. The time integration is centered with  $\theta = 1/2$ . The results in Figure 2 indicate that, for a regular grid, the model converges with the optimal rates predicted from theory,  $O(\lambda^2)$ . Refining the time step from 600 to

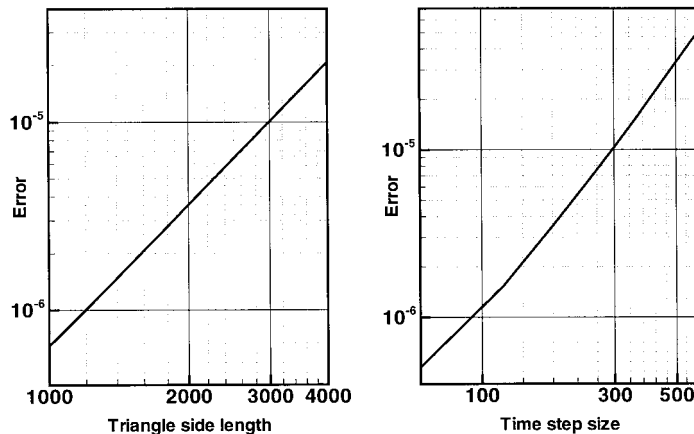


Figure 2. Convergence rates with grid refinement (left) and with time step refinement (right).

60 s, the results indicate that the time discretization is also second-order as expected from theory (see Figure 2).

On the other hand, the  $L^2$  error norm is applicable to the finite element version of this model, where the standard error is integrated over the spatial domain. Previous results [10] show that the finite element model converges at  $O(\lambda)$  as predicted by theory. Theory also indicates that there is a superconvergence for this regular grid where the convergence rate is  $O(\lambda^2)$ .

For irregular grids, these optimal convergence rates deteriorate to  $O(\lambda)$  because the approximation for gradients are no longer second-order accurate. Carefully grading the size of the elements and maintaining a nearly equilateral shape helps to mitigate this deterioration.

### 7.2. Big Lost River flood

The Big Lost River grid was originally developed to determine the inundation area in this reach for several floods with fixed discharges. Subsequently, simulations using this grid have been used to test the stability and efficiency of several numerical models. Here, we use this grid to test the model developed herein.

The reach of the Big Lost River, ID, USA, covers an area of about  $1 \times 1.2$  km<sup>2</sup>. The outflow elevation in the lower right part of the grid is maintained at a height of 0 m with respect to the local co-ordinate system, which corresponds to an elevation of 1540 m from mean sea level (MSL) in the global co-ordinate system. The topography with respect to the local co-ordinates is shown in Figure 3, where the local elevation ranges from  $-3$  to  $8$  m. There is a river channel that curves along the top boundary with several adjacent basins that are surrounded by higher elevations. The grid for this reach (Figure 4) contains  $N_p = 12622$  triangles and  $N_s = 19042$  sides. It was generated using the methods described in [20]. The forcing condition is a discharge of  $206 \text{ m}^3 \text{ s}^{-1}$ , which is injected at the upper left end of the river channel. The simulation is two-dimensional ( $N_z = 1$ ) and starts from rest with a water surface elevation of 0 m. For this simulation,  $f = 0$ . Moreover, since  $\theta = 1/2$  is at the limit of the stability range, the value  $\theta = 0.6$  has been chosen to ensure stability.

The large discharge applied suddenly creates a flood wave that moves down the channel causing rapid inundation of the low-lying areas and flows that are almost critical. As an example, the water surface elevation obtained after 8 min of simulated time with a time step  $\Delta t = 10$  s is characterized by a train of waves that emanate from a constriction in the flow (Figure 5). The velocity field at the constriction shows an eddy on the inside of the major bend and another eddy where the main flow passes the flooded area on the left (Figure 6). These are common features that are expected in these types of flow.

The flow is advection-dominated, such that the major part of the force balance is between advection and the pressure gradient force when the simulation approaches a steady state. This can be seen in the results where the acceleration of the flow through constrictions causes the large inundation areas above the constrictions. Numerous simulations with several models have shown that, without advection included, the results are entirely

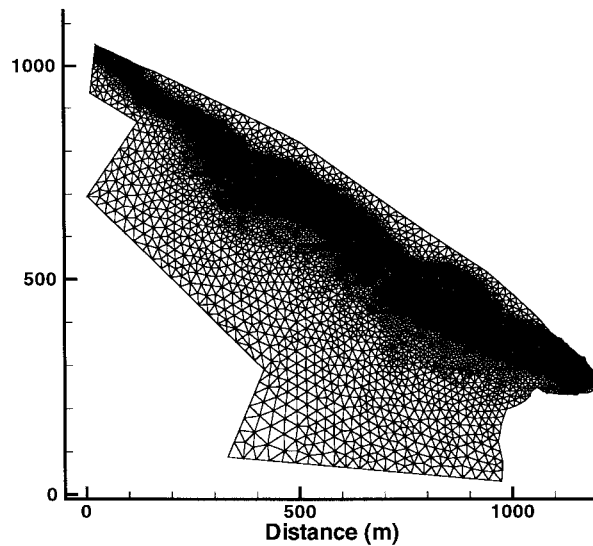


Figure 3. Topography for the Big Lost River reach.

different and show a lowered water surface elevation and unrealistically high velocities. Because of the strong advective effects, the flow is relatively insensitive to the bottom friction coefficient. Thus, this problem is a rather severe test case for numerical models [10].

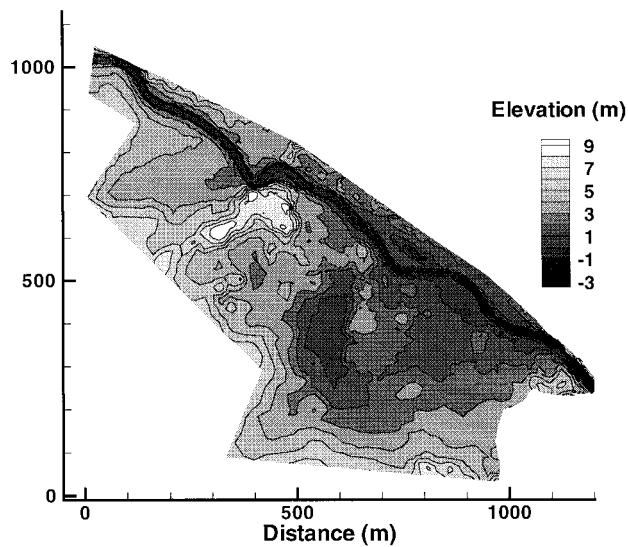


Figure 4. Triangular grid for the Big Lost River reach.

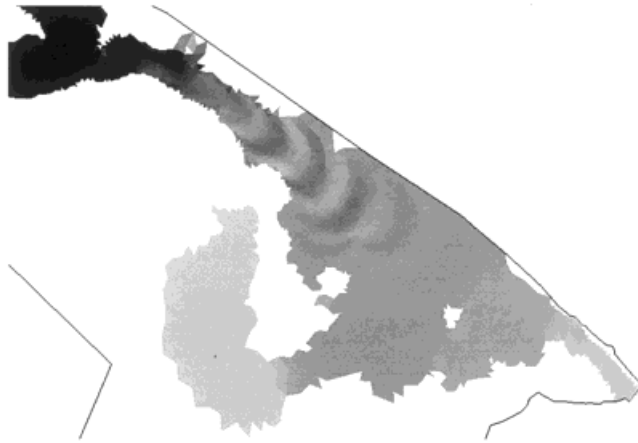


Figure 5. Water levels obtained after 8 min of simulated time.

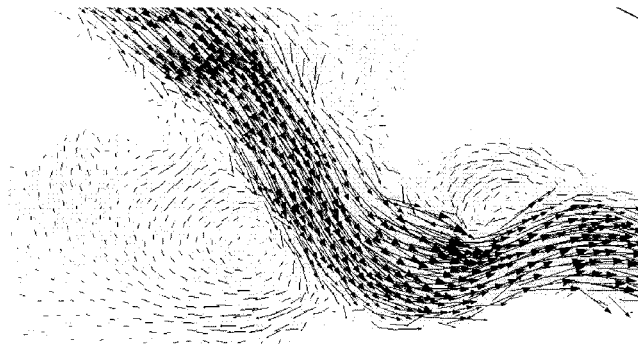


Figure 6. Velocity at the constriction obtained after 8 min of simulated time.

### 7.3. Jade–Weser Estuary

The last example is circulation in the Jade–Weser Estuary, located along Germany's North Sea coast. The water flow is complicated because of a highly irregular topography, the presence of tidal flats and man-made structures, large tidal range, strong winds, and a relatively large fresh water input. The Jade–Weser Estuary grid covers an area of about 1974 km<sup>2</sup> and has a maximum depth from MSL of about 28 m (see Figure 7).

The grid for the Jade–Weser Estuary was generated at Bundesanstalt für Wasserbau (BAW), Hamburg. It has a very fine resolution, with  $N_p = 150286$  irregular triangles, whose size varies from 10 to 80068 m<sup>2</sup>. The total number of sides is  $N_s = 228655$ . The use of an unstructured orthogonal grid for this site enables accurate resolution of the flow region,

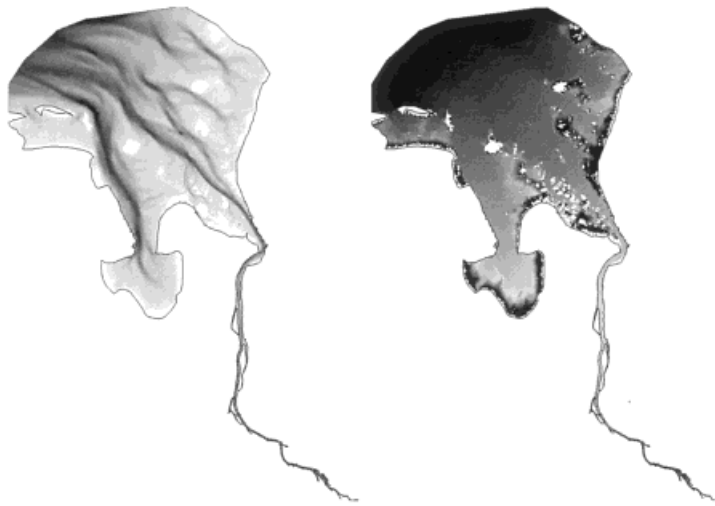


Figure 7. Bathymetry of the Jade–Weser Estuary (left) and water surface elevation during flood (right).

including detailed representation of man-made structures and narrow branches of the Weser River.

In the present simulations, the flow is driven by a constant fresh water input of  $180 \text{ m}^3 \text{ s}^{-1}$  from the Weser River, and by a surface elevation prescribed at the open boundary with the North Sea. These boundary conditions were interpolated from field data and include the 13 days period starting at 14:30 h on 12 June 1990. At the open boundary, the tidal amplitudes are of the order of 1.25 m and are dominated by the  $M_2$  (12.42 h) constituent. The time step is  $\Delta t = 300 \text{ s}$ , the duration of the simulation is 13 days, and the implicitness factor is  $\theta = 0.6$ .

Three different runs have been made corresponding to a two-dimensional simulation with  $N_z = 1$  and three-dimensional simulations with  $N_z = 6$  and  $N_z = 60$ . The corresponding vertical resolution is  $\Delta z_j^n = H_j^n$ ,  $\Delta z_{j,k}^n \leq 5 \text{ m}$  and  $\Delta z_{j,k}^n \leq 0.5 \text{ m}$  respectively. Table I summarizes the total number of active prisms ( $N_p^{3D}$ ), the total number of horizontal velocity points ( $N_s^{3D}$ ) and the ratio  $R = (\text{real time})/(\text{CPU time})$ , for each case. The simulations were performed on a 500 MHz DEC Alpha 21164, which has a SpecFP 95 rating of 17.8. The largest amount of CPU time is spent for determining the free surface elevation in the system of Equation (15), and for computing the Lagrangian trajectories needed to approximate the advection terms with an

Table I. The model's performance on the Jade–Weser Estuary

	$N_p^{3D}$	$N_s^{3D}$	$R$
$N_z = 1$	150 286	228 655	10.34
$N_z = 6$	257 442	370 705	7.66
$N_z = 60$	1 740 356	2 466 926	3.67

Eulerian–Lagrangian method. The computational effort needed for solving system (15) is independent from  $N_z$ , while the CPU time needed for the advection terms is proportional to  $N_s^{3D}$ .

Figure 7 shows the computed water surface elevation during the flood phase, obtained with  $N_z = 60$  at a time when the specified water surface elevation at the open boundary with the North Sea is near zero. There are extensive mud flats that wet and dry during the tidal cycle. The dark areas in the right panel of Figure 7 show the areas of water that have been trapped by the falling tide. There is a considerable phase difference between the open boundary and the Weser River.

Figure 8 presents characteristic plots of time series for sea level. The station with the smallest amplitude is near the center of the open boundary; the station with the intermediate amplitude is at the mouth of the Weser River, and the station with largest amplitude is about two-thirds of the way upstream of the Weser River in Figure 7. Note the increase in amplitude and a phase lag with increasing distance from the open boundary. A comparison between observations and model results shows that the model duplicates this behavior quite well. In addition, note the distortion of the tidal wave as the wave progresses further upstream the Weser River. Again the model produces a good representation of this non-linear behavior.

The surface velocity distribution at the start of the ebb tide at the mouth of the Weser River is shown in Figure 9. The flow in the shallow areas along the channel has changed to ebb while the flow in the main channel is still on flood. This is a common feature in estuaries and tidal rivers, and is readily reproduced by the model. The plot in the right panel of Figure 9 shows the flow at 12 m below the surface in the channel. The pattern of velocity at the start of the flood cycle is similar to the pattern shown here except that the velocity directions are reversed and more dry area is exposed because of the lower water level. Vertical profiles of velocity at various points in the grid show a typical boundary layer profile that changes direction over the tidal cycle [12].

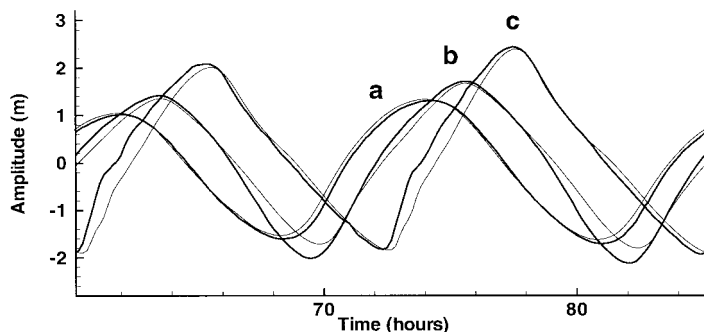


Figure 8. Sea level variation at three locations: (a) near the open boundary, (b) at the mouth of the Weser River, and (c) upstream. The heavy lines are the observations and the light lines are model results.



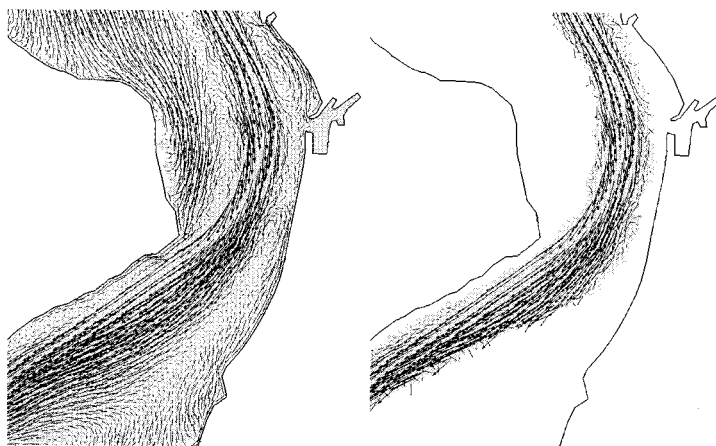


Figure 9. Horizontal velocity near the mouth of the Weser River. Surface flow (left), flow at 12 m below the surface (right).

## 8. CONCLUSIONS

A semi-implicit algorithm for the three-dimensional shallow water flow has been derived, discussed and applied. The horizontal discretization consists of an orthogonal unstructured grid that provides a great amount of flexibility for fitting boundaries and for local mesh refinements. The resulting model is numerically stable, relatively simple and very efficient. In the particular case that only one vertical layer is specified, this algorithm solves the vertically integrated, two-dimensional shallow water equations. Computationally, the resulting scheme conserves mass and is suitable for the simulations of complex one-, two- and three-dimensional hydrostatic flows using fine spatial resolution and relatively large time steps. The present formulation allows for the simulation of flooding and drying in a straightforward manner. The computed results for the three applications verify the performance and robust nature of the model.

## ACKNOWLEDGMENTS

The authors gratefully acknowledge D. Ambrosi, L. Bonaventura, L. Paglieri, F. Saleri, A. Quarteroni and P. Zanolli for helpful discussions; and C. Berenbock and G. Lang for providing the grids of the Big Lost River and the Jade–Weser Estuary respectively.

## REFERENCES

1. V. Casulli, 'Semi-implicit finite difference methods for the two-dimensional shallow water equations', *J. Comput. Phys.*, **86**, 56–74 (1990).
2. R.P. Signell and B. Butman, 'Modeling tidal exchange and dispersion in Boston Harbor', *J. Geophys. Res.*, **97**, 15591–15606 (1992).

3. R.T. Cheng, V. Casulli and J.W. Gartner, 'Tidal, residual, intertidal mudflat (TRIM) model and its applications to San Francisco Bay, California', *Estuar. Coast. Shelf Sci.*, **36**, 235–280 (1993).
4. V. Casulli and R.T. Cheng, 'Semi-implicit finite difference methods for three-dimensional shallow water flow', *Int. J. Numer. Methods Fluids*, **15**, 629–648 (1992).
5. V. Casulli and E. Cattani, 'Stability, accuracy and efficiency of a semi-implicit method for three-dimensional shallow water flow', *Comput. Math. Appl.*, **27**, 99–112 (1994).
6. A.F. Blumberg and G.L. Mellor, 'A description of a three dimensional coastal ocean circulation model', in N.S. Heaps (ed.), *Three Dimensional Coastal Ocean Circulation Models, Coastal and Estuarine Sciences 4*, AGU, Washington, DC, 1987, pp. 1–16.
7. J. van der Molen and G.S. Stelling, 'A non-orthogonal finite difference method for shallow water flow', in J. van der Molen (ed.), *Tides in a Salt-Marsh*, Feboedruk BV, Enschede, Netherlands, 1997, pp. 29–59.
8. E.S. Oran and J.P. Boris, *Numerical Simulation of Reactive Flow*, Elsevier, New York, 1987.
9. W.C. Thacker, 'Irregular grid finite-difference techniques: simulations of oscillations in shallow circular basins', *J. Phys. Oceanogr.*, **7**, 284–292 (1977).
10. R.A. Walters and V. Casulli, 'A robust finite element model for hydrostatic surface water flows', *Commun. Numer. Methods Eng.*, **14**, 931–940 (1998).
11. G. F. Carey (ed.), *Finite Element Modeling of Environmental Problems*, Wiley, New York, 1995.
12. V. Casulli and P. Zanolli, 'A three-dimensional semi-implicit algorithm for environmental flows on unstructured grids', *Proc. Conf. on Numerical Methods for Fluid Dynamics*, University of Oxford, 1998.
13. S. Rebay, 'Efficient unstructured mesh generation by means of delaunay triangulation and Bowyer–Watson algorithm', *J. Comput. Phys.*, **106**, 125–138 (1993).
14. R.A. Nicolaides, 'The covolume approach to computing incompressible flows', in M.D. Gunzburger and R.A. Nicolaides (eds.), *Incompressible Computational Fluid Dynamics*, Cambridge University Press, Cambridge, 1993, pp. 295–333.
15. A. Staniforth and C. Temperton, 'Semi-implicit semi-Lagrangian integration schemes for a barotropic finite element regional model', *Month. Weather Rev.*, **114**, 2078–2090 (1986).
16. E.S. Gross, V. Casulli, L. Bonaventura and J.R. Koseff, 'A semi-implicit method for vertical transport in multidimensional models', *Int. J. Numer. Methods Fluids*, **28**, 157–186 (1998).
17. G.H. Golub and C.F. van Loan, *Matrix Computations*, 3rd edn, J. Hopkins, London, 1996.
18. S. Gravel and A. Staniforth, 'Variable resolution and robustness', *Month. Weather Rev.*, **20**, 2633–2640 (1992).
19. R.A. Walters and R.T. Cheng, 'Accuracy of an estuary hydrodynamic model using smooth elements', *Water Resour. Res.*, **16**, 187–195 (1980).
20. R.F. Henry and R.A. Walters, 'A geometrically-based, automatic generator for irregular triangular networks', *Commun. Numer. Methods Eng.*, **9**, 555–566 (1993).