Journal of Scientific Computing, Volumes 22 and 23, June 2005 (© 2005)
DOI: 10.1007/s10915-004-4139-3

A Discontinuous Galerkin Method for Three-Dimensional Shallow Water Equations

Clint Dawson[1] **and Vadym Aizinger**[1]

Received August 4, 2003; accepted (in revised form) February 25, 2004

We describe the application of a local discontinuous Galerkin method to the numerical solution of the three-dimensional shallow water equations. The shallow water equations are used to model surface water flows where the hydrostatic pressure assumption is valid. The authors recently developed a DG method for the depth-integrated shallow water equations. The method described here is an extension of these ideas to non-depth-integrated models. The method and its implementation are discussed, followed by numerical examples on several test problems.

KEY WORDS: Shallow water equations; discontinuous Galerkin method; free surface.

1. INTRODUCTION

In this paper, we describe a local discontinuous Galerkin (LDG) method for the three-dimensional (3D) shallow water equations.

The shallow water equations are derived from the incompressible Navier–Stokes equations defined on a domain with a moving free (top) surface. The shallow water assumption reduces the vertical momentum equation to the hydrostatic pressure relation

$$\frac{\partial p}{\partial z} = -\rho g,$$

where p is the hydrostatic pressure, ρ is density, and g is gravitational acceleration. In many cases of practical importance, vertical effects are

[1]Center for Subsurface Modeling – C0200, Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin, Austin, TX 78712, USA. E-mail: clint@ices.utexas.edu

negligible and one can integrate the horizontal momentum equations and the continuity equation over the depth, applying appropriate boundary conditions at the bottom and free surface, to obtain the 2D shallow water equations. Where eddy viscosity and/or density variations are important, for example, where salinity and temperature are spatially varying, then the full 3D model should be employed. See [16, 17] for a discussion of shallow water models in both two and three dimensions.

Shallow water equations are complicated by many nonlinear effects and are defined on complex domains involving varying bottom topography and coastal shorelines. Viscosity effects, especially horizontal viscosity, are usually relatively small, and algorithms which are stable and accurate for smooth to highly advective flows on general geometries are of interest for the numerical solution of these models. A substantial literature exists on the application of various finite difference and finite element methods to the 3D shallow water equations; see, for example, [3, 10, 12, 13]. The search is still on, however, for methods which are locally mass conservative, can handle very general types of elements, and are stable and accurate under highly varying flow regimes. Recently developed algorithms such as the discontinuous Galerkin (DG) method are therefore of great interest within the shallow water community.

DG finite element methods are promising because of their ability to handle geometrically complex elements, use of shock-capturing numerical fluxes, adaptivity in polynomial order, and mass conservation properties; see [7] for a historical overview of DG methods. In [2, 4], we investigated DG and related finite volume methods for the solution of the 2D shallow water equations. Viscosity (second-order derivative) terms are handled in this method through the so-called LDG framework [5], which involves rewriting the model as a first-order system of equations. In this paper, we describe the extension of the LDG method to 3D shallow water models.

The rest of this paper is organized as follows. In the next section, we introduce the governing equations and boundary conditions for our model. In Sec. 3, we outline our solution strategy, which is described in detail in Secs. 4 and 5. In Sec. 6, some preliminary numerical results are given for problems with smooth and rough analytical solutions as well as for a typical tidal flow problem.

2. MATHEMATICAL FORMULATION

The mathematical model of 3D constant density shallow water flow includes momentum equations for the horizontal velocity components, a continuity equation, and boundary and initial conditions. The domain

over which these equations are defined has a free surface described by a kinematic boundary condition.

The conservative form of the momentum conservation equations can be written as follows:

$$\frac{\partial \mathbf{u}_{xy}}{\partial t} + \nabla \cdot \left(\mathbf{u}_{xy}\mathbf{u}^{\mathrm{T}} - \mathcal{D}\nabla \mathbf{u}_{xy} \right) + g\nabla_{xy}\xi - f_{\mathrm{c}}\mathbf{k} \times \mathbf{u}_{xy} = \mathbf{G}, \tag{2.1}$$

where $\nabla_{xy} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$, ξ is the value of the z coordinate at the free surface, $\mathbf{u} = (u, v, w)^{\mathrm{T}}$ is the velocity vector, $\mathbf{u}_{xy} = (u, v)^{\mathrm{T}}$ is the vector of horizontal velocity components, f_{c} is the Coriolis coefficient, $\mathbf{k} = (0, 0, 1)$ a unit vertical vector, g is acceleration due to gravity, $\mathbf{G} = \left(G_x, G_y \right)$ is a vector of body forces (it can include atmospheric pressure terms, tidal forcing, etc.), and $\mathcal{D}$ is a tensor of eddy viscosity coefficients defined as follows:

$$\mathcal{D} = \begin{pmatrix} D_u & 0 \\ 0 & D_v \end{pmatrix} \tag{2.2}$$

with D_u, D_v being 3×3 symmetric positive-definite matrices.

The continuity equation can be written as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \tag{2.3}$$

We augment the system with the following boundary conditions:

- At the bottom, we have no normal flow

$$\mathbf{u}(z_{\mathrm{b}}) \cdot \mathbf{n} = 0 \tag{2.4}$$

and no slip for horizontal velocity components

$$u(z_{\mathrm{b}}) = v(z_{\mathrm{b}}) = 0, \tag{2.5}$$

where z_{b} is the value of the z coordinate at the sea bed and $\mathbf{n} = (n_x, n_y, n_z)^T$ is an exterior unit normal to the face.

- The free surface boundary conditions have the form:

$$\frac{\partial \xi}{\partial t} + u(\xi)\frac{\partial \xi}{\partial x} + v(\xi)\frac{\partial \xi}{\partial y} - w(\xi) = 0. \tag{2.6}$$

$$\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial v}{\partial \mathbf{n}} = 0. \tag{2.7}$$

On the lateral boundaries, we can have several types of boundary conditions (note that we assume all lateral boundaries to be strictly vertical; therefore, if $\mathbf{n} = (n_x, n_y, n_z)^{\mathrm{T}}$ is an exterior normal to a lateral boundary face then $n_z = 0$):

- River boundary: Prescribed normal u_{n} and tangential u_τ velocities

$$un_x + vn_y = u_{\mathrm{n}}, \qquad -un_y + vn_x = u_\tau, \tag{2.8}$$

  and prescribed surface elevation $\xi_{\mathrm{r}}(x, y, t)$

$$\xi = \xi_{\mathrm{r}}(x, y, t). \tag{2.9}$$

- Land boundary: No normal flow

$$u_{\mathrm{n}} = \mathbf{u} \cdot \mathbf{n} = 0, \tag{2.10}$$

  and zero shear stress

$$\frac{\partial u_\tau}{\partial \mathbf{n}} = 0. \tag{2.11}$$

- Open sea boundary: Zero normal derivative of the horizontal velocity components

$$\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial v}{\partial \mathbf{n}} = 0, \tag{2.12}$$

  and prescribed surface elevation $\xi_{\mathrm{s}}(x, y, t)$

$$\xi = \xi_{\mathrm{s}}(x, y, t). \tag{2.13}$$

- Radiation boundary: Zero normal derivative of the horizontal velocity components

$$\frac{\partial u}{\partial \mathbf{n}} = \frac{\partial v}{\partial \mathbf{n}} = 0. \tag{2.14}$$

Analytically, the free surface elevation can be computed using Eq. (2.6), but, numerically, a more robust way to do it is to integrate the continuity Eq. (2.3) over the depth and, taking into account the boundary conditions at the top and bottom, obtain the following 2D equation for the surface

elevation commonly called the primitive continuity equation (PCE):

$$\frac{\partial \xi}{\partial t} + \frac{\partial}{\partial x}\int_{z_b}^{\xi} u\, dz + \frac{\partial}{\partial y}\int_{z_b}^{\xi} v\, dz = 0. \tag{2.15}$$

The equation above or, alternatively, the wave equation derived by substituting the 2D momentum equations in Eq. (2.15) is a standard part of just about every 2- and 3D shallow water model in existence today, though using this equation to compute the position of the free surface within a framework of a 3D solver based on the DG method is not trivial, that is unless we resort to solving an auxiliary 2D problem as is done in many 3D shallow water solvers. We will discuss this issue in some detail in Sec. 4.

Let us denote $h = \xi - z_b$. Then, we can rewrite conservation Eqs. (2.15) and (2.1) in the following compact form:

$$\frac{\partial h}{\partial t} + \nabla_{xy} \cdot \mathbf{C_H} = 0, \tag{2.16}$$

$$\frac{\partial \mathbf{u}_{xy}}{\partial t} + \nabla \cdot (\mathcal{C}_M - \mathcal{D}\nabla \mathbf{u}_{xy}) = \mathbf{F}, \tag{2.17}$$

where

$$\mathbf{C_H} = \begin{pmatrix} \int_{z_b}^{\xi} u\, dz \\ \int_{z_b}^{\xi} v\, dz \end{pmatrix}, \quad \mathcal{C}_M = \begin{pmatrix} u^2 + gh & uv \\ uv & v^2 + gh \\ uw & vw \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} G_x + g\dfrac{\partial z_b}{\partial x} + f_c v \\ G_y + g\dfrac{\partial z_b}{\partial y} - f_c u \end{pmatrix}.$$

Thus, our system consists of the PCE (2.16), two momentum conservation equations for horizontal velocity components (2.17), and the continuity equation (2.3).

3. GENERAL ISSUES AND SOLUTION STRATEGY

3.1. Solution Strategy

The general solution strategy employed in our implementation is not substantially different from ones found in other 3D solvers. The main differences lie in the fact that all state variables are approximated in space with functions which are continuous on each element but can be discontinuous across inter-element boundaries. For time stepping, we use explicit Runge–Kutta methods choosing the time step and scheme order appropriately to keep the error of the time stepping algorithm well within the error of space discretization.

Within a time step (substep in case of multistage Runge–Kutta methods), first, we solve the mass and momentum conservation equations. These equations are tightly coupled and must be dealt with simultaneously. Then, for given values of horizontal velocity components $\mathbf{u}_{xy} = (u, v)^{\mathrm{T}}$, we compute vertical velocity w from the discrete continuity equation to obtain a divergence-free velocity field. The latter equation is not time dependent and is solved as an initial value problem layer-by-layer starting at the bottom and using the solution from the layer below (or boundary condition (2.4) at the bottom in case of the bottommost layer) as an initial value.

Every few time steps the mesh geometry is updated using computed values of the surface elevation. Frequency of mesh update can be chosen according to the type of problem we are solving.

3.2. Computational Mesh

The most common type of mesh employed in 3D finite element simulations of shallow water flow is a 2D grid projected vertically and subdivided into layers using a Cartesian or σ-stretched coordinate system. This approach agrees well with the physical anisotropy of the problem, in which the vertically directed gravity force usually is the main body force acting on the system. The grids in our implementation also belong to this type. In our model, we use prismatic elements with triangular cross-section, strictly vertical lateral face and flat but not necessarily parallel top and bottom faces.

An important feature of most modern 3D shallow water solvers, the dynamic free surface, turns out to be the main stumbling block on the way to a stable DG implementation. Since the surface elevation is one of our state variables living in a discontinuous approximation space, the grid determined by our discrete solution would also have a discontinuous free surface, thus, the lateral faces shared by neighboring elements in the uppermost layer of the mesh would, in general, not match. This causes the boundary integrals over those faces to be ill-defined. In order to avoid this problem, we smooth the free surface of our mesh (e.g., by the least squares fit) and compute all 3D integrals on a grid with a globally continuous free surface. This procedure only affects the geometry of elements and faces in the uppermost layer of the computational mesh and does not change in any way the computed values of the state variables or the discontinuous character of the numerical solution. Thus, the local conservation properties of the LDG method are not degraded by this mesh smoothing postprocessing. In our experiments, the stability and accuracy of the numerical

solution did not seem to be significantly influenced by the mesh smoothing algorithm used.

3.3. Treatment of Discontinuities

In order to better understand the issues discussed below we have to keep in mind the fact that we are solving a system of PDEs by a numerical method in which the approximations to all state variables are discontinuous across inter-element boundaries. This type of discretization gives rise to a Riemann problem at every Gauss integration point on those boundaries. In addition, the mass and momentum conservation equations are tightly coupled. That puts us in a difficult situation: Eq. (2.16) is a 2D equation and is not well-defined on the lateral boundaries of our 3D elements unlike the momentum Eq. (2.17). Uncoupling these equations or computing the surface elevation from (2.6) results in an unstable numerical model because, in that case, the boundary discontinuities are not resolved properly. Since we did not want to resort to solving an auxiliary 2D shallow water problem for the surface elevation, we modified instead the boundary integrals in the discrete mass conservation equation to bring them in a 3D form, thus obtaining a well defined Riemann problem at every Gauss point on the lateral 3D faces.

Here, it must be also noted that the Riemann problems corresponding to the horizontal faces are significantly less demanding in terms of stability than the ones on lateral faces. This situation is due to the fact that the surface elevation is continuous across horizontal inter-element interfaces. Our experience shows that computing nonlinear boundary fluxes on those faces by upwinding or central differences does not cause any instability. The more complex Riemann problems on lateral interfaces are handled in our implementation with the help of Roe's solver that will be discussed in some detail in a separate section.

4. SPACE DISCRETIZATION, THE LDG METHOD

To discretize our problem in space using the LDG method, we first introduce an auxiliary flux variable $\mathcal{Q}$ and rewrite the momentum conservation equations in the mixed form:

$$\frac{\partial \mathbf{u}_{xy}}{\partial t} + \nabla \cdot (\mathcal{C}_{\mathrm{M}} + \sqrt{\mathcal{D}}\mathcal{Q}) = \mathbf{F}, \tag{4.1}$$

$$\mathcal{Q} = -\sqrt{\mathcal{D}}\nabla \mathbf{u}_{xy}. \tag{4.2}$$

Let $\mathcal{T}_h$ be a general partition of our 3D domain Ω and let $\Omega_{\mathrm{e}} \in \mathcal{T}_h$. To obtain a weak formulation we multiply the equations above by smooth

test functions $\boldsymbol{\phi}, \Psi$; integrate them on each element $\Omega_e \in \mathcal{T}_h$; and, finally, integrate by parts obtaining the following expressions:

$$\int_{\Omega_e} \frac{\partial \mathbf{u}_{xy}}{\partial t}\ \boldsymbol{\phi}\ dx\,dy\,dz + \int_{\partial\Omega_e} (\mathcal{C}_M + \sqrt{\mathcal{D}}\mathcal{Q})\cdot\mathbf{n}\ \boldsymbol{\phi}\ ds - \int_{\Omega_e} (\mathcal{C}_M + \sqrt{\mathcal{D}}\mathcal{Q})\cdot\nabla\boldsymbol{\phi}\ dx\,dy\,dz = \int_{\Omega_e} \mathbf{F}\ \boldsymbol{\phi}\ dx\,dy\,dz, \tag{4.3}$$

$$\int_{\Omega_e} \sqrt{\mathcal{D}^{-1}}\ \mathcal{Q}\ \Psi\ dx\,dy\,dz = -\int_{\partial\Omega_e} \mathbf{u}_{xy}\ \Psi\cdot\mathbf{n}\ ds + \int_{\Omega_e} \mathbf{u}_{xy}\ \nabla\cdot\Psi\ dx\,dy\,dz, \tag{4.4}$$

where $\mathbf{n}$ is a unit exterior normal to Ω_e. This weak formulation is well defined for any $\mathbf{u}_{xy}(t,x,y,z) \in H^1(0,T;\ V^{d-1})$; $w(t,x,y,z) \in V,\ \forall t \in [0,T]$; $\boldsymbol{\phi}(x,y,z) \in V^{d-1}$; $\mathcal{Q}(t,x,y,z) \in Y^{d-1},\ \forall t \in [0,T]$; and $\Psi(x,y,z) \in Y^{d-1}$, where

$$V \stackrel{\text{def}}{=} L^2(\Omega) \bigcap \{u : u_{|\Omega_e} \in H^1(\Omega_e),\quad \forall \Omega_e \in \mathcal{T}_h\}, \tag{4.5}$$

$$Y \stackrel{\text{def}}{=} L^2(\Omega)^d \bigcap \{\mathbf{q} : \mathbf{q}_{|\Omega_e} \in H^1(\Omega_e)^d,\quad \forall \Omega_e \in \mathcal{T}_h\}. \tag{4.6}$$

Next, we seek to approximate $(\mathbf{u}_{xy}(t,\cdot), w(t,\cdot), \mathcal{Q}(t,\cdot))$, the solution to the weak problem, with a function $(\mathbf{U}_{xy}(t,\cdot), W(t,\cdot), \mathcal{Q}(t,\cdot)) \in U_h \times W_h \times Z_h$, where $U_h \subset V^{d-1}$, $W_h \subset V$, and $Z_h \subset Y^{d-1}$ are some finite-dimensional subspaces. In order to do so we can use our weak formulation with one important modification. Since our approximation space does not guarantee continuity across the inter-element boundaries, the integrands in the boundary integrals have to be replaced by suitably chosen numerical fluxes preserving consistency and stability of the method. The semi-discrete finite element solution $(\mathbf{U}_{xy}(t,\cdot), \mathcal{Q}(t,\cdot))$ is obtained by requiring that for any $t \in [0,T]$, all $\Omega_e \in \mathcal{T}_h$, and for all $(\boldsymbol{\phi}, \Psi) \in U_h \times Z_h$ the following holds:

$$\int_{\Omega_e} \frac{\partial \mathbf{U}_{xy}}{\partial t}\ \boldsymbol{\phi}\ dx\,dy\,dz + \int_{\partial\Omega_e} (\hat{\mathbf{C}}_{M,\mathbf{n}} + \sqrt{\mathcal{D}}\hat{\mathcal{Q}}\cdot\mathbf{n})\ \boldsymbol{\phi}\ ds - \int_{\Omega_e} (\mathcal{C}_M + \sqrt{\mathcal{D}}\mathcal{Q})\cdot\nabla\boldsymbol{\phi}\ dx\,dy\,dz = \int_{\Omega_e} \mathbf{F}\ \boldsymbol{\phi}\ dx\,dy\,dz, \tag{4.7}$$

$$\int_{\Omega_e} \sqrt{\mathcal{D}^{-1}}\ \mathcal{Q}\ \Psi\ dx\,dy\,dz = -\int_{\partial\Omega_e} \hat{\mathbf{U}}_{xy}\ \Psi\cdot\mathbf{n}\ ds + \int_{\Omega_e} \mathbf{U}_{xy}\ \nabla\cdot\Psi\ dx\,dy\,dz, \tag{4.8}$$

where $\hat{\mathbf{C}}_{M,\mathbf{n}}$ is a solution to the Riemann problem for the nonlinear boundary flux $\mathcal{C}_M\cdot\mathbf{n}$. We set $\hat{\mathbf{U}}_{xy}, \hat{\mathcal{Q}}$ equal to the averages of the

corresponding variables on both sides of the discontinuity. (Note that there are other possible choices of $\hat{\mathbf{U}}_{xy}$ and $\hat{\mathcal{Q}}$.)

Discretization of the primitive continuity equation is done in a similar way. Let $\Omega_{\mathrm{e},xy}$ be the orthogonal projection of Ω_{e} into the xy-plane. We multiply Eq. (2.16) by a smooth test function $\delta = \delta(x, y)$, integrate it over $\Omega_{\mathrm{e},xy}$, and integrate by parts. Then, the mass balance in the water column corresponding to the 2D element $\Omega_{\mathrm{e},xy}$ can be expressed as

$$\int_{\Omega_{\mathrm{e},xy}} \frac{\partial h}{\partial t}\ \delta\ dx\,dy + \int_{\partial\Omega_{\mathrm{e},xy}} \mathbf{C_H}\cdot\mathbf{n}\ \delta\ ds - \int_{\Omega_{\mathrm{e},xy}} \mathbf{C_H}\cdot\nabla_{xy}\ \delta dx\,dy = 0. \tag{4.9}$$

Noting that $\mathbf{C_H} = \left(\int_{z_\mathrm{b}}^{\xi} u dz, \int_{z_\mathrm{b}}^{\xi} v dz\right)^{\mathrm{T}}$ and using the facts that δ is independent of z and that $h > 0$ we can transform the equation above as follows:

$$\int_{\Omega_{\mathrm{e}},xy} \frac{\partial h}{\partial t}\ \delta\ dx\,dy\ + \sum_{\Omega_\mathrm{e}\in\mathrm{col}(\Omega_{\mathrm{e},xy})} \int_{\partial\Omega_{\mathrm{e,lat}}} \frac{\mathbf{u}_{xy}h\cdot\mathbf{n}}{h}\ \delta\ ds$$
$$- \sum_{\Omega_e\in col(\Omega_{e,xy})} \int_{\Omega_e} \frac{\mathbf{u}_{xy}h\cdot\nabla_{xy}}{h}\ \delta\ dx\,dy\,dz = 0, \tag{4.10}$$

where $\partial\Omega_{\mathrm{e,lat}}$ denotes the lateral boundary faces of prism Ω_{e}, and $\mathrm{col}(\Omega_{\mathrm{e},xy})$ is the set of 3D elements in the water column corresponding to $\Omega_{\mathrm{e},xy}$. Note that the expression above is well defined for any $\delta(x, y) \in \mathcal{H} \stackrel{\mathrm{def}}{=} L^2(\Omega_{xy}) \bigcap \{h : h_{|\Omega_{\mathrm{e},xy}} \in H^1(\Omega_{\mathrm{e},xy}), \forall \Omega_\mathrm{e} \in \mathcal{T}_h\}$ and $h(t, x, y) \in H^1(0, T; \mathcal{H})$, where Ω_{xy} is the orthogonal projection of the domain Ω into the xy-plane.

Analogous to the momentum conservation equations, we seek an approximation $H(t, \cdot) \in \mathcal{H}_h$ to the solution of the weak problem $H(t, \cdot)$, where $\mathcal{H}_h \subset \mathcal{H}$ is some finite dimensional subspace. Using the weak formulation (4.10) and replacing integrands in the boundary integrals by a suitable numerical flux we obtain our semi-discrete finite element solution $H(t, \cdot) \in \mathcal{H}_h$ by requiring that for any $t \in [0, T]$, all $\Omega_\mathrm{e} \in \mathcal{T}_h$, and for all $\delta \in \mathcal{H}_h$ the following holds:

$$\int_{\Omega_{\mathrm{e}},xy} \frac{\partial H}{\partial t}\ \delta\ dx\,dy\ + \sum_{\Omega_\mathrm{e}\in\mathrm{col}(\Omega_{\mathrm{e},xy})} \int_{\partial\Omega_{\mathrm{e,lat}}} \frac{\hat{C}_H}{\xi_\mathrm{s} - z_\mathrm{b}}\ \delta\ ds$$
$$- \sum_{\Omega_\mathrm{e}\in\mathrm{col}(\Omega_{\mathrm{e},xy})} \int_{\Omega_\mathrm{e}} \frac{\mathbf{U}_{xy}H\cdot\nabla_{xy}}{\xi_\mathrm{s} - z_\mathrm{b}}\ \delta\ dx\,dy\,dz = 0, \tag{4.11}$$

where $\hat{C}_H$ is a solution to the Riemann problem for the normal boundary flux $\mathbf{U}_{xy}H\cdot\mathbf{n}$, and ξ_s denotes the value of the z coordinate at the free

surface of the smoothed mesh. This boundary flux formulation has several important advantages. It transforms integrals over 2D edges into integrals over lateral faces of 3D elements, thus allowing us to solve the Riemann problem for elevation simultaneously with the corresponding problem for the momentum equations; it is consistent with the continuous formulation; and it takes into account the coupling between velocity and elevation which is crucial for stability of our numerical scheme.

Finally, we turn to the space discretization for continuity Eq. (2.3). Unlike the mass and momentum conservation equations it is not time-dependent, its main role being computation of the vertical velocity component w to maintain a divergence-free velocity field. Regarding the continuity equation with the boundary condition at the bottom as an initial value problem for w, we can solve it element by element in each water column starting at the bottom and using the solution from the element below to provide an initial condition.

Multiplying (2.3) by a smooth test function σ, integrating it over Ω_e, integrating by parts, and re-ordering terms we obtain a weak formulation (recall that $n_z = 0$ on the lateral faces):

$$\int_{\partial\Omega_{e,top}} w\, n_z\, \sigma\, ds - \int_{\Omega_e} w \left(\frac{\partial \sigma}{\partial z}\right) dx\, dy\, dz = \int_{\Omega_e} \mathbf{u}_{xy} \cdot \nabla_{xy} \sigma \; dx\, dy\, dz - \int_{\partial\Omega_{e,bot}} w\, n_z\, \sigma\, ds - \int_{\partial\Omega_e} \mathbf{u}_{xy} \cdot \mathbf{n}\, \sigma\, ds, \tag{4.12}$$

where $\partial\Omega_{e,top}, \partial\Omega_{e,bot}$ denote the top and bottom boundaries of element Ω_e.

We seek $W(t, \cdot) \in W_h$, where W_h is some finite dimensional subspace of V, such that for given values of $\mathbf{U}_{xy}(t, \cdot) \in U_h$, for all $\Omega_e \in \mathcal{T}_h$, and for all $\sigma \in W_h$ the following holds:

$$\int_{\partial\Omega_{e,top}} W\, n_z\, \sigma\, ds - \int_{\Omega_e} W \left(\frac{\partial \sigma}{\partial z}\right) dx\, dy\, dz = \int_{\Omega_e} \mathbf{U}_{xy} \cdot \nabla_{xy} \sigma \; dx\, dy\, dz - \int_{\partial\Omega_{e,bot}} W^-\, n_z\, \sigma\, ds - \int_{\partial\Omega_e} \hat{C}_w\, \sigma\, ds, \tag{4.13}$$

where W^- is an initial value for W taken from the element below (or a boundary condition at the bottom in the case of the bottommost element), and $\hat{C}_w$ is a numerical flux for the normal boundary flux function $\mathbf{U}_{xy} \cdot \mathbf{n}$. On the lateral boundaries $\hat{C}_w$ should be set equal to $\hat{C}_H / \xi_s - z_b$ (cf. (4.11)) in order to preserve the local mass conservation properties of our

numerical scheme. On horizontal faces it can be taken equal to the average or upwinded value of the corresponding variables.

5. SOLUTION OF THE RIEMANN PROBLEM

In this section, we will see the main advantage of our formulation for the surface elevation. Since we were able to modify the discrete form of the primitive continuity equation to transform the boundary integrals into 3D form, we end up with well-posed Riemann problems on the inter-element boundaries. This allows us to tap into an extensive literature on approximate Riemann solvers (see, e.g., Ref. [15]). Here, we demonstrate how to compute nonlinear boundary fluxes using Roe's solver [14]. Compared to other Riemann solvers we tested [1] (Lax–Friedrichs, HLL), this solver provided the sharpest resolution of discontinuities, and its well-documented limitations with regard to resolution of certain rarefaction waves did not appear to cause any problems in any of our test cases.

Let $\mathbf{x}$ be a point on Γ, where Γ is an interior lateral boundary face in our 3D mesh. Let $\mathbf{n} = (n_x, n_y, n_z)$ be a unit normal to Γ at $\mathbf{x}$. We denote $\mathbf{c} = (h, u, v)^T$ the vector of state variables (we do not include w in $\mathbf{c}$ because it enters the normal fluxes only multiplied by n_z, and $n_z = 0$ for lateral boundary faces). Then, we define the left and right states $\mathbf{c}_\mathrm{L}, \mathbf{c}_\mathrm{R}$ at $\mathbf{x}$ as follows:

$$\mathbf{c}_\mathrm{L} = \lim_{\varepsilon \to 0-} \mathbf{c}(\mathbf{x} + \varepsilon \mathbf{n}), \qquad \mathbf{c}_\mathrm{R} = \lim_{\varepsilon \to 0+} \mathbf{c}(\mathbf{x} + \varepsilon \mathbf{n}). \tag{5.1}$$

Our task is to compute an approximation to nonlinear boundary flux $\mathbf{C}_n = (\mathbf{u}_{xy} h \cdot \mathbf{n}, \mathbf{c}_u \cdot \mathbf{n}, \mathbf{c}_v \cdot \mathbf{n})$ at $\mathbf{x}$.

In Roe's solver, an approximation to the normal boundary flux is given by

$$\hat{\mathbf{C}}_n(\mathbf{c}_\mathrm{L}, \mathbf{c}_\mathrm{R}) = \mathbf{C}_n(\mathbf{c}_\mathrm{L}) + \sum_{i=1}^{3} \alpha_i \hat{\lambda}_i^- \hat{\mathbf{r}}_i, \tag{5.2}$$

where $\hat{\lambda}_i$ are the eigenvalues and $\hat{\mathbf{r}}_i$ the corresponding eigenvectors of a constant-coefficient matrix $\hat{R}_n(\mathbf{c}_\mathrm{L}, \mathbf{c}_\mathrm{R})$ defined below, $x^- = \min\{0, x\}$, and α_i are calculated from

$$\sum_{i=1}^{3} \alpha_i \hat{\mathbf{r}}_i = \mathbf{c}_\mathrm{R} - \mathbf{c}_\mathrm{L}. \tag{5.3}$$

The matrix $\hat{R}_n(\mathbf{c}_\mathrm{L}, \mathbf{c}_\mathrm{R})$ has to satisfy the following three conditions [14]:

(i) $\hat{R}_n(\mathbf{c}_L, \mathbf{c}_R)(\mathbf{c}_R - \mathbf{c}_L) = \mathbf{C}_n(\mathbf{c}_R) - \mathbf{C}_n(\mathbf{c}_L)$;
(ii) $\hat{R}_n(\mathbf{c}_L, \mathbf{c}_R)$ is diagonalizable with real eigenvalues;
(iii) $\hat{R}_n(\mathbf{c}_L, \mathbf{c}_R) \to C'_n(\mathbf{c})$ smoothly as $\mathbf{c}_L, \mathbf{c}_R \to \mathbf{c}$, where

$$C'_n(\mathbf{c}) = \begin{pmatrix} un_x + vn_y & hn_x & hn_y \\ gn_x & 2un_x + vn_y & un_y \\ gn_y & vn_x & un_x + 2vn_y \end{pmatrix}. \tag{5.4}$$

We claim that setting $\hat{R}_n(\mathbf{c}_L, \mathbf{c}_R)$ equal to $C'_n(\bar{\mathbf{c}})$, where $\bar{\mathbf{c}} = 1/2(\mathbf{c}_L + \mathbf{c}_R)$, satisfies conditions on $\hat{R}_n$. Indeed, we obtain the following eigenvalues and corresponding eigenvectors for $C'_n(\mathbf{c})$:

$$\begin{aligned} \lambda_1(\mathbf{c}) &= \tfrac{3}{2}u_n - \tfrac{1}{2}a, & \mathbf{r}_1(\mathbf{c}) &= \begin{pmatrix} h \\ u - \frac{n_x}{2}(u_n + a) \\ v - \frac{n_y}{2}(u_n + a) \end{pmatrix}; \\ \lambda_2(\mathbf{c}) &= \tfrac{3}{2}u_n, & \mathbf{r}_2(\mathbf{c}) &= \begin{pmatrix} 0 \\ -n_y \\ n_x \end{pmatrix}; \\ \lambda_3(\mathbf{c}) &= \tfrac{3}{2}u_n + \tfrac{1}{2}a, & \mathbf{r}_3(\mathbf{c}) &= \begin{pmatrix} h \\ u - \frac{n_x}{2}(u_n - a) \\ v - \frac{n_y}{2}(u_n - a) \end{pmatrix}; \end{aligned} \tag{5.5}$$

where $u_n = un_x + vn_y$ and $a = \sqrt{u_n^2 + 4gh}$. Therefore, condition (ii) is satisfied. Clearly, $C'_n(\bar{\mathbf{c}}) \to C'_n(\mathbf{c})$ smoothly as $\mathbf{c}_L, \mathbf{c}_R \to \mathbf{c}$. Finally, the first condition can be verified by simply substituting the appropriate values in (i).

6. NUMERICAL RESULTS

Now, we will present some preliminary numerical results obtained with our 3D solver. In the first test, we use a problem with a known smooth analytical solution to obtain convergence rates. Next, we compute a steady-state solution to a supercritical flow problem to demonstrate stability of the method in a case involving discontinuities. Our last example is a standard tidal flow simulation on a domain with complex boundary and bathymetry. In all runs, the time stepping was performed using explicit TVD Runge–Kutta schemes described in Ref. [6] of the order matching the order of the space discretization. The mesh smoothing, mentioned in Sec. 3.2, was carried out by simply averaging the values of the elevation from the elements incident to each node and then connecting the nodes in the surface layer with a piecewise linear globally continuous surface mesh. The

Riemann problems on the lateral boundary faces were handled by Roe's solver and those on horizontal faces by an upwind flux.

6.1. Analytical Test Problem with a Forcing Term

In this section, we will show the results of convergence studies for a test problem with a smooth analytical solution. Since designing a non-trivial function satisfying our nonlinear system and boundary conditions at the same time is not a trivial task, we chose a slightly different approach. We had the true solution satisfy the continuity equation and solved a Dirichlet problem for a modified system that included a forcing term.

The 3D solver was tested on the following analytical test example:

$$
\begin{aligned}
\xi(t,x,y) &= e\,(\sin(\omega(x+t))+\sin(\omega(y+t)))\,,\\
u(t,x,y,z) &= d\,(z-z_{\mathrm{b}})\ \sin(\omega(x+t)),\\
v(t,x,y,z) &= d\,(z-z_{\mathrm{b}})\ \sin(\omega(y+t)),\\
w(t,x,y,z) &= d\,(z-z_{\mathrm{b}})\left(\frac{\partial z_{\mathrm{b}}}{\partial x}\sin(\omega(x+t))+\frac{\partial z_{\mathrm{b}}}{\partial y}\sin(\omega(y+t))\right)\\
&\quad -\tfrac{1}{2}\ d\ \omega\,(z-z_{\mathrm{b}})^2\ \ (\cos(\omega(x+t))+\cos(\omega(y+t)))\,,
\end{aligned}
$$

where $e=0.01$, $d=0.1$, and $\omega=0.01$. $\xi(t,x,y)$ and $z_{\mathrm{b}}(x,y)$ are the values of the z coordinate at the surface and bottom, respectively, and (u,v,w) is the velocity vector. The 3D domain Ω is a $[0,100]\times[0,100]$ square with a dynamic free surface and the bathymetry varying linearly as $z_{\mathrm{b}}(x,y)=-(2-0.005(x+y))$.

The problem was solved for $t\in[0,T]$ with $T=0.001$ days using 64-bit floating point arithmetic. In Tables I–III below, we list the L^2 norm of the error at time T for the surface elevation and velocity. We denote by Ne the number of surface triangles, Nl the maximum number of vertical layers, Δt the time step in seconds, Er$(\cdot)$ the norm of the error, and $C(\cdot)$ the convergence rate. The second-order diffusion terms were set equal to zero for these test runs.

The convergence rates we observed for the surface elevation and horizontal velocity components were similar to those obtained with the LDG method for linear problems. ($\Delta x^{k+\frac{1}{2}}$ for approximations with complete polynomials of order k on irregular meshes.) However, we have to emphasize that we are dealing here with a much more involved situation than a linear hyperbolic problem. For example, note a slight degradation of the convergence order for the surface elevation in the piecewise quadratic case. The probable reason for this phenomenon is the meshing error at the free

Table I. Piecewise Constant Approximation, Forward Euler

N	Ne	Nl	Er(ξ)	$C(\xi)$	Er(u)	$C(u)$	Er(v)	C(v)	Er(w)	$C(w)$	Δt
1	2	1	8.69×10^{-1}		5.20×10^{-1}		5.20×10^{-1}		5.31×10^{-2}		16
2	8	2	4.97×10^{-1}	0.81	7.99×10^{-1}	−0.62	7.99×10^{-1}	−0.62	4.52×10^{-2}	0.23	8
3	32	4	2.82×10^{-1}	0.82	4.08×10^{-1}	0.97	4.08×10^{-1}	0.97	2.60×10^{-2}	0.80	4
4	128	8	1.47×10^{-1}	0.94	2.72×10^{-1}	0.58	2.72×10^{-1}	0.58	1.44×10^{-2}	0.85	2
5	512	16	7.362×10^{-2}	1.00	1.73×10^{-1}	0.65	1.73×10^{-1}	0.65	8.23×10^{-3}	0.81	1
6	2048	32	3.67×10^{-2}	1.00	1.11×10^{-1}	0.64	1.11×10^{-1}	0.64	4.77×10^{-3}	0.79	0.5

Table II. Piecewise Linear Approximation, Two-Stage Runge–Kutta

N	Ne	Nl	Er(ξ)	C(ξ)	Er(u)	$C(u)$	Er(v)	$C(v)$	Er(w)	$C(w)$	Δt
1	2	1	3.36×10^{-1}		5.28×10^{-1}		5.49×10^{-1}		3.61×10^{-2}		8
2	8	2	1.11×10^{-1}	1.60	2.04×10^{-1}	1.37	2.07×10^{-1}	1.41	1.24×10^{-2}	1.54	4
3	32	4	2.98×10^{-2}	1.90	7.53×10^{-2}	1.44	7.58×10^{-2}	1.45	4.40×10^{-3}	1.49	2
4	128	8	7.63×10^{-3}	1.97	2.39×10^{-2}	1.66	2.40×10^{-2}	1.66	1.72×10^{-3}	1.36	1
5	512	16	1.94×10^{-3}	1.98	7.27×10^{-3}	1.72	7.29×10^{-3}	1.72	7.49×10^{-4}	1.20	0.5
6	2048	32	4.92×10^{-4}	1.98	2.15×10^{-3}	1.76	2.16×10^{-3}	1.76	3.27×10^{-4}	1.20	0.25

surface boundary, where the geometry was approximated by a piecewise linear (globally continuous) surface mesh generated by our smoothing algorithm.

In the case of the vertical velocity component the situation is even more complicated because w is computed from the numerical approximations for u and v using continuity Eq. (4.13). Therefore, it should not come as a surprise that w converges at a slower rate than u and v. Another important thing to note in this context is the fact that the ratio of vertical to horizontal velocity components corresponds to the ratio of vertical to horizontal length scales (this is one of the fundamental properties of the shallow water model). The same property is also seen for the mag-

Table III. Piecewise Quadratic Approximation, Three-Stage Runge–Kutta.

N	Ne	Nl	Er(ξ)	$C(\xi)$	Er(u)	$C(u)$	Er(v)	C(v)	Er(w)	C(w)	Δt
1	2	1	1.57×10^{-2}		1.34×10^{0}		1.34×10^{0}		4.35×10^{-2}		2
2	8	2	3.42×10^{-3}	2.20	2.18×10^{-1}	2.62	2.18×10^{-1}	2.62	1.18×10^{-2}	1.88	1
3	32	4	4.61×10^{-4}	2.89	2.80×10^{-2}	2.96	2.80×10^{-2}	2.96	2.28×10^{-3}	2.37	0.5
4	128	8	6.96×10^{-5}	2.73	3.65×10^{-3}	2.94	3.65×10^{-3}	2.94	5.57×10^{-4}	2.03	0.25
5	512	16	1.30×10^{-5}	2.42	4.90×10^{-4}	2.90	4.90×10^{-4}	2.90	1.73×10^{-4}	1.69	0.125

nitude of the error for the corresponding terms. Thus, the error for w is in absolute terms much lower than the error for u and v. Hence, at realistic grid resolutions and domain aspect ratios, a somewhat slower convergence for w should not degrade the accuracy of the scheme for other variables.

6.2. Supercritical Flow Problem

In this section, we test our numerical model on a 2D problem with a known analytical solution and compare results from 2D and 3D simulations. Supercritical channel flows subject to a change in cross-section can lead to formation of shock and rarefaction waves. Here, we take one particular configuration given in Zienkiewicz and Ortiz [18]. We are given a channel of uniform depth whose boundary wall is constricted on both sides with a constriction angle of 5° resulting in a cross-wave pattern. Flow is induced through the left boundary with no flow through the top and bottom boundaries. The inlet Froude number is defined by $Fr = U/\sqrt{gH}$, where U and H are the mean velocity and total depth of the fluid at the inlet, respectively. When $Fr > 1$, the flow is said to be supercritical. For our test, we chose $Fr = 2.5$. This problem can be solved analytically using methods presented by Ippen in Ref. [9]. The discontinuous true solution projected on our (smoothed) mesh is demonstrated in Fig. 1.

The coarse 2D mesh consists of 3155 triangular elements with no particular orientation. We project these elements vertically to obtain a 3D grid with prismatic elements. The 3D mesh can be refined horizontally by subdividing every triangle in four using edge bisection, or/and vertically by subdividing into layers. In our test runs, we set vertical and horizontal eddy viscosities equal to zero and used a free-slip boundary condition at the bottom.

As we can see in Figs. 2 and 3, the numerical solutions for this problem, certainly very challenging for traditional methods, show no instability. This fact is especially interesting in the case of piecewise linear approximation because no slope limiters or any other stabilization techniques were employed (or needed) in the course of the simulation. We also verified that increasing the number of layers does not affect the solution of this problems substantially (cf. Table IV) just as expected in case of an essentially 2D problem.

In Table IV, we demonstrate the L^2 norm of the error in surface elevation for numerical solutions computed with different meshes and approximation spaces. For comparison, we also list the errors for solutions obtained using our 2D LDG simulator.

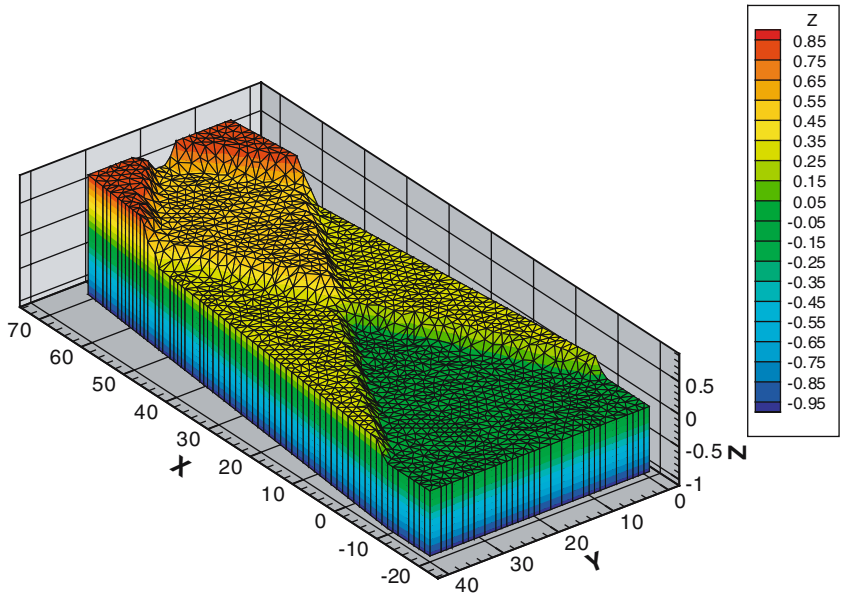

Fig. 1. Constricted channel flow problem: true solution, one layer.

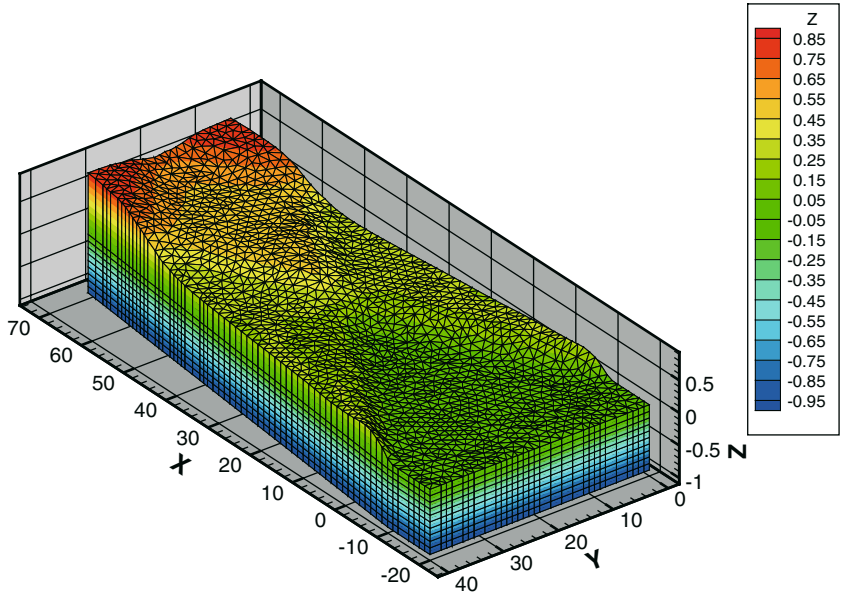

Fig. 2. Constricted channel flow problem: piecewise constants, 10 layers.

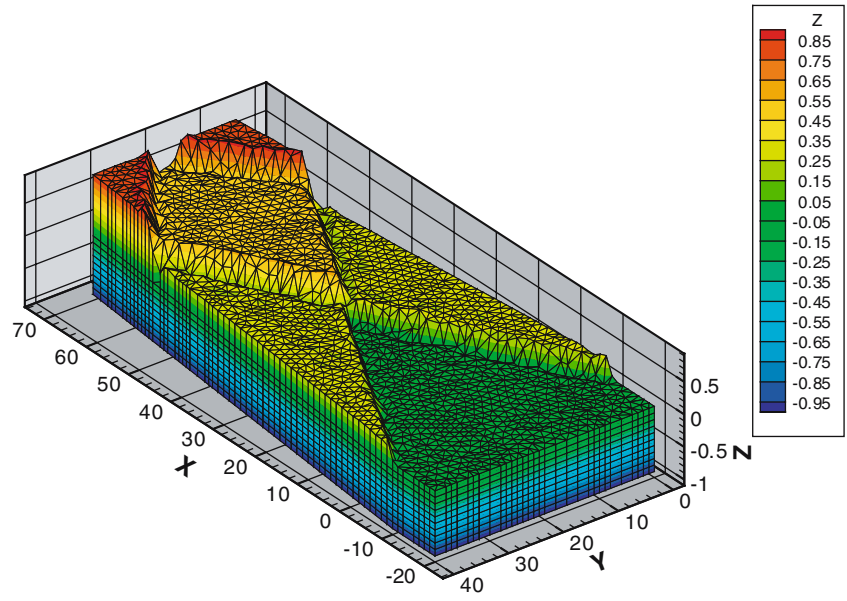

Fig. 3. Constricted channel flow problem: piecewise linears, 10 layers.

6.3. Tidal Flow near Bahamas Islands

In this test run, we simulate tide-driven flow near the Bahamas Islands. The domain geometry and the coarse finite element mesh consisting of 1696 elements are shown in Fig. 4. Finer meshes were obtained by subdividing each triangle into four using edge bisection (we refer to this mesh as "refined") and by subdividing into vertical layers. The bathymetry is shown in Fig. 5.

Table IV. Error in Surface Elevation for Supercritical Flow Problem

Model type	Coarse mesh, constants, one layer	Coarse mesh, constants, 10 layers	Refined twice mesh, constants, one layer	Coarse mesh, linears, one layer	Coarse mesh, linears, 10 layers
3D	2.85	2.85	1.74	1.53	1.57
2D	2.75	N/A	1.61	1.42	N/A

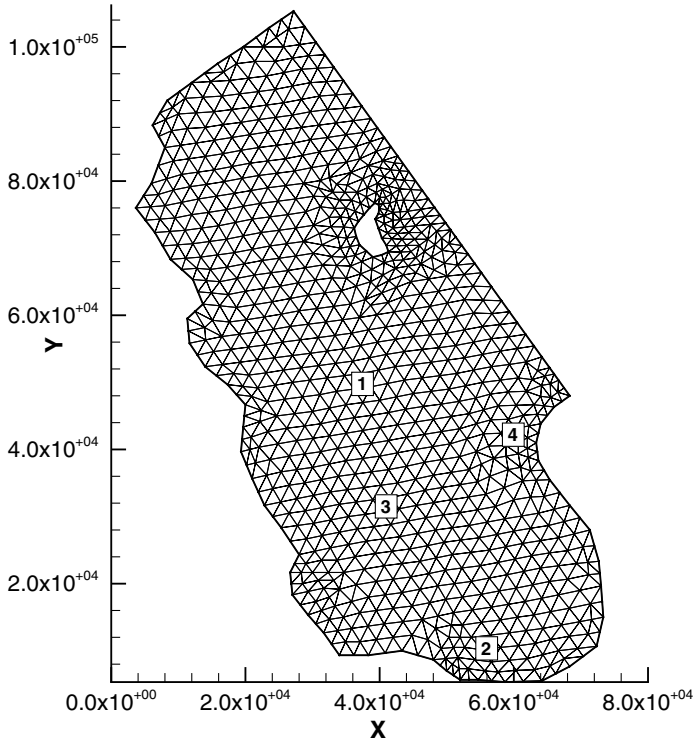

Fig. 4. Bathymetry for tide-driven flow past an island (Bahamas). Lengths are in meters.

The following tidal forcing with time (t) in hours was imposed at the open sea boundary:

$$\begin{aligned}\hat{\xi}(t) = {} & 0.075\cos\left(\frac{t}{25.82}+3.40\right)\\ & +0.095\cos\left(\frac{t}{23.94}+3.60\right)\\ & +0.100\cos\left(\frac{t}{12.66}+5.93\right)\\ & +0.395\cos\left(\frac{t}{12.42}+0.00\right)\\ & +0.060\cos\left(\frac{t}{12.00}+0.75\right)\quad \text{(meters)}. \end{aligned} \tag{6.1}$$

The simulations were cold-started and the tidal forcing was imposed gradually over a period of 2 days. The Coriolis parameter was set to $3.19 \times 10^{-5}\text{s}^{-1}$. The horizontal eddy viscosity was set equal to zero, the vertical

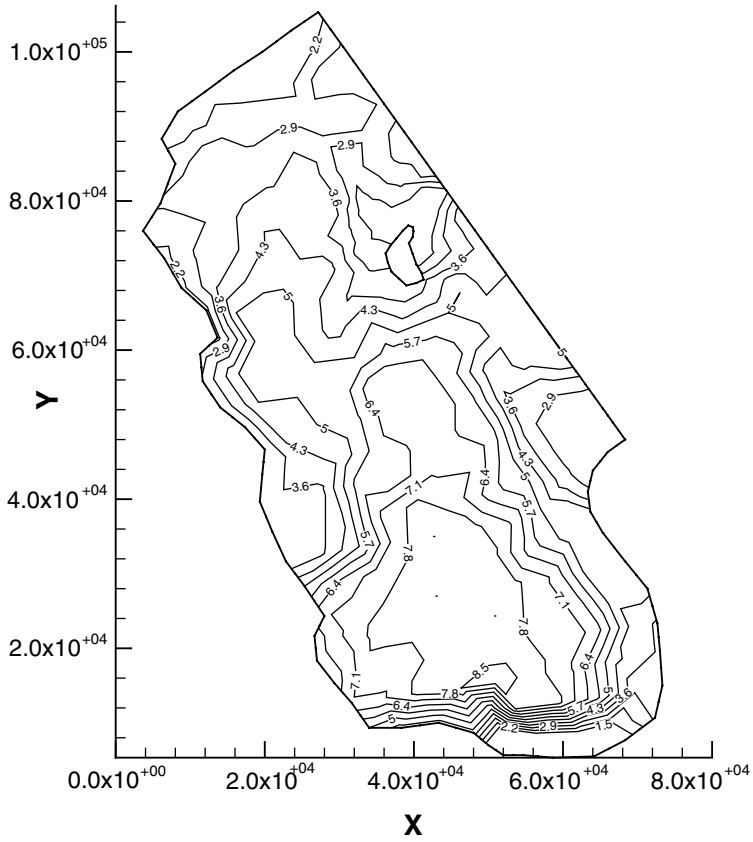

Fig. 5. Mesh for tide-driven flow past an island (Bahamas). Lengths are in meters.

eddy viscosity was modeled using a quadratic formula given by Davies in Ref. [8]: $v_t = K_t \frac{(\bar{u}^2+\bar{v}^2)^{\frac{1}{2}}}{\omega_a}$, where $\bar{u}, \bar{v}$ are depth averaged horizontal velocity components, ω_a a typical long wave frequency taken as $10^{-4}s^{-1}$, and $K_t = 2 \times 10^{-5}$ a dimensionless coefficient.

The elevation and velocities were monitored at four different stations, whose coordinates in meters are (38666.66, 49333.32), (56097.79, 9612.94), (41262.60, 29775.73), and (59594.66, 41149.62), for the duration of days 11 and 12 of the simulation at an increment of 600 s. A constant time step of 20 s was used in the runs performed on the coarse mesh, and a time step of 10 s was taken for the refined mesh.

In Ref. [2], we simulated this tidal flow problem in 2D using UT-BEST, our LDG solver for the depth-integrated shallow water equations,

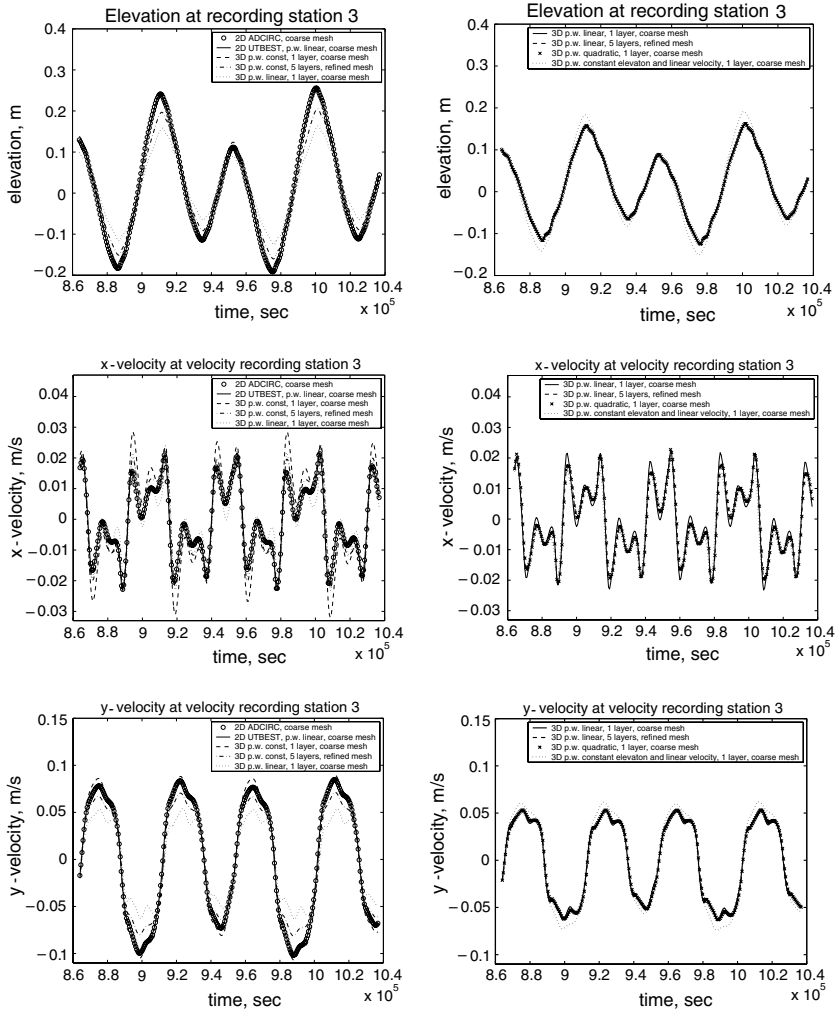

Fig. 6. Elevation and velocity approximations at Bahamas station 3 for days 11–12.

and compared the results to the 2D ADCIRC model of Luettich *et al.* [12]. This time, we undertook a number of numerical experiments with our 3D solver utilizing different approximation spaces and meshes. In Figs. 6 and 7, we present the surface elevation and horizontal velocity time series for the 3D LDG model together with those of ADCIRC and UTBEST.

For readability, we plotted data for each recording station on two separate graphs, the first one containing the results from both 2D models, the 3D LDG model with piecewise constants on various meshes, and the 3D

LDG with piecewise linears on the coarse mesh. The second graph compares several higher order 3D LDG solutions. For sake of conciseness, we only included output from recording stations 3 and 4 noting that the other two stations displayed good agreement similar to station 3.

Commenting on the results of these numerical experiments, we first want to note that the 3D model is very sensitive to the choice of vertical eddy viscosity closure, whereas the 2D models are just as sensitive to the bottom friction formulation. Nevertheless, the results of 2- and 3D models displayed an excellent agreement in phase and a good agreement in amplitude — the last one with an exception of the anomaly at recording station 4 (see middle left graph in Fig. 7). The reason for this anomaly at velocity recording station 4 is not quite clear, although we have to note that ADCIRC and UTBEST solutions, which virtually lie on top of each other in all other plots, also show some disagreement in amplitude in this particular graph. However, the agreement between higher order 3D solutions is very good leading us to conclude that the piecewise linear solution on the coarse mesh is sufficiently close to the converged result. This type of behavior was also seen in our 2D test runs, where the graphs for piecewise quadratic solutions were almost indistinguishable from those for piecewise linear approximations. The results of the simulation that employs piecewise constants for elevation and piecewise linear polynomials for velocity are also very close to the fully linear solution. This fact is very encouraging since we want to have an option of using piecewise constants for problems with "rough" surface elevation.

The graphs for piecewise constant polynomial spaces merit a closer look. Though rather similar to the piecewise linear solution in most cases, they can be a poor approximation at some points in space and time, and even refining the mesh does not guarantee convergence to the linear solution at every point (compare the middle left graph in Fig. 7). The theory of LDG methods for linear convection–diffusion problems states convergence of order Δx^k for general triangulations, where $k \geqslant 1$ is the order of the polynomial space [5]. Thus, there is no guarantee of convergence for piecewise constant approximations. Based on the results of our numerical experiments for the piecewise constant approximation, we can conclude that our model displays behavior in agreement with the linear theory.

7. CONCLUSIONS

In this paper, we have described an implementation of the LDG method for the 3D shallow water equations, and given some preliminary numerical results. The methodology shows promise for modeling both high Froude number and tidal flows. More extensive testing of this

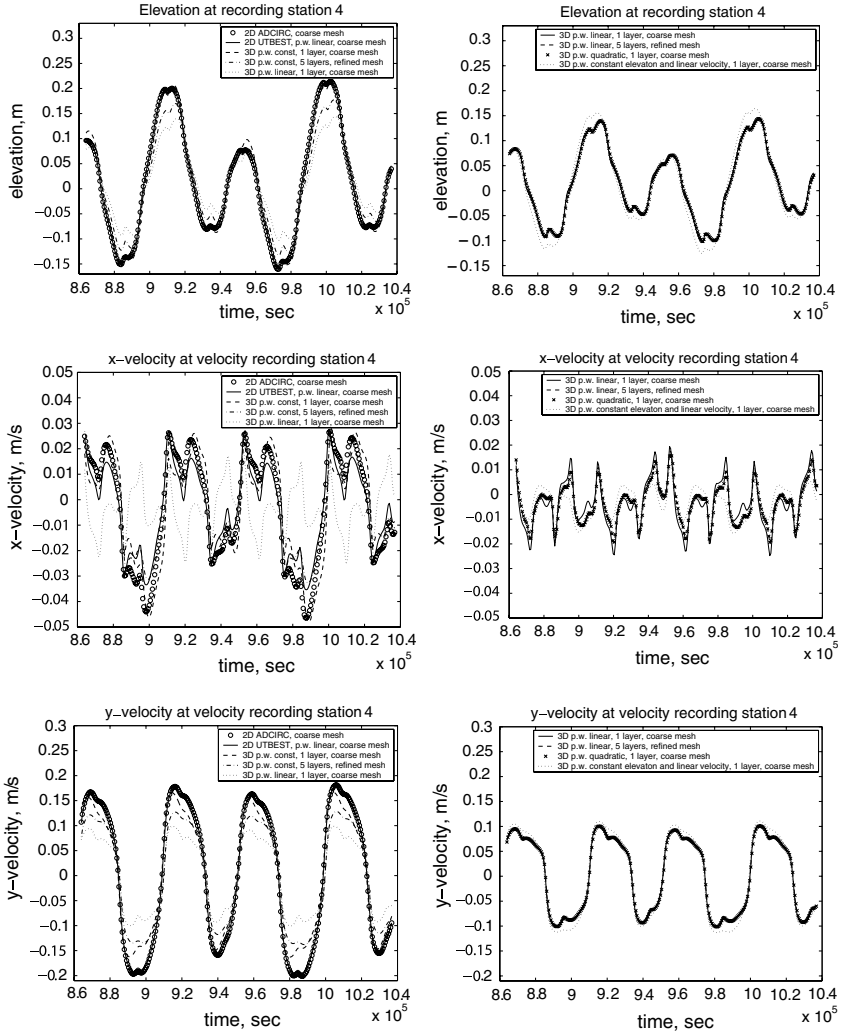

Fig. 7. Elevation and velocity approximations at Bahamas station 4 for days 11–12.

algorithm and analysis of its convergence properties is underway and will be reported in future papers.

REFERENCES

1. Aizinger, A. (2004). *A Discontinuous Galerkin Method for Two- and Three-Dimensional Shallow-Water Equations*, Ph.D. Dissertation, University of Texas at Austin.

2. Aizinger, A., and Dawson, C. (2002). A discontinuous Galerkin method for two-dimensional flow and transport in shallow water. *Adv. Water Resour.* **25**, 67–84.
3. Casulli, V., and Walters, R. A. (2000). An unstructured grid, three- dimensional model based on the shallow water equations. *Int. J. Numer. Meth. Fluids* **32**, 331–348.
4. Chippada, S., Dawson, C. N., Martinez, M., and Wheeler, M. F. (1998). A Godunov-type finite volume method for the system of shallow water equations. *Comput. Meth. Appl. Mech. Eng.* **151**, 105–129.
5. Cockburn, B., and Shu, C.-W. (1998). The local discontinuous Galerkin finite element method for convection-diffusion systems. *SIAM J. Numer. Anal.* **35**, 2440–2463.
6. Cockburn, B., and Shu, C.-W. (1989). TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework. *Math. Comput.* **52**, 411–435.
7. Cockburn, B., Karniadakis G., and Shu, C.-W. (2000). The development of discontinuous Galerkin methods. In Cockburn, B. Karniadakis, G. and Shu, C.-W. (eds.), *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Lecture Notes in Computational Science and Engineering, volume 11, Part I: Overview, Springer-Verlag, Berlin, pp. 3–50.
8. Davies, A. M. (1986). A three-dimensional model of the Northwest European continental shelf, with application to the M4 tide. *J. Phys. Oceanogr.* **16**(5), 797–813.
9. Ippen, A. T. (1951). Mechanics of supercritical flow. *Trans. ASCE*, **116**, 268–295.
10. Johnson, B. H., Kim, K. W., Heath, R. E., Hsieh, B. B. and Butler, H. L. (1991). *Development and verification of a three-dimensional numerical hydrodynamic, salinity and temperature model of Chesapeake Bay*, Technical report HL-91-7, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.
11. Le Veque, R. J. (1992). *Numerical Methods for Conservation Laws*, Basel, Birkhäuser.
12. Luettich, R. A., Westerink, J. J., and Scheffner, N. W. (1991). ADCIRC: *An Advanced Three-Dimensional Circulation Model for Shelves, Coasts and Estuaries*, Report 1, U.S. Army Corps of Engineers, Washington, D.C. 20314-1000.
13. Lynch, D. R., and Werner, F. E. (1991). Three-dimensional hydrodynamics on finite elements. Part II: Non-linear time-stepping model. *Int. J. Numer. Meth. Fluids* **12**, 507–533.
14. Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **43**, 357–372.
15. Toro, E. F. (2001). *Shock-Capturing Methods for Free-Surface Shallow Flows*, Wiley, Chichester.
16. Vreugdenhil, C. B. (1994). *Numerical Methods for Shallow-Water Flow*, Kluwer Academic Publishers, Dordrecht.
17. Weiyan, T. (1992). *Shallow Water Hydrodynamics*, Elsevier Oceanography Series, 55, Elsevier, Amsterdam.
18. Zienkiewicz, O. C., and Ortiz, P. (1995). A split-characteristic based finite element model for the shallow water equations. *Int. J. Numer. Meth. Fluids* **20**, 1061–1080.