ERDC/CHL TR-22-9

US Army Corps
of Engineers®
Engineer Research and
Development Center

Foundational Principles in the Development of AdH-SW3, the Three-Dimensional Shallow Water Hydrodynamics and Transport Module within the Adaptive Hydraulics/Hydrology Model

R. C. Berger

June 2022

Coastal and Hydraulics Laboratory

The US Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdc.usace.army.mil.

To search for other technical reports published by ERDC, visit the ERDC online library at https://erdclibrary.on.worldcat.org/discovery.

ERDC/CHL TR-22-9
June 2022

Foundational Principles in the Development of AdH-SW3, the Three-Dimensional Shallow Water Hydrodynamics and Transport Module within the Adaptive Hydraulics/Hydrology Model

R. C. Berger

Coastal and Hydraulics Laboratory
US Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Final report

Prepared for US Army Corps of Engineers, Engineer Research and Development Center
Vicksburg, MS 39180-6199

Under MIPR W74RDV12944824

Abstract

This report details the design and development of the three-dimensional shallow water hydrodynamics formulation within the Adaptive Hydraulics/Hydrology model (AdH-SW3) for simulation of flow and transport in rivers, estuaries, reservoirs, and other similar hydrologic environments. The report is intended to communicate principles of the model design for the interested and diligent user. The design relies upon several layers of consistency to produce a stable, accurate, and conservative model. The mesh design can handle rapid changes in bathymetry (e.g., steep-sided navigation channels in estuaries) and maintain accuracy in density-driven transport phenomena (e.g., thermal, or saline stratification and intrusion of salinity).

Contents

Figures and Tables

Figures

Tables

Preface

This report was created for the US Army Corps of Engineers, Research and Development Center (ERDC), under MIPR W74RDV12944824. The report was funded through the Model Maintenance program, specifically for the Adaptive Hydraulics/Hydrology (AdH) model system.

At the time of publication of this report, Mr. David P. May was chief, River and Estuarine Engineering Branch, ERDC Coastal and Hydraulics Laboratory (CHL); Dr. Cary A. Talbot was chief, Flood and Storm Protection Division (ERDC-CHL). The deputy director of ERDC-CHL was Mr. Keith Flowers, and the director was Dr. Ty V. Wamsley.

Dr. Berger designed and created the nucleus of the AdH-SW3. The columnar data and refinement structures were created by Dr. Jeff Hensley of the ERDC Information Technology Laboratory (ITL). Ms. Jennifer T. McAlpin (ERDC-CHL) managed the two-dimensional and three-dimensional shallow water model modules of AdH (AdH-SW2 and AdH-SW3, respectively). Dr. Gaurav Savant (ERDC-CHL) did much of the testing of the model and added the turbulence library. Dr. Corey Trahan (ERDC-ITL) subsequently has added many additional forces such as wave stresses.

The Commander of ERDC was COL Teresa A. Schlosser, and the Director was Dr. David W. Pittman.

1 Introduction

1.1 Background

Performance of river, reservoir and estuary projects are typically solved using shallow water-type models. This type of modeling approach is relevant to open channel problems in which vertical inertia is small. This is often the case in rivers, reservoirs, and estuaries as well as other areas such as coasts and oceans. These project evaluations typically require long-term evaluations, frequently years or multi-year simulations. Shallow water equations are most suited to these problems. Another difficulty with many of these simulations is that small features such as shocks, plumes, and local erosion/sedimentation require fine resolution of the mesh in specific places. These sharp features may be transient and therefore local high resolution may not be required for the entire simulation. The Adaptive Hydraulics/Hydrology (AdH) framework adds and removes resolution throughout the simulation. This makes it possible to make lengthy simulations that are accurate.

The AdH model was originally developed as a groundwater flow and constituent transport simulator by J. H. Schmidt in the 1990s (Stagg et al. 2000). The model uses an unstructured mesh that automatically adds and removes refinement. The design of the code is such that it is straightforward to add other physical problems to its capabilities. The author added a Navier-Stokes solver soon after the original model was built and later created the two-dimensional (2D) shallow water module (AdH-SW2).

Many projects can be evaluated in 2D. However, typical problems in estuaries and reservoirs cannot. The presence of density currents means that the velocity variation over depth is large. The currents at times flow in opposite directions from near bed to surface. Additionally, even in river systems the presence of bends, channels and other features means that the 2D assumptions about the current variation over depth is misleading. A three-dimensional (3D) shallow water model is then needed. The need for a 3D shallow water module for AdH is apparent.

AdH-SW3 was developed from approximately 2005 to 2009. The report

draft followed within 2 yr[1]. The author retired about that time, and the report draft was not published. Subsequently, journal articles with some of this information have been published. When the author returned to the US Army Engineer Research and Development Center in emeritus status, the draft was revived and clarifications made. The chronology of the AdH-SW3 publications is somewhat out of order. This report is the authoritative document that most accurately describes the original thoughts, decisions, and development of AdH-SW3. Details concerning drivers, specific boundary conditions, and coefficients will be contained in later documentation reports. This report concerns major model design principles.

1.2 Objective(s)

This report is designed to explain the logic behind the development of the AdH 3D shallow water module, referred to as AdH-SW3. This report focuses on the hydrodynamics and constituent transport. Subsequent reports will detail other aspects of AdH-SW3 development.

1.3 Approach

The report is written with the curious model user in mind. Computational methods expertise is not required to understand the basic ideas in AdH-SW3. The mathematical model and the form of the numerical approach are stated first. The statement of all the equations up front with little explanation helps the reader to see the larger picture of what is being solved and is helpful for easy reference. This is followed by a long explanation of all the key properties, and the decisions are discussed. Often the descriptions are related to physical behaviors and visual displays to make what can be difficult mathematics more understandable. In this manner, the diligent user will be able to understand and perhaps to suggest new and better approaches. The report makes a point to identify places where other decisions could have been made and to describe alternative approaches that could be investigated.

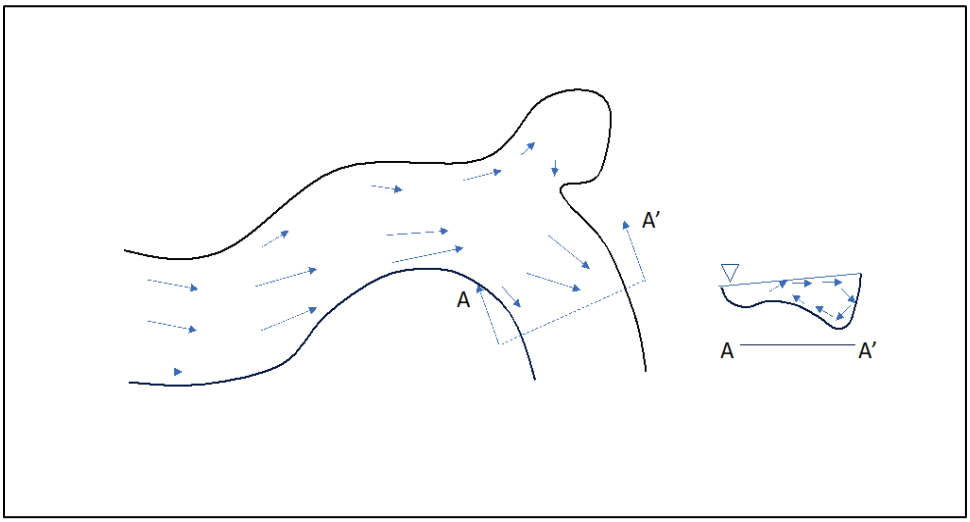
[1] For a full list of the spelled-out forms of the units of measure used in this document, please refer to *US Government Publishing Office Style Manual*, 31st ed. (Washington, DC: US Government Publishing Office 2016), 248-52, https://www.govinfo.gov/content/pkg/GPO-STYLEMANUAL-2016/pdf/GPO-STYLEMANUAL-2016.pdf.

2 Motivating Physical Systems

The shallow water equations, a simplified form of the Navier-Stokes equations, include only the conservation of mass and the horizontal momentum equations, subject to the hydrostatic assumption that the water pressure at any point is determined completely by the weight of the water above it. Shallow water equations assume that vertical inertia is small enough to be neglected. The vertical velocity then must be determined from conservation of mass alone. These shallow water equations are approximated in 3D by the computational model AdH-SW3.

The target problems for a numerical model of this sort are generally shallow, open water bodies with long wave drivers that still contain important vertical variability. Shallow water models are used effectively in all dimensional varieties. A one-dimensional (1D) model makes cross-sectional averaged calculations. These 1D models are effective in that they can make long-term calculations, and in many circumstances, cross-sectional results are good enough. Often the 1D model assumes uniform velocity and water surface height over the entire cross section. This is not actually a requirement. The model developer could make other assumptions about the velocity distribution. However, the calculations are limited by this assumption. Similarly, there are 2D shallow water models. Most commonly these are depth-integrated calculations. This means that the model makes calculations of water surface and velocity in the *horizontal* plane and makes some assumption (usually a constant) about the velocity and any other constituent over depth. The 3D model is necessary when significant velocity and constituent variation occurs in the vertical as well as the other dimensions. Three common environments demonstrate the need for a 3D model. The first example is of a river. Figure 1 illustrates this case.

Figure 1. Example of a river in a bend.



The plan view shows the bend and some surface currents. The cross-section view shows the in-plane velocity circulation caused by the bend.

The portion of the figure to the left is a plan view of a river with portions of two bends. The current velocities shown are a schematic of the surface currents. The fastest currents approach a bend on the inside and then swing to the outside as they exit the bend. Consequently, the maximum bottom currents do the opposite. They are pushed to the inside downstream of the bend. There is also a small bay; this is filled and emptied in the figure. Currents flow into and back out of the bay. Depending upon the depth of the bay and its location along the bend, the fraction of the currents passing into the bay may be more surface or bottom water from the river. Cross section "A-A'" is illustrated in the right side of the figure. Note that the water surface slopes upward toward the outside of the bend. The currents projected onto the cross-section plane are also shown in this figure. The surface currents tend toward the outside of the bend and the bottom currents to the inside of the bend. Here there are variations in currents in all three dimensions.

The river example is one in which density considerations do not significantly impact the currents. The term for this is *barotropic*. The currents in many rivers are strong enough that density variations due to temperature or other constituents do not significantly modify the currents. The next two examples are of a reservoir and an estuary. In these cases, the density variation does impact the currents. When density variation impacts currents, this is termed *baroclinic*.

An example of a stratified reservoir with an inflow river water that is of a midrange temperature is shown in Figures 2 and 3.

Figure 2. Plan view of an example reservoir.

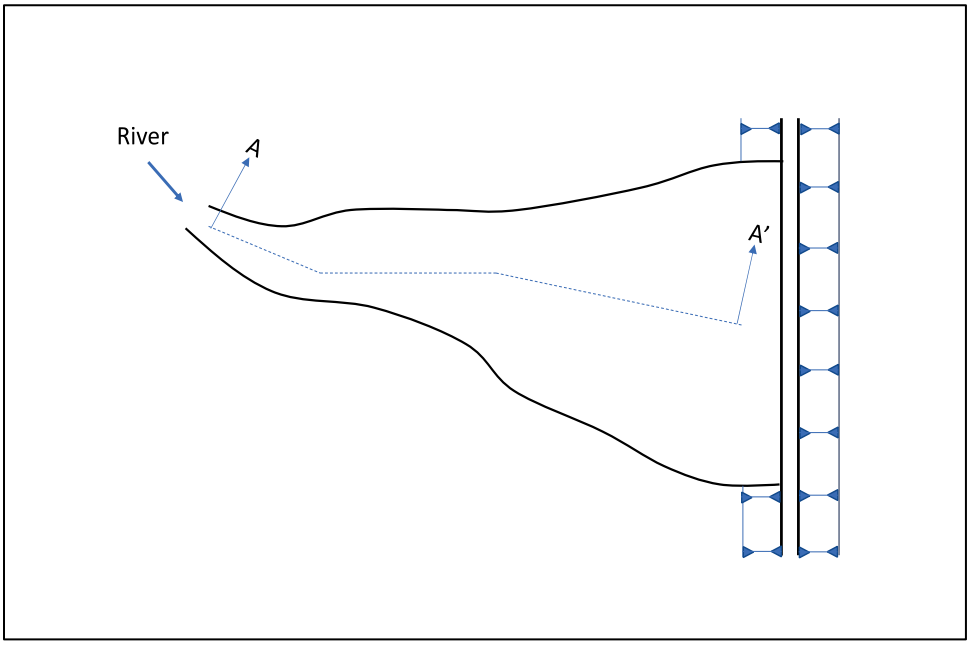


Figure 3. Elevation view of the same reservoir profile, with circulation, river inflow, and density depiction.

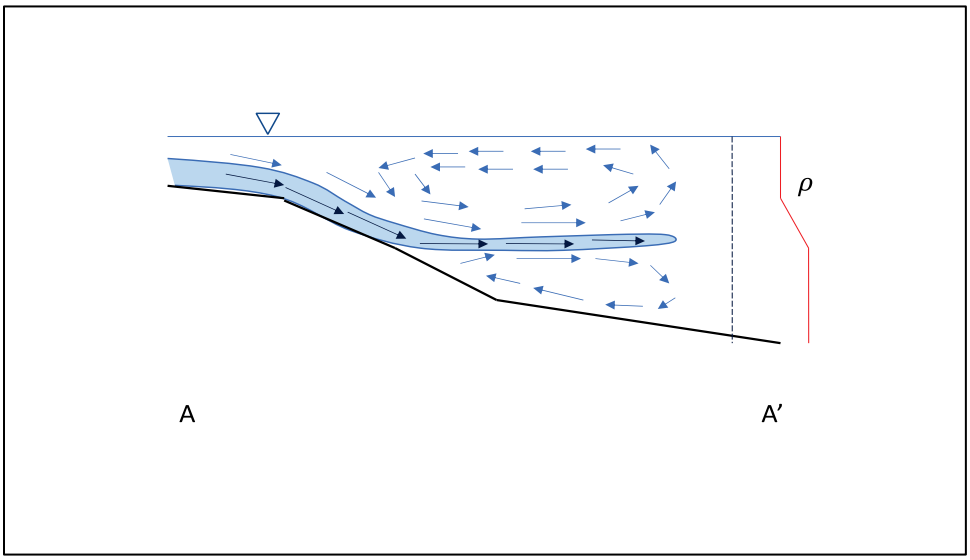
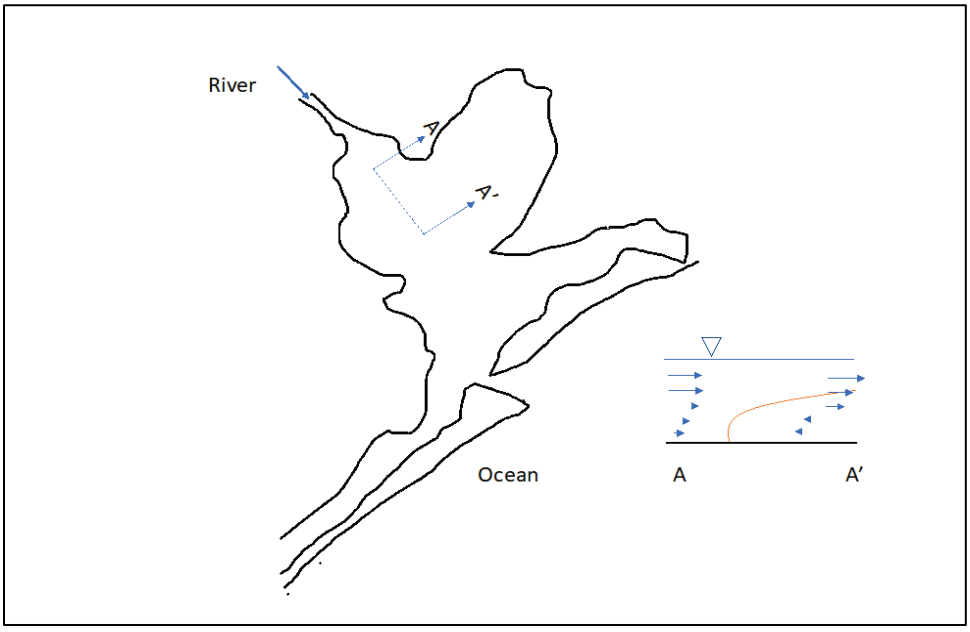


Figure 2 shows the dam on the right and the river inflow on the left. Figure 3 is of the central profile of the reservoir. The reservoir is stratified. This is illustrated by the vertical density profile shown on the right side of the figure. The reservoir is stratified because of temperature. In the

summer season, the sun warms the upper layers. The warmer water then remains on the surface. Currents in a reservoir are typically weak and are unable to overcome the temperature differences, and the reservoir becomes stratified. This is significant for more than just temperature. The bottom waters become effectively cut off from the atmosphere. Over time, the dissolved oxygen can become depleted. The figure shows the stratification and river water entering that is of a midrange temperature. That is, the river water has a temperature greater than the temperature of the lower layer and less than the surface layer of the reservoir. The river inflow then tends to plunge below the surface and follow the reservoir bed until it comes to a location where the river temperature matches the reservoir. The river plume then separates from the bed and flows along this intermediate temperature layer. The river flow causes circulation eddies in opposite directions in the upper and lower layers of the reservoir. Additionally, the circulation distribution laterally will vary across the cross section. The reservoir circulation is typically 3D.

The third example of typical applications of AdH-SW3 is that of an estuary. Figure 4 illustrates this example. The ocean area has a tide and supplies saline water to the bay. A river brings freshwater. Section A-A' shows a profile along part of the channel. Saline water halocline (steepest gradient in salinity) is illustrated by the red solid line. The vectors show the density-induced currents.

Figure 4. An example estuary, patterned after Galveston Bay, Texas.

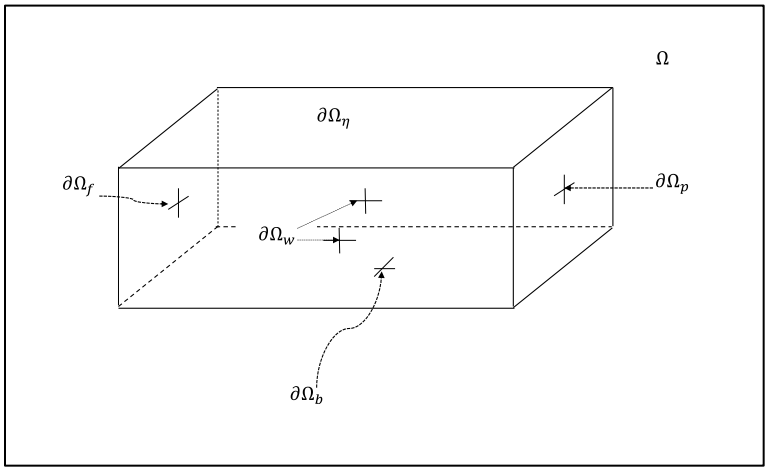


The estuary is primarily driven by ocean tide, river inflow, density currents, and winds. Salinity stratification is usually more important than thermal density variations, though AdH-SW3 represents both. The tide fills and empties the bay. The salty water from the ocean tends to force a net transport in the upstream, or flood, direction in the lower water columns and in deeper locations. The ebb, or downstream, drift is typically more significant in the upper water column and in the shallows. An example of circulation one might expect in an estuary is shown in the right side of the figure. Here, section A-A' is shown with the halocline, in red, and the circulation indicated by the vectors. These vectors show that gravity currents are strong in the flood direction in the lower water column. The upper water column has currents that are in the ebb direction. Currents also move laterally and longitudinally in the bay. The 3D currents are dominant. Modeling of estuaries and reservoirs necessarily need to simulate long periods of time. These systems react slowly, and the principal time frame of activity is approximately the water year. A 3D model that is computationally efficient is important in simulation of these flow environments.

3 Analytic Equations (Mathematical Model)

The conventional form of the shallow water equations will be defined followed by the weak form. The domain and the boundary definitions are included in Figure 5.

Figure 5. Definition of the domain.



3.1 Analytical shallow water equations

The shallow water equations are

$$\boldsymbol{\nabla} \cdot \boldsymbol{v} = 0 \tag{1}$$

$$\frac{\partial v_1}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{v} v_1) + \frac{1}{\rho_0}\frac{\partial P}{\partial x} - \dot{\omega} v_2 - \frac{1}{\rho_0}(\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_{xm}) - \tau_{x_s} = 0 \tag{2}$$

$$\frac{\partial v_2}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{v} v_2) + \frac{1}{\rho_0}\frac{\partial P}{\partial y} + \dot{\omega} v_1 - \frac{1}{\rho_0}\left(\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_{ym}\right) - \tau_{y_s} = 0 \tag{3}$$

Here, $\boldsymbol{v} = \{v_1, v_2, v_3\}$ is the velocity vector in the three coordinate directions, ρ_0 is a reference water density, $\dot{\omega}$ is the Coriolis rotation rate, P is the hydrostatic pressure, and τ_{x_s,y_s} are surface stresses (free-surface, bed, or sidewalls). Equation (1) is the continuity equation. Equations (2) and (3) are the horizontal x-direction and y-direction momentum equations. The stress terms in the momentum equations, $\boldsymbol{\sigma}_{xm}$ and $\boldsymbol{\sigma}_{ym}$ are defined as

$$\boldsymbol{\sigma}_{xm} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \end{bmatrix}, \boldsymbol{\sigma}_{ym} = \begin{bmatrix} \sigma_{yx} \\ \sigma_{yy} \\ \sigma_{yz} \end{bmatrix} \tag{4}$$

which are rows of the symmetric stress tensor.

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{5}$$

These stresses include the turbulent and molecular stresses. The convention used here is that the first subscript indicates the direction and the second subscript indicates the plane normal to that axis. The pressure, $P(z)$, is dependent only on the weight of the water above some elevation z and the atmospheric surface pressure.

$$P(z) = P_a + \int_z^\eta g\rho\,(z)dz \tag{6}$$

Here, P_a, is atmospheric pressure, and η is the water surface elevation. This relationship holds if the assumption that vertical inertia is small enough to be neglected, compared with the magnitude of the other terms. This is a big assumption that is used commonly in open channel models. The user must determine if this assumption is valid for each application. Typically, this assumption is less accurate close to hydraulic structures or in shorter wavelength conditions. AdH-SW3 allows the density to be dependent upon temperature, salinity, and, possibly, sediment concentration. It is not particularly difficult to add other constituent dependencies to the equation of state for density. AdH-SW3 allows libraries of equations of state to be produced as needed.

Boundary conditions at the surface and bed are required to find the water depth, $h(x, y, t)$. These are supplied by the kinematic boundary conditions for the free-surface and the bed.

$$\frac{\partial \eta}{\partial t} + v_{1_\eta}\frac{\partial \eta}{\partial x} + v_{2_\eta}\frac{\partial \eta}{\partial y} - v_{3_\eta} = S_\eta \tag{7}$$

$$\frac{\partial b}{\partial t} + v_{1_b}\frac{\partial b}{\partial x} + v_{2_b}\frac{\partial b}{\partial y} - v_{3_b} = S_b \tag{8}$$

Equation (7) includes the subscript η to indicate this variable is evaluated at the free-surface. Equation (8) is the same except the variable is evaluated at the bed, b. The terms S_η and S_b indicate water sources and sinks at the free-surface and the bed, respectively. At the free-surface, the source/sink is typically rainfall or evaporation. At the bed, this is usually exchange with groundwater. Leaving aside the source/sink term, the

kinematic free-surface Equation (7) says that the free-surface moves to make sure no water passes through it. In many applications, the bed kinematic condition requires that the bed is not moving and no flow passes through the bed. When the bed is moving, it is imposed by some outside calculation. Most often this is from calculations of sediment movement although an imposed bed movement could come from landslides or earthquakes, for example.

Additionally, the convection-diffusion equation for transport of constituents needs to be solved. These may be *dilute* constituents, which do not feed back into the hydrodynamic calculations, or the constituents may significantly impact the density of water. Typically, this is the case with salinity and temperature. These baroclinic constituents change the density and the pressure and can produce dramatically different circulation patterns in estuaries and reservoirs.

$$\frac{\partial c}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{v}c) - \boldsymbol{\nabla} \cdot (\boldsymbol{D}\boldsymbol{\nabla}\mathrm{c}) - S_c = 0 \tag{9}$$

Here, c is any constituent. This may be either baroclinic or not. S_c is any source/sink of this constituent. The term $\boldsymbol{D}$ is a vector of diffusion coefficients and is defined next.

$$\boldsymbol{D} = \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} \tag{10}$$

3.2 Weak form of the analytic equations

The weak forms of the preceding equations are created by integration against some sort of *test function*. It is the basis for finite element and finite volume modeling and is often included in the description of the numerical method. In this report, it is included with the analytic equations to emphasize that the weak form is not an approximation. It is, in fact, exact/analytic. The approximation occurs later when the test function is limited in the application of a discretization scheme. For each of the fundamental partial differential equations above, each equation is multiplied by a test function and integrated over the domain.

Consider a simple example:

$$L(u(x)) = 0, on\ 0 < x \leq 1, u(0) = 0$$

where $L(u(x)) = 0$ is the differential equation on the domain $0 < x \leq 1$, subject to the boundary condition at $x = 0$. The dependent variable is u.

The weak form of this same equation is $\int_0^1 w_0(x)\, L\big(u(x)\big)dx = 0, u(0) = 0$. The test function, $w_0(x)$, is a sufficiently smooth (defined later) function over the domain. The subscript indicates that it has a fixed value of 0 at $x = 0$. Otherwise, the test function is arbitrary over the domain, meaning that it can have any shape (subject to certain smoothness requirements). For this reason, a solution to the original differential equation is also a solution to the weak form. A simple mental experiment confirms that this is true. Imagine that the proposed $L\big(u(x)\big)$ is not zero over some small segment of the domain from 0 to 1. As stated earlier, the weak solution's test function can be any shape. If this test function chooses to have the shape of the proposed nonzero solution, then the integral will be nonzero in the weak form. Thus, any shape that is not a solution to the differential equation is also not a solution to the weak form.

One of the advantages of the weak form is that in addition to returning classical solutions, if they exist, it can also return weak solutions even when the classical solutions do not exist. This allows solutions to such problems as point/line/surface singular type sources, hydraulic jumps, material discontinuities and is helpful in imposing boundary conditions. This report will not detail all possible weak forms for the set of differential equations. Instead, the primary intent is to describe the underpinnings of the computational model for the interested user; therefore, the report includes detail of just the relevant finite element statements.

4 The Discrete Model (Approximation of the Mathematical Model)

To this point, the mathematical model is exact within the previously defined assumptions. This includes the weak form, which is exact and admits the classical solution if it exits, but it can also find weak solutions if the true solution is not smooth. In general, these equations that form the mathematical model cannot be solved directly. Instead, numerical methods must be found to get approximate solutions. The approach used for AdH-SW3 is from the overall form of discrete methods called *finite element*. Since AdH-SW3 (as with the overall AdH system) solves the conservative form of the equations, one can show that the method is also a finite volume method. Here, finite volume is meant to indicate that there is a discrete domain about which the computational flux balances the time rate of mass change in the domain. For AdH-SW3, this mass balance domain is the individual element. Of course, if the individual elements are locally conservative, then groups of elements would be as well. As such, the global domain is also conservative. The AdH-SW3 development remains obsessively committed to local, element-wise conservation. More background on this topic is found in Hughes et al. (2000) and Berger and Howington (2002). This topic will be discussed later in more detail.

The 3D elements used in AdH-SW3 are linear tetrahedral elements (with one exception, for pressure, described later). Any 2D surface domains are linear triangular elements. With linear elements, the description of the bed elevations and horizontal locations (independent variables) are spatially linear within an element, as are all the dependent variables. The shape functions that describe this linear variation through the element are the linear Lagrange functions, which is typical of finite element models. As such, the values in the interior of the element are determined by the values at the nodes, which are at the vertices of the element. For a tetrahedral element, there are four vertices, and so four nodes. Prior to this step, the equations were exact/analytic. To distinguish variables that are approximate within the discrete model, a subscript h is used. Analytic variables, such as those used in the previous equations, contain no such subscripts. Shape functions provide interpolation within the element. The shape functions have a value of 1 at one node and 0 at all the other nodes/vertices of that element. The shape function is zero for all other elements that do not contain that node. This results in linear interpolation

within each element. As an example, the discrete representation of concentration over an element is given as

$$c_h \equiv \sum_j \phi_j(x,y,z)c_j$$

The shape function is given as ϕ_j, and it is a function of the spatial dimensions. The nodal value of concentration at node j is given by c_j. The element velocity vector over an element is as follows:

$$v_h \equiv \begin{Bmatrix} v_{1_h} \\ v_{2_h} \\ v_{3_h} \end{Bmatrix} \equiv \begin{Bmatrix} \sum_j \phi_j(x,y,z)v_{1_j} \\ \sum_j \phi_j(x,y,z)v_{2_j} \\ \sum_j \phi_j(x,y,z)v_{3_j} \end{Bmatrix}$$

The average velocity over an element is designated as $\boldsymbol{V_h}$. Here, an upper-case letter indicates an element average value, and **bold face** indicates a vector or matrix.

This chapter will offer only a brief explanation of the equations and reasoning. The point of the chapter is to supply the equation sets, and the detailed explanation for each part is given in the following chapter "Discussion of the Discrete Model." The approach in AdH-SW3 is unique and requires an understanding of numerical processes. The discussions are very long. The reader needs to have the complete equation solution set in place before following discussions of individual parts. Therefore, the direct listing of the discrete approach is given in this chapter, and the detailed description and justification in the next.

4.1 Mesh movement

An extra complication in this numerical model compared to the stated analytic equations is that the mesh is allowed to move. This results in some modifications of these equations for the discrete model. The mesh moving changes the temporal derivative. Consider a general case involving an integral over a small domain, Ω_h. Here Ω_h and $\partial\Omega_h$ are the discrete 3D

domain and its 2D boundary, respectively. Assume that the velocity field is stagnant; the flow is zero everywhere. The movement of the mesh will impact the total derivative. As before, the discretized version of each of analytic domains is given a subscript h. The result is as follows:

$$\frac{\partial}{\partial t}\int_{\Omega_h} c_h d\Omega_h = \int_{\Omega_h} \frac{\partial c_h}{\partial t} d\Omega_h + \int_{\partial\Omega_h} c_h \boldsymbol{v_{m_h}} \cdot \boldsymbol{n} da_h$$

This is a statement of the Reynolds Transport Theorem. Here, $\boldsymbol{v}_{m_h}$, is the mesh velocity vector. $\partial\Omega_h$ represents the exterior surface of Ω_h, and $\boldsymbol{n}$ is the outward normal vector for this surface. The term $\frac{\partial c_h}{\partial t}$ is how the concentration changes in time at a fixed location. The total derivative on the left-hand side is how the concentration changes with the moving mesh. The flux terms are on the far-right side of the equation. This equation then is presenting that the concentration rate of change for a volume moving with the mesh is given by the rate of change of the concentration at a fixed location plus the flux at the boundary. Now, including a nonzero flow field and rearranging and using the definition of divergence results in the following form of the equation for a moving mesh used in this report.

$$\int_{\Omega_h} \frac{\partial c_h}{\partial t} d\Omega_h = \frac{\partial}{\partial t}\int_{\Omega_h} c_h d\Omega_h - \int_{\Omega_h} \boldsymbol{\nabla} \cdot \left(\boldsymbol{v_{m_h}} - \boldsymbol{v_h}\right)(c_h) d\Omega_h \tag{11}$$

4.2 Time advancement

Nodal values for the variables are associated with a particular time level or time-step. AdH-SW3 is implicit, meaning that all the solution variables are computed simultaneously at the new time level. Since newest time level, $n+1$, is common, variables at this time level will not have a superscript for the remainder of this document. For example, c_h^{n-1}, c_h^n, and c_h indicate time levels $n-1, n,$ and $n+1$, respectively.

The temporal derivative uses a standard finite difference approximation technique. It is defined by a weighting of a first- and second-order backward difference. This method is used for all the temporal derivatives of the dependent variables and the mesh speed. This is identical to the method used for AdH-SW2. As an example, the variable η, which typically stands for water surface elevation, is here a stand-in for any variable.

$$\frac{\partial \eta}{\partial t}_i \approx \alpha_t \frac{\left(3\eta_i - 4\eta_i^n + \eta_i^{n-1}\right)}{\left(2\Delta t^{n+1}\right)} + (1-\alpha_t)\frac{\left(\eta_i - \eta_i^n\right)}{\Delta t^{n+1}} \tag{12}$$

The subscript i indicates a particular node. The superscript indicates at which time level. ADH is an implicit model, so time level $n+1$ is the newest time level. It is also the time level at which all the spatial calculations are made. As a reminder, the variable η_i is at time level $n+1$. Time levels n and $n-1$ are at one time level earlier and two time levels earlier, respectively. All the variable values are known at the earlier two time levels. Only the variable values at time level $n+1$ are not known. The term Δt^{n+1} indicates the length of the time-step from time level n to time level $n+1$. α_t is a weight that the model user chooses. It must be $0 \leq \alpha_t \leq 1$. The higher the value of α_t, within the set bounds, the higher-order the temporal derivative.

While this method is consistent with AdH-SW2, it does not consider that the time-step size may not be constant. A modeler will take smaller time-steps when the activity is more rapid. Something to consider is to modify the AdH approach to account for this change in time-step size.

$$\frac{\partial \eta}{\partial t_i} \approx \alpha_t \frac{\Delta t^n\left(2\Delta t^{n+1}+\Delta t^n\right)\eta_i-\left(\Delta t^{n+1}+\Delta t^n\right)^2\eta_i^n+\left(\Delta t^{n+1}\right)^2\eta_i^{n-1}}{\left(\Delta t^{n+1}\left(\Delta t^{n+1}+\Delta t^n\right)^2-\left(\Delta t^{n+1}+\Delta t^n\right)\left(\Delta t^{n+1}\right)^2\right)} + (1-\alpha_t)\frac{\left(\eta_i-\eta_i^n\right)}{\Delta t^{n+1}} \quad (13)$$

Here, Δt^n is the time difference between time levels n and $n-1$. This formulation is a result of a straight-forward Taylor Series expansion with a variable time-step.

4.3 Computational sequence

Before describing the computational sequence in some detail, there are basics concepts that were lodestar guidance in the model development approach are listed. These can be summarized as *consistency*. This consistency is from several different *directions*.

The important principles that are followed in the development of AdH-SW3 are the following:

1. Consistency in terms of the test functions is applied for the entire equation set to the maximum degree possible. This leads to a model that is higher-order convergent and less subject to obscure behavior problems.
2. AdH-SW3 is designed to be locally mass conservative on each element.
3. The calculation of the depth-integrated continuity step and the next step to calculate the vertical velocity is composed of the same

individual finite element equations. The first is the sum of the second. This makes AdH-SW3 remain conservative.

4. AdH-SW3 is consistent with the stabilization equations for two dimensions as well. The 3D model stabilizes the 2D oscillatory modes like AdH-SW2 and stabilizes any additional 3D oscillatory modes.
5. Finally, AdH is an implicit model. This means that the new time-step is found using spatial terms that involve this same new time-step data. The other method of time advancement is termed *explicit*. In that case, the new time-step data at a location are calculated using all older data from space. The implicit method has no theoretical time-step limitation for stability. The explicit approach is limited by the time it takes a wave or particle to traverse an element. For the type of problems AdH-SW3 is targeting, this is far too restrictive. The implicit approach is much more efficient.

The basic steps in the calculation of the hydrodynamics and constituent transport proceeds in a particular staged manner. The various steps are consistent from one step to the next. The first two steps involve purely hydrodynamics, and the third step would be any constituent transport calculations.

4.3.1 Step 1. Horizontal momentum and depth-integrated continuity equations

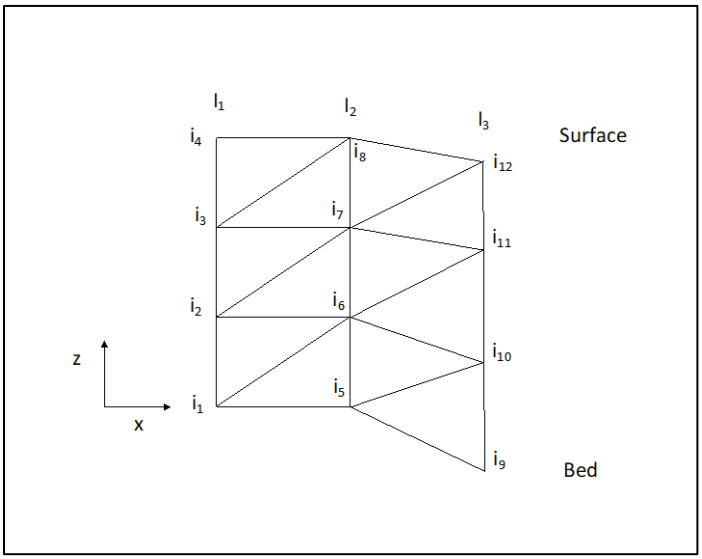
The depth-integrated mass conservation (continuity) equation is solved as a single step along with the horizontal momentum equations. This allows the calculation of the water surface elevation and the two horizontal components of velocity. The depth-integrated continuity finite element equations are now described.

$$\sum_{i \in I} \left(W_i + \sum_e PWW_i + \sum_e PWM_{x_i} + \sum_e PWM_{y_i} \right) = 0 \qquad (14)$$

The naming convention used here uses "W" to indicate the continuity equation. "M_x" and "M_y" indicate the x- and y-momentum equations, respectively. The "P" means that this is a Petrov-Galerkin perturbation term. The order of the terms in Equation (14) with a prefix "P" is "P" to indicate the perturbation term, followed by the equation that is being solved, and finally the equation that is contributing to that equation. As an example, the term "PWM_{x_i}" is the perturbation part of the Petrov-Galerkin terms that contribute to the continuity equation. The x-momentum

equations are making that contribution. Here, i is the index of a particular distributed (over depth) test function, and I is the index of a particular column of test functions. An example is given in Figure (6).

Figure 6. A two-dimensional mesh in the x-z plane.



The vertices are nodes, and each triangle constitutes an element. The nodes at the surface have both an individual test function and can be regarded as being the sum of the test functions of the nodes directly below them. This would make them a combined test function. These surface nodes are labeled with a capital letter "I," representing the depth-integrated column.

I_2 is a particular column number in this figure, and i_5, i_6, i_7, and i_8 make up the individual nodes (and test function indices) that comprise that column. The depth integration is a sum of the individual vertically distributed test functions. The individual continuity equations (one for each test function) are the numerical representation of Equation (1) with boundary conditions in Equations (7) and (8). Previous models have solved the depth-integrated continuity equation and the vertical velocity calculations by simply applying numerical methods to the differential equations for each independently. This approach creates problems. The method will not preserve mass locally. Instead, what must be done is to be sure to use precisely the same numerical representation for the vertical velocity calculation (using continuity) and the depth-integrated continuity. AdH-SW3 sums the individual continuity equations over the vertical to get the depth-integrated continuity equation. This results in the calculation of the

water surface location. In the next step, the exact same individual continuity equations are used again to calculate the vertical velocity. The domain and boundary designations are shown in Figure (5). For each node in the column, Equation (15) shows the contributions from a moving water surface, a moving bed, fluxes at specified pressure or flux boundaries, fluxes at the water surface (+ and -) and fluxes at the bed (+ and -). For the interior nodes, the surface and bottom integrals will be zero.

$$
\begin{aligned}
W_i \equiv & -\int_{\Omega_h} \boldsymbol{\nabla}\phi_i \cdot \boldsymbol{v}_h d\Omega_h \\
& + \int_{\partial\Omega_{\eta_h}} \phi_i \frac{\partial \eta_h}{\partial t} da_{z_h} - \int_{\partial\Omega_{b_h}} \phi_i \frac{\partial b_h}{\partial t} da_{z_h} \\
& + \int_{\partial\Omega_{p_h}} \phi_i \boldsymbol{n}_h \cdot \boldsymbol{v}_h da_h - \int_{\partial\Omega_{f_h}} \phi_i \mathcal{F}\, da_h \\
& - \int_{\partial\Omega_{\eta_h}} \phi_i S_{\eta_+} da_{z_h} + \int_{\partial\Omega_{\eta_h}} \phi_i S_{\eta_-} da_{z_h} \\
& - \int_{\partial\Omega_{b_h}} \phi_i S_{b_+} da_{z_h} + \int_{\partial\Omega_{b_h}} \phi_i S_{b_-} da_{z_h}
\end{aligned}
\tag{15}
$$

The boundary designated, $\partial\Omega_{p_h}$, is a boundary where pressure is specified, which, for the continuity equation, means that mass is allowed to flow out unimpeded. This is what that natural boundary conditions specifies. The boundary segment designated, $\partial\Omega_{f_h}$, is a flux inflow boundary condition. The flow per unit area is $\mathcal{F}$. The boundary segments designated $\partial\Omega_{\eta_h}$ and $\partial\Omega_{b_h}$ are the free-surface and the bed, respectively. S_{η_+} and S_{η_-} are non-negative values indicating volume flux per unit area at the free-surface, where the subscript "+" means a source and the subscript "-" means a sink. For example, these might be rain and evaporation, respectively. Similarly, S_{b_+} and S_{b_-} are the volume flux per unit area through the bed. Typically, this is exchange with the groundwater. The term a_{z_h} is the projected area normal to the positive vertical z-axis.

$$
\begin{aligned}
PWW_i \equiv & \int_{\Omega_e} \frac{\alpha l_w}{\left(V_{1_h}^2(\eta_h) + V_{2_h}^2(\eta_h) + gH_h\right)^{1/2}} \Big(V_{1_h}(\eta_h) \frac{\partial \phi_i}{\partial x} \\
& + V_{2_h}(\eta_h) \frac{\partial \phi_i}{\partial y} \Big) \{ (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h)\, d\Omega_e \} - \int_{\Omega_e} \alpha_z l_z \frac{\partial \phi_i}{\partial z} \boldsymbol{\nabla} \cdot \boldsymbol{v}_h \, d\Omega_e
\end{aligned}
\tag{16}
$$

$$PWM_{x_i} \equiv \frac{\alpha l_w}{\left(V_{1_h}^2(\eta_h) + V_{2_h}^2(\eta_h) + gH_h\right)^{1/2}} \frac{\partial \phi_i}{\partial x} \left(\frac{\partial}{\partial t} \int_{\Omega_e} v_{1_h} \, d\Omega_e + \int_{\Omega_e} \left(\boldsymbol{v}_{r_h} \cdot \boldsymbol{\nabla} v_{1_h} - v_{2_h} \dot{\omega} + \frac{1}{\rho_0} \frac{\partial P_h}{\partial x} - \frac{1}{\rho_0} \boldsymbol{\nabla} \cdot \boldsymbol{\tau}_{x_h} \right) d\Omega_e \right) \tag{17}$$

$$PWM_{y_i} \equiv \frac{\alpha l_w}{\left(V_{1_h}^2(\eta_h) + V_{2_h}^2(\eta_h) + gH_h\right)^{1/2}} \frac{\partial \phi_i}{\partial y} \left(\frac{\partial}{\partial t} \int_{\Omega_e} v_{2_h} \, d\Omega_e + \int_{\Omega_e} \left(\boldsymbol{v}_{r_h} \cdot \boldsymbol{\nabla} v_{2_h} + v_{1_h} \dot{\omega} + \frac{1}{\rho_0} \frac{\partial P_h}{\partial y} - \frac{1}{\rho_0} \boldsymbol{\nabla} \cdot \boldsymbol{\tau}_{y_h} \right) d\Omega_e \right) \tag{18}$$

The momentum equations are not depth integrated over the column in any fashion. These yield the horizontal velocity components at all nodal locations. l_w is the square root of the horizontally projected area of the element.

The finite element approximation for the prior equations is now given.

The x-direction finite element equation follows.

$$M_{x_i} + \sum_e PM_x W_i + \sum_e PM_x M_{x_i} = 0 \tag{19}$$

$$\begin{aligned} M_{x_i} \equiv \frac{\partial}{\partial t} \int_{\Omega_h} \phi_i v_{1_h} d\Omega - \int_{\Omega_h} \left(\boldsymbol{\nabla} \phi_i \cdot \left(\boldsymbol{v}_{\boldsymbol{r_h}} v_{1_h} \right) + \phi_i v_{2_h} \dot{\omega} \right) d\Omega_h \\ + \int_{\Omega_h} \frac{\partial \phi_i}{\partial x} \frac{P_h}{\rho_0} d\Omega + \int_{\Omega_h} \boldsymbol{\nabla} \phi_i \cdot \boldsymbol{\tau}_{\boldsymbol{x_h}} d\Omega \\ + \int_{\partial \Omega_{p_h}} \left(\boldsymbol{n} \cdot \boldsymbol{v}_{\boldsymbol{r_h}} v_1 + n_x \frac{\hat{P}}{\rho_0} \right) da_h \\ + \int_{\partial \Omega_{w_h}} \left(-\phi_i \hat{\boldsymbol{\tau}}_{\boldsymbol{x}} \cdot \boldsymbol{n} + n_x \frac{P_h}{\rho_0} \right) da_h \end{aligned} \tag{20}$$

$$PM_x W_i \equiv \frac{\alpha l_m g H_h}{\left(\boldsymbol{V}_{\boldsymbol{r_h}} \cdot \boldsymbol{V}_{\boldsymbol{r_h}} + g H_h\right)^{\frac{1}{2}}} \int_{\Omega_e} \frac{\partial \phi_i}{\partial x} \boldsymbol{\nabla} \cdot \boldsymbol{v_h} d\Omega_e \tag{21}$$

$$\begin{aligned} PM_x M_{x_i} \equiv & \frac{\alpha l_m}{\left(\boldsymbol{V}_{\boldsymbol{r_h}} \cdot \boldsymbol{V}_{\boldsymbol{r_h}} + g H_h\right)^{\frac{1}{2}}} \left(\boldsymbol{V}_{\boldsymbol{r_h}} \cdot \boldsymbol{\nabla} \phi_i\right) \left(\frac{\partial}{\partial t} \int_{\Omega_e} v_{1_h} \, d\Omega_e \right. \\ & \left. + \int_{\Omega_e} \left(\boldsymbol{v}_r \cdot \boldsymbol{\nabla} v_{1_h} - v_2 \dot{\omega} + \frac{1}{\rho_0} \frac{\partial P_h}{\partial x} - \frac{1}{\rho_0} \boldsymbol{\nabla} \cdot \boldsymbol{\tau}_{\boldsymbol{x_h}} \right) d\Omega_e \right) \end{aligned} \tag{22}$$

The y-direction momentum equation is next.

$$M_{y_i} + \sum_e PM_y W_i + \sum_e PM_y M_{y_i} = 0 \tag{23}$$

$$\begin{aligned} M_{y_i} \equiv & \frac{\partial}{\partial t} \int_{\Omega_h} \phi_i v_{2_h} d\Omega - \int_{\Omega_h} \left(\boldsymbol{\nabla} \phi_i \cdot \left(\boldsymbol{v}_{\boldsymbol{r_h}} v_{2_h} \right) - \phi_i v_{1_h} \dot{\omega} \right) d\Omega \\ & + \int_{\Omega_h} \frac{\partial \phi_i}{\partial y} \frac{P_h}{\rho_0} d\Omega + \int_{\Omega_h} \boldsymbol{\nabla} \phi_i \cdot \boldsymbol{\tau}_{\boldsymbol{y_h}} d\Omega \\ & + \int_{\partial \Omega_{p_h}} \left(\boldsymbol{n} \cdot \boldsymbol{v}_{\boldsymbol{r_h}} v_{2_h} + n_y \frac{\hat{P}}{\rho_0} \right) da \\ & + \int_{\partial \Omega_{w_h}} \left(-\phi_i \hat{\boldsymbol{\tau}}_{\boldsymbol{y}} \cdot \boldsymbol{n} + n_y \frac{P_h}{\rho_0} \right) da \end{aligned} \tag{24}$$

$$PM_y W_i \equiv \frac{\alpha l_m g H_h}{\left(\boldsymbol{V}_{\boldsymbol{r_h}} \cdot \boldsymbol{V}_{\boldsymbol{r_h}} + g H_h\right)^{\frac{1}{2}}} \int_{\Omega_e} \frac{\partial \phi_i}{\partial y} \boldsymbol{\nabla} \cdot \boldsymbol{v_h} d\Omega_e \tag{25}$$

$$\begin{aligned} PM_y M_{y_i} \equiv & \frac{\alpha l_m}{\left(\boldsymbol{V}_{\boldsymbol{r_h}} \cdot \boldsymbol{V}_{\boldsymbol{r_h}} + g H_h\right)^{\frac{1}{2}}} \left(\boldsymbol{V}_{\boldsymbol{r_h}} \cdot \boldsymbol{\nabla} \phi_i\right) \left(\frac{\partial}{\partial t} \int_{\Omega_e} v_{2_h} \, d\Omega_e \right. \\ & \left. + \int_{\Omega_e} \left(\boldsymbol{v}_{r_h} \cdot \boldsymbol{\nabla} v_{2_h} + v_{1_h} \dot{\omega} + \frac{1}{\rho_0} \frac{\partial P_h}{\partial y} - \frac{1}{\rho_0} \boldsymbol{\nabla} \cdot \boldsymbol{\tau}_{y_h} \right) d\Omega_e \right) \end{aligned} \tag{26}$$

where

$$l_m \equiv Max\left((5\Omega_e)^{\frac{1}{3}}, \frac{\boldsymbol{V}_{r_h} \cdot \boldsymbol{V}_{r_h}}{\sum_j^4 \left|\boldsymbol{V}_{r_h} \cdot \boldsymbol{\nabla}\phi_j\right|}\right)$$

In all the preceding equations, $\boldsymbol{v_h} \equiv \begin{Bmatrix} v_{1_h}^{n+1} \\ v_{2_h}^{n+1} \\ v_{3_h}^{n} \end{Bmatrix}$. This is subtle. However, what it is saying is that the vertical velocity, $v_{3_h}^n$, is the previous time-step value since it is not being calculated and updated. The other two velocity components (horizontal components) are being calculated and updated. Therefore, the horizontal components are at the new time level. This is true in all the velocity vectors ($\boldsymbol{v}_{r_h}$, $\boldsymbol{V_h}$, and $\boldsymbol{V}_{r_h}$). Remember that capital letters, such as "$\boldsymbol{V}$", indicate element average quantities. In the case of the vectors that are relative velocity, the grid speed is recalculated each iteration and so is the new grid speed. The pressure is also updated when the mesh moves and is also the newest value. The AdH solver uses a Newton method to calculate the Jacobian of the residual equations. The pressure and grid speed are included in this Jacobian via numerical differentiation. Although the pressure and its derivative with respect to water surface are calculated in a separate step rather than the residual equation direct differentiation.

4.3.2 Step 2: Vertical velocity calculation

In a hydrostatic code the vertical velocity does not depend on vertical inertia. In fact, vertical inertia is assumed to be negligible. This does not mean that vertical velocity is zero. It means that there is no physical bound on vertical acceleration. A fluid can accelerate as quickly as is dictated by mass conservation.

The distributed conservation of mass equation then is

$$W_{d_i} + \sum_e PWW_i + \sum_e PWM_{x_i} + \sum_e PWM_{y_i} = 0 \qquad (27)$$

where

$$W_{d_i} \equiv -\int_{\Omega_h} \boldsymbol{\nabla}\phi_i \cdot \boldsymbol{v}_h d\Omega + \int_{\partial\Omega_{\eta_h}} \phi_i \frac{\partial \eta_h}{\partial t} da_z - \int_{\partial\Omega_{\eta_h}} \phi_i S_{\eta_+} da_z + \int_{\partial\Omega_{\eta_h}} \phi_i S_{\eta_-} da_{z_h} \tag{28}$$

which is the same as the depth-integrated conservation of mass contribution as W_i, except all bed boundary conditions are removed. The three perturbation stabilization terms PWW_i, PWM_{x_i}, and PWM_{y_i} are defined previously.

Within the PWW_i (see Equation (16)) there is the term

$$-\sum_e \int_{\Omega_e} \alpha_z l_z \frac{\partial \phi_i}{\partial z} \boldsymbol{\nabla} \cdot \boldsymbol{v}_h \, d\Omega_e$$

Its purpose is to provide a direction to the calculations so that only information from above a node is used in the vertical velocity calculation and not from below. The continuity equation (Equation (1)) is a first-order differential equation. This permits only a single boundary condition. Because AdH-SW3 is solving for a free-surface as well, the equations permit a second boundary condition to get the depth-integrated equations. After the first step of calculations, the horizontal velocity at all nodes and the location of the free surface are known. At this second step here, AdH-SW3 is allowed only a single boundary condition. This second step, in which the vertical velocity is calculated, must be identical to the equations of continuity in the first step. At this point it would be possible to simply start at either the bed or the free surface with the appropriate kinematic boundary condition and sum individual continuity equations to get the vertical velocity at all the nodes in a column. By being perfectly identical to the segments of the depth-integrated continuity equation the boundary condition at the opposite boundary is automatically satisfied. If one chose to start at the bed with the bed kinematic boundary condition, the sum would result in a velocity field that satisfied the free-surface boundary condition as well. This additional term, shown above, forces a backward difference statement in the vertical that does not use nodal values below the node "i". In an unstructured tetrahedral mesh, it is difficult to simply start at the surface and make row by row calculations down the column to get the individual vertical velocities. Instead, all the vertical velocities are solved from a single linear algebra solution. This additional term keeps the

lower (in the column) nodes from entering the calculation. This term is of course also in the depth-integrated conservation of mass statement. There, these terms go away as one sums over depth.

However, since there is no bed boundary condition, flow can pass through the bed if the method is not fully consistent with Step 1. To that end, there are procedures that are followed to make sure that Step 1 and Step 2 are consistent. The main issue is with the Petrov Galerkin perturbation test function terms, PWM_{x_i} and the M_{y_i} . These terms contain v_{3_h} terms that must use the old time-step values. This is consistent with Step 1 since it is updating the nodal horizontal velocities but not the nodal vertical velocities.

As a final comment on the vertical velocity calculation, the choice of using either the surface or bottom kinematic boundary condition is arbitrary. To try the bottom boundary instead the modeler would need to reverse the sign on the terms above. The author chose to use the surface kinematic boundary condition since the water surface is relatively smooth compared to the bed. The thought was that perhaps precision in the calculation of the surface boundary condition would be better. The other reason is that when sedimentation calculations are being made, the bed will move. The sedimentation and bed movement are done in later steps. It seems simpler to just deal with water surface calculations, making consistency easier.

The sidewall boundary conditions are now addressed. In the AdH-SW2 module, the flow is enforced along the sidewall by use of natural boundary conditions. The integration by parts that takes place for the flux terms results in boundary terms that are the flux of water, momentum, and constituents through the sidewall boundary. In two-dimensions this is quite simple. By assuming that these terms are zero, no flux of any of these entities is allowed through the boundary. In terms of the water and momentum, the water surface and therefore the pressure are calculated in a manner that does not allow a flux through the boundary. Any transported constituent has a zero flux through the boundary by setting the constituent flux to zero as well. This method is not only simple; it is robust. For the AdH-SW3 module, this approach is not enough. The transported constituent flux and water mass flux can be set to zero successfully. The problem comes about in assuring no momentum flux through the boundary. The hydrostatic pressure is a single degree of freedom per nodal column. This does not allow enough flexibility to enforce a no-momentum flux condition over all the nodes/test functions of

that column. Therefore, AdH-SW3 enforces the depth-averaged water volume and momentum along the boundary through the natural boundary statement. However, the no-momentum penetration condition for the individual test functions over a column is handled by forcing the flow to be parallel to the sidewall.

In a manner like that used in the hydrodynamic model RMA10 (King 1982), a single parallel flow momentum equation on the sidewall is created using the components of the cartesian discrete momentum equations. The tangent to the sidewall will be denoted $\boldsymbol{J_i} = \{J_{x_i}, J_{y_i}\}$. The tangent direction is derived as the weighted average of the element faces that contain node i. The new sidewall momentum equation is then formed as the linear combination of the horizontal plane momentum equations.

$$J_{x_i} Eq(19) + J_{y_i} Eq(23) = 0 \tag{29}$$

The terms $Eq(19)$ and $Eq(23)$ stand for Equation (19) and Equation (23). This method functions reasonably well. Where there are sharp breaks in the sidewall, the results become erratic, however.

The way sidewalls are handled is an area that deserves further investigation. Perhaps constraints on the individual momentum equations that allow no momentum pass through the boundary could be invoked. These constraints would likely end up mimicking wall horizontal pressure gradients that enforce this condition.

4.3.3 Step 3: Constituent transport calculation

The transport for an individual constituent is given by the following discrete equation:

$$\mathbb{C}_i + \sum_e P\mathbb{C}_i + \sum_e C_h PWW_i + \sum_e C_h PWM_{x_i} + \sum_e C_h PWM_{y_i} = 0 \tag{30}$$

Here

$$\begin{aligned}\mathbb{C}_i \equiv \frac{\partial}{\partial t}\int_\Omega \phi_i\, c_h d\Omega - \int_\Omega \boldsymbol{\nabla}\phi_i \cdot \left(\boldsymbol{v}_{\boldsymbol{r_h}} c_h\right) d\Omega \\ + \int_\Omega \boldsymbol{\nabla}\phi_i \cdot (\boldsymbol{D}\boldsymbol{\nabla} c_h) d\Omega - \int_\Omega \phi_i S_c d\Omega \\ + \int_{\partial\Omega_p} \phi_i \boldsymbol{n_h} \cdot \left(\boldsymbol{v}_{\boldsymbol{r_h}} c_h\right) da - \int_{\partial\Omega_f} \phi_i \mathcal{F}\hat{c}_f da \\ - \int_{\partial\Omega_\eta} \phi_i S_{\eta_+}\hat{c}_\eta da_z - \int_{\partial\Omega_b} \phi_i S_{b_+}\hat{c}_b da_z\end{aligned} \tag{31}$$

$$P\mathbb{C}_i \equiv \alpha_c\left(\boldsymbol{V}_{\boldsymbol{r_h}} \cdot \boldsymbol{\nabla}\phi_i\right)\int_{\Omega_e}\left\{\frac{\partial c_h}{\partial t} + \boldsymbol{v}_{\boldsymbol{r_h}} \cdot \boldsymbol{\nabla} c_h - \boldsymbol{\nabla}\cdot(\boldsymbol{D}\boldsymbol{\nabla} c_h) - S_c\right\} d\Omega \tag{32}$$

where

$$\alpha_c = \frac{1}{2}\xi l_c \frac{1}{\left\|\boldsymbol{V}_{r_h}\right\|} \tag{33}$$

$$\xi = \begin{cases}\frac{1}{3}\alpha_p, \alpha_p \le 3 \\ 1, \alpha > 3\end{cases} \tag{34}$$

$$\alpha_p \equiv \frac{1}{2}\frac{\left\|\boldsymbol{V}_{\boldsymbol{r_h}}\right\| l_c}{D_{iso}} \tag{35}$$

α then is a Peclet number, D_{iso} is an estimate of an isotropic diffusion value that comes from the defined diffusion tensor, and l_c is an estimate of the 3D element length in the direction of flow.

$$l_c = 2\left(\sum_{i=1}^{npe}\left|\frac{\boldsymbol{V}_{\boldsymbol{r_h}}}{\left\|\boldsymbol{V}_{\boldsymbol{r_h}}\right\|} \cdot \boldsymbol{\nabla}\phi_i\right|\right)^{-1} \tag{36}$$

Here, *npe* is the number of nodes per element. In the case of a tetrahedral element, this is four nodes. This streamline upwind Petrov-Galerkin (SUPG) approach is a modification of the form of Tezduyar and Park(1986).

The terms $\sum_e C_h PWW_i + \sum_e C_h PWM_{x_i} + \sum_e C_h PWM_{y_i}$ are there to provide consistency with the water conservation discrete equation. Basically, the condition of constant concentration with no source/sinks should be a solution in the discrete equations. However, it is found that while this is true in some average sense, at the smallest spatial intervals there will be oscillations. These come about if the transport conservation of mass equation is not truly consistent with the water conservation of mass equation. These terms provide this consistency. A more detailed account is given in the "Discussion" chapter. One thing to consider is that these additional terms will make the transport equation nonlinear, although this will not be strongly nonlinear.

4.4 Solution using Newton approach

Within each step above, the discrete equations are solved. This section discusses this solution approach. Steps 1 and 2 are nonlinear. The unknowns not only depend on each other, they are formed by products of themselves and each other or raised to some power. The result is that the algebraic equations are nonlinear. Unlike linear algebraic equations, these equations need to be solved in an iterative fashion. AdH uses a Newton method for this solution. The residual of one of these nonlinear algebraic equations is the result of substituting estimates of the unknowns and calculating the result. If this residual is zero, then these estimates satisfy the equation. If not, the calculations use a Newton-type technique to drive the residuals to zero. In a Newton method the way the residual is driven to zero is by calculating how the residual changes with each unknown variable. This can be done analytically or by numerical differentiation. The challenge with finding these derivatives analytically is daunting. There are libraries that feed the residual, and the equations themselves are complicated. Examples of libraries are the friction library or a turbulence library. These are calculated separately in another affiliated computer code. In this case, the parent code would not be able to calculate the impact on the residuals of changes in velocity or water depth in an analytic fashion. Instead, AdH uses numerical differentiation to calculate the derivatives that describe how the residuals change due to each unknown variable. This approach automatically includes the libraries and any complicated interactions between unknown variables.

As with any differentiation technique, the perturbations used to make these calculations must be sufficiently small so that the derivative is accurate. AdH uses perturbations that are scaled by the size of single precision zero

on the computer. This assures that the results (which are double precision) will not contain higher-order terms. In the case of the transport equations (Step 3), the discrete equations are typically linear. However, AdH continues to use the Newton method and direct differentiation. The transport equations should converge in a single iteration since the equation set is linear. Using the size of the single precision size perturbation in AdH, if the transport does not converge in a single iteration, this means that there is a computer code mistake and should be corrected.

5 Discussion of the Discrete Model

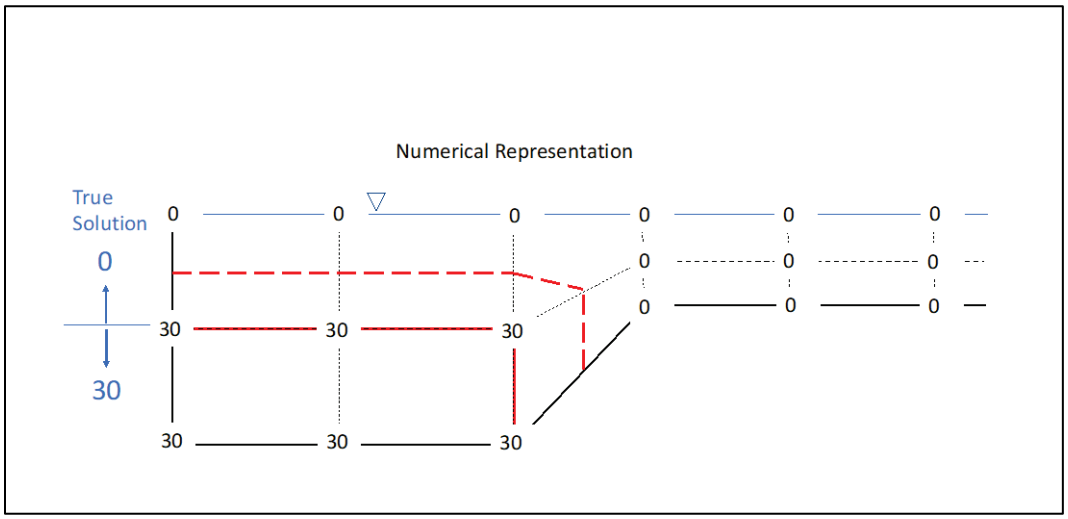
This section first explains in some detail the basic principles that are used in AdH-SW3. These principles apply for all the calculations (steps). The second part of the discussion section explains each of the individual steps of the calculation in order. The discussion, where possible, will explain these elements in a less mathematical way. The approach is intended to make AdH-SW3 understandable to a wider audience.

5.1 Mesh issues

An important issue to address is how to accurately model stratification in a channel with steep side slopes. In an estuary with a navigation channel, the channel will likely have steep side slopes. These are typically on the order of 20% slope. This is much steeper than natural estuarine terrain. There are issues associated with the mild slope assumption inherent in shallow water models. However, a larger concern is maintaining stratification in the channel. One of the first cases (perhaps the first) in which this problem was discovered was in the modeling of Chesapeake Bay by B. H. Johnson at the US Army Corps of Engineers, Waterways Experiment Station, Hydraulics Laboratory, approximately 1988 (Johnson et al. 1991). The model that was being implemented had a bed-following mesh that is called σ-plane transformation. This terminology is a result of a transformation of the physical vertical space to a computational space that falls between a σ value of zero at the bed to a value of 1 at the surface. The result is that any stratification that develops in the navigation channel is quickly dispersed from the channel by a combination of artificial pressure gradients and artificial diffusion. This led to the reprogramming of the model code so that the mesh could be laid out in horizontal layers. This type of mesh is called a z-plane mesh. This approach allows horizontal stratification to persist.

The problem with the pressure is apparent from Figure 7.

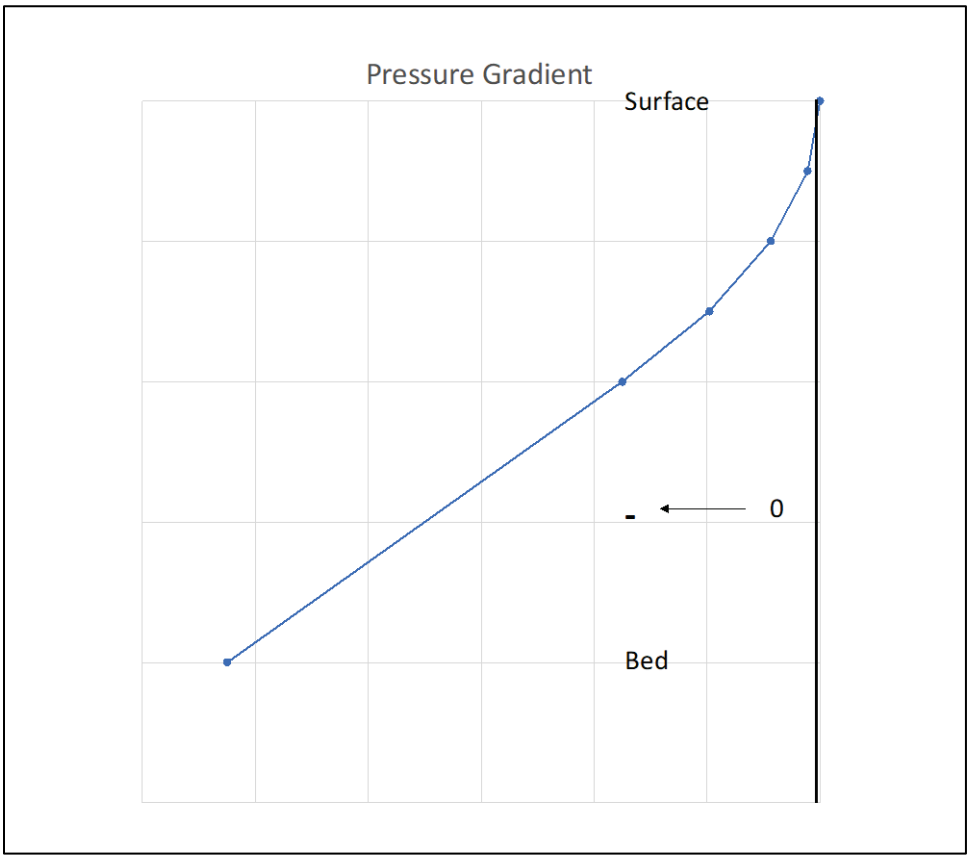
Figure 7. Cross section of a channel with a steep side slope.



A two-level stratification is being approximated. On the left in blue is the true solution. The channel is represented by a mesh that follows the bed (σ mesh). An element is shown as a rectangle. The values of salinity at each vertex (node) are assigned based upon the true solution. This results in nodes below a particular level have a salinity of 30 ppt, and above that level a salinity of 0 ppt. The red solid line shows a contour of the 30 ppt isohaline. The dashed red line shows the 15 ppt isohaline.

This mesh in Figure 7 is attempting to represent a stable stratification. This is salinity in two layers. The lower layer is 30 ppt and the upper layer is 0 ppt. The elements (rectangles) are interpolating concentration linearly. The mesh then results in the two contours of salinity (isohalines) shown in the figure. The true solution is two horizontal layers. This is a true solution of the 3D shallow water equations. The resulting flow field should be zero everywhere. Unfortunately, this mesh cannot produce that result. It will generate currents and tend to move salinity out of the channel. This is apparent by considering the two isohaline contours. The true solution has only horizontal isohalines. This numerical result has segments of the isohalines that are not horizontal. There are horizontal gradients in salinity (that should not exist), and this means there are horizontal pressure gradients as well. Figure 8 shows an example that is like what a vertical line halfway up the side slope would show.

Figure 8. The pressure gradient over depth.

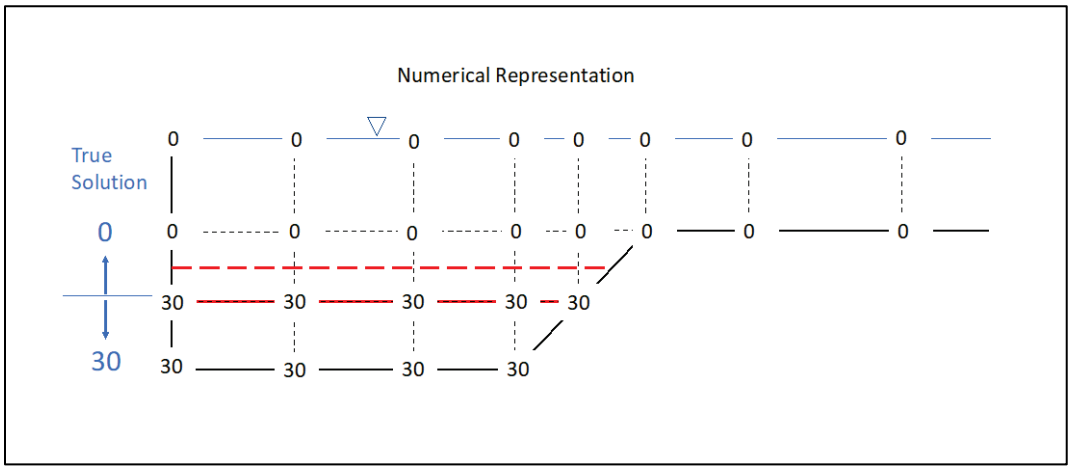


This is the pressure from just the increased density due to salinity. The gradient over depth at this location is always negative. The gradient is larger in magnitude over depth.

The figure shows that the gradient is negative over all depth. Since the water surface is flat, then the pressure gradient is due to the salinity gradient. The gradient is larger in magnitude with depth. The result is that there will be a current generated to the right near the bed. This will lead to a current to the left near the surface. The result is a tendency to move salt water from the channel.

If a mesh is generated that is nearly horizontal (like a *z*-plane mesh), the result would look something like Figure 9.

Figure 9. A mesh laid out in horizontal layers.



The solid red line represents the 30 ppt isohaline. The dashed red line represents the 15 ppt isohaline.

The mesh shown in Figure 9 does reasonably represent a two-layer stratified channel. The isohalines are horizontal. There is no horizontal pressure gradient. This configuration will not allow saline water to escape the channel for artificial reasons.

The computational approach that was used for that Chesapeake Bay model project was a Boundary-Fitted Coordinate approach (e.g., Thompson 1982). As the name implies, this approach defines a new coordinate system amenable to the shape of the computational domain, allowing the computational mesh to *fit* the boundaries. This does require that the differential equations be transformed into this new coordinate system. For example, a z-plane coordinate system has horizontal layers, and the mesh necessarily does too. In AdH-SW3, no such transformation is used. The actual physical space is used with a Cartesian coordinate system. This is advantageous in at least two regards. First, the differential equations are less complex. Second, there is no inaccuracy introduced by any severe transformations at things like bathymetric breaks in slope. The mesh is independent of the coordinate system. The knowledgeable model user is then able to create a mesh with horizontal layers in regions (navigation channels) with stratification and steep side slopes but have more bed-following mesh in other flatter regions.

A transformation is valuable in that the mesh movement with the free surface is directly included in the differential equations. For an implicit

method, this transformation would mean that the mesh speed is found simultaneously with the calculation of the free-surface. In AdH-SW3, this is still accomplished in an implicit framework through the computational techniques employed. AdH uses Newton's method to address the nonlinearity of the partial differential equations. The Jacobian is found by numerical differentiation. The free-surface movement is fed into the mesh movement and is calculated during the numerical differentiation step. Therefore, the method is fully implicit in the first step of the AdH-SW3 calculation.

5.2 Conservation

The design and implementation of AdH-SW3 goes to great lengths to assure that each step is *locally conservative*. Local conservation in this context means that the computational fluxes around an element accurately balance the *mass* change within the element.

AdH-SW3 can be considered a Finite Element Method or a Finite Volume Method. (For a definition of the finite volume method, see Hirsch [1988]). The finite volume method has fluxes defined over the edges of a cell/element that balance sources/sinks and temporal changes within the element. This section will illustrate that since AdH-SW3 is a conservative form representation of the differential equations, it is locally conservative over its most basic spatial form, an element. This also means that AdH-SW3 can be thought of as a Finite Volume Method as well as a Finite Element Method.

In a traditional Finite Volume Method, the edge fluxes are defined to be unique. The discrete representation of the conservation differential equation is then solved to satisfy conservation over an element or cell.

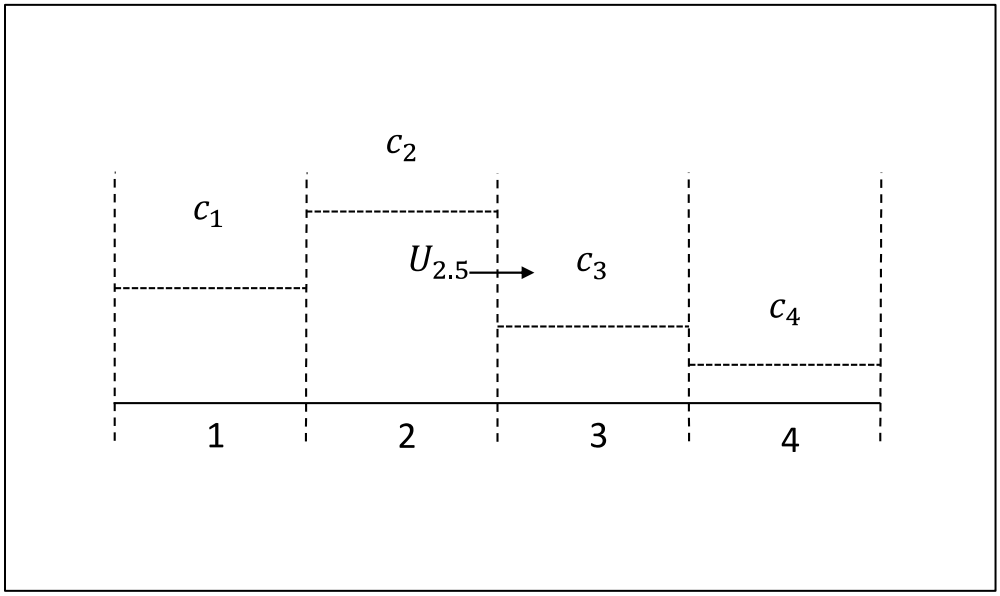
In AdH-SW3, the node-centered edge fluxes are derived in a manner to implicitly satisfy local conservation about an element. The conservation differential equation discrete solution then enforces that the edge fluxes are unique.

As an example, consider the transport equation in conservative form.

$$\frac{\partial c}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{v}c) = 0, on\ \Omega \tag{37}$$

In a finite volume approach, often the concentration is defined at the center of the cell and the velocities at the edges of the cell. A finite-element-trained developer might think that this means that the concentration is a constant over the cell. If that were true, then having a single velocity at the edge would suggest that there is a mass flux jump between cells (with no point/line source) and the method would not be conservative. Consider Figure 10 showing this inference.

Figure 10. Cell constant concentrations with the velocity between cells 2 and 3 indicated by the term U_2.5.



The finite element view of this finite volume representation is that the cell centered concentration is a constant. It is apparent then that at the interface between cells 2 and 3, there is a single velocity ($U_{2.5}$) and different concentrations on either side in the adjoining cells. In this figure, the concentration in cell 2, (c_2), is higher than the concentration in cell 3, (c_3). This means that concentration mass is disappearing at this interface. This does not conserve mass.

Of course, this is not how a finite volume method would interpret this situation. A finite volume approach would assume that the cell-centered concentration represents the average value in the cell, without specifying the distribution. The concentration flux at the interfaces is defined uniquely at each interface based upon concentrations and velocities in the nearby region. Thus, the concentration flux is defined in a way that it is the same no matter which side of the interface an approach is made; that is,

the flux is unique. The concentration transport equation is then solved discretely in each cell in a manner that satisfies local mass conservation.

Now consider the finite element approach to the concentration transport equation, followed by the method of understanding used in AdH-SW3. A SUPG type representation would be

$$\int_{\Omega_h}\left(\phi_i\frac{\partial c_h}{\partial t}-\boldsymbol{\nabla}\phi_i\cdot(\boldsymbol{v}_h c_h)\right)d\Omega+\oint_{\partial\Omega_h}\phi_i(\boldsymbol{v_h}c_h)\cdot\boldsymbol{n}\,da + \sum_e\left(\alpha l\frac{\boldsymbol{V}_h}{\|\boldsymbol{V}_h\|^{\frac{1}{2}}}\cdot\boldsymbol{\nabla}\phi_{\boldsymbol{i}}\right)\left(\frac{\partial c_h}{\partial t}-\boldsymbol{\nabla}\cdot(\boldsymbol{v}_h c_h)\right)d\Omega_e = 0 \tag{38}$$

The subscript h indicates the approximation of the function by some sort of interpolation. To be consistent with AdH, assume that concentration and velocity are linear and continuous. The functions that describe this distribution are termed the trial functions. Also, the use of upper-case letters in the case of $\boldsymbol{V}_h$ means an elemental constant velocity vector. In the case of AdH, this is a linear approximation within the element. As discussed in Berger and Howington (2002), the coefficients of the perturbation terms in the Petrov Galerkin formulation need to be elemental constants to preserve elemental conservation. Consider a group of elements that make up the domain of a particular test function. The node centered edge flux is defined as follows:

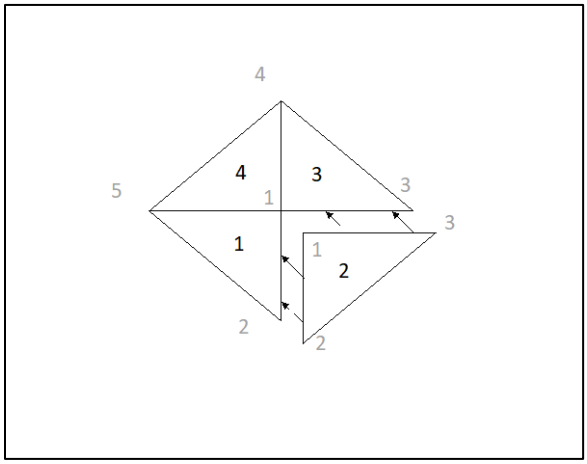
$$N_i^e\equiv-\int_{\Omega_e}\left(\phi_i\frac{\partial c_h}{\partial t}-\boldsymbol{\nabla}\phi_i\cdot(\boldsymbol{v_h}c_h) + \left(\alpha l\frac{\boldsymbol{V}_h}{\|\boldsymbol{V}_h\|^{\frac{1}{2}}}\cdot\boldsymbol{\nabla}\phi_i\right)\left(\frac{\partial c_h}{\partial t}-\boldsymbol{\nabla}\cdot(\boldsymbol{v}_h c_h)\right)\right)d\Omega_e \tag{39}$$

N_i^e is the nodal flux centered at node i in element e. In this viewpoint, the trial functions are used just as a method to derive the node-centered edge flux. The distribution of this computational flux may be more complex than the fluxes that these trial functions suggest. In any case, this computational flux is not the same as one would get from the trial functions. However, at the edges these fluxes obey the definition as stated.

This node flux automatically satisfies local mass conservation. When applied as part of the solution of Equation (38), this flux becomes unique. That is, if the node-centered flux is calculated in the single element or by all the other elements that surround this location, the flux will be identical.

Consider a 2D example shown in Figure 11.

Figure 11. A 2D demonstration.



Element numbers are black, and node numbers are gray. Element 2 is pulled out just for illustrative purposes; it is adjoining the other elements as shown by the arrows.

In terms of the finite element statement, like the discrete equation shown above, the solved equation for node 1 test function is actually

$$-N_1^1 - N_1^2 - N_1^3 - N_1^4 = 0 \tag{40}$$

This means that the numerical enforcement of the finite element statement results in the defined node fluxes being unique. This is apparent when the same equation is solved for the flux N_1^2.

$$-N_1^1 - N_1^3 - N_1^4 = N_1^2 \tag{41}$$

Therefore, the defined node-centered fluxes implicitly conserve mass locally by element, and the finite element statement implementation forces these fluxes to be unique. One note is that it is assumed that there are no point/line/edge inflows. This means the flux jump across edges is zero. If there were to be an edge inflow it would simply be included in the finite

element statement as normal. In the example problem, if there were a line source along the edge of one of our element edges, the numerical solution might be

$$-N_1^1 - N_1^2 - N_1^3 - N_1^4 = \int_\Omega \phi_1 \, q_c(s) d\Omega \tag{42}$$

where, $q_c(s)$ is a line source with values on the arc, s, and zero off the arc. An element edge must follow this arc, and there is a flux jump across the arc.

A further explanation of the finite element statement in relation to the SUPG contribution is warranted. The Galerkin portion of the test function is integrated over the entire domain, Ω, whereas the perturbation test function is integrated over each element and summed. The subtle distinction is that the perturbation test function applies to the element interiors and not over the edges.

The earlier section discussed traditional finite volumes and a one-dimensional example was given. This section concludes using the same example but for the finite element representation but viewed as finite volume.

Figure 12 shows an example of what the computational fluxes might look like from an initial guess of nodal concentrations. These are not the fluxes suggested by the trial functions from interpolation of the values of concentration and velocity. There are two values of these fluxes at each of the interior edges. Since the problem has no designated point sources at the nodes, the flux jump should be zero. The edge computational fluxes should be unique. The solution of the differential equations then enforces that condition, and the solution ends up looking something like Figure 13. (While this figure is used for illustration of an idea, it is possible to calculate these *computational fluxes* in 1D by the same method used to find the end fluxes at the node.)

Figure 12. Computational fluxes based upon some initial estimate of concentrations. The elements are shown by solid black number, and the nodes by gray numbers.

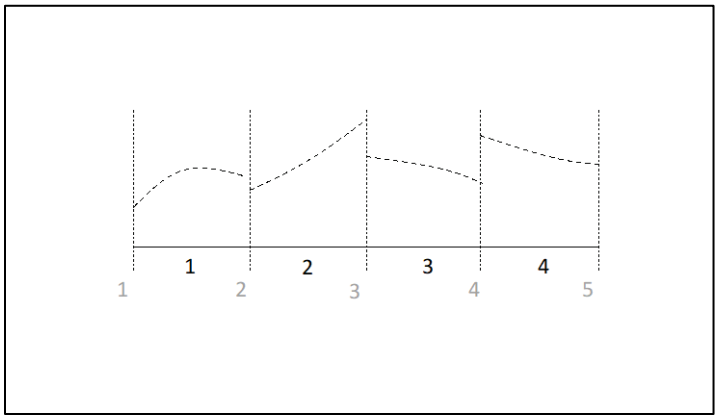


Figure 13. The same problem as shown in Figure 5, but now with the proper solution that there is no flux jump at the element edges.

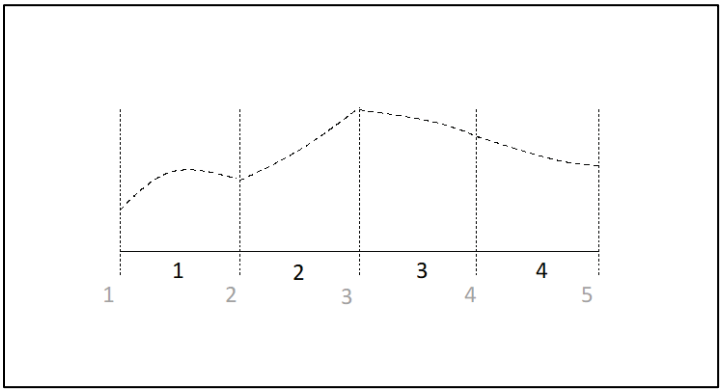


Figure 13 is shown after the solution of the differential equation. The solution forces the computational fluxes to be unique. The distribution of the computational fluxes between the edges (within the element) is just for illustrative purposes. The edge fluxes are all that is being enforced. In the case where there is a point source, there is a flux jump at that node. The flux on one side of the node will differ from that on the other side by the amount of the point source. The flux associated with an element and its end nodes are still unique in that whether calculated from the left or the right, the same value is produced for the flux. However, the value of the point source must be included.

5.3 Test function consistency

The test function should be applied to the entire differential equation. As will be seen later, the AdH test function is made up of two parts. The first

part is the Galerkin test function and is like the shape functions. It is linear within an element and continuous across element boundaries. The second part of the test function is a perturbation function that is made up of the gradient of the Galerkin test function component. The two components make up the entire test function. To maintain higher-order accuracy (higher order convergence rate), the test function must be applied to the entire discrete differential equation. This is a goal in AdH-SW3 but is not achieved perfectly. The perturbation test function is made up of gradients of the Galerkin portion of the test function component and as such is an elemental constant. The shape functions (trial functions) for the velocity and concentrations are linear. The diffusion term requires two spatial derivatives. Therefore, these terms drop out of the discrete differential equation when applied against the perturbation test function component. This will lead to lower convergence rates when diffusion is strong. Fortunately, this is not often the case in the types of flows addressed by AdH-SW3. Also, there are limits that reduce the test function when diffusion is high. The actual application of this SUPG scheme is turned off (or reduced) for cases of small Peclet numbers, where convection is not large compared with diffusion. In any case, this test function consistency is a necessary goal in any future development in AdH-SW3.

5.4 Step 1: Horizontal momentum and depth-integrated continuity

The explanation presented here on Step 1 focuses on three areas:

1. Elemental conservation via discrete consistency. The depth-integrated conservation of mass equation is made up of the vertical summation of the individual conservation equations that are used in Step 2 to find the vertical velocity. This is critical.
2. The stabilization method in AdH-SW3 is a higher dimension version of the AdH-SW2 approach. As such, AdH-SW3 stabilizes the 2D spurious modes similarly to the way AdH-SW2 does. It also stabilizes the higher 3D spurious modes.
3. Finally, there is an explanation of how the pressures are calculated. The vertical distribution of pressure is of a higher order than that of density. Density is linear per element; therefore, the correct calculation of pressure should be elementally quadratic. As such, AdH-SW3 treats pressure as quadratic.

The overall philosophy of the development of AdH-SW3 is discrete consistency. This means that the fluxes that result in conservation are

those coming from the nodal fluxes. These result in locally conservative numerical calculations. Also, the vertical velocity calculations and the depth-integrated calculations of *mass* use the same numerical statements. The depth-integrated statement is the sum of the individual mass conservation statements used to find the vertical velocity. The method to provide a stable numerical scheme, AdH-SW3 uses an approach that is consistent with the AdH-SW2 module. In fact, the 2D result is a type of projection of the 3D method.

The numerical approach for the stabilization of AdH-SW3 is grounded in the SUPG approach. This method began as an application to the transport equation (e.g., Hughes and Brooks [1982]). It can be extended to the 1D shallow water equations as a natural progression by recognizing that this system of equations can be written as two independent transport equations by use of the Riemann invariants as the dependent variables. This type of approach is shown in Berger and Winant (1991) and Hicks and Steffler (1992) for 1D and then extended to 2D in Berger (1992). Note that the 1D extension using the Riemann invariants is not truly possible in multiple dimensions. However, a simpler approximation can be made. The 3D version for AdH-SW3 is found by recognizing that this approach should be an extension of these lower dimension approaches. The 3D should degenerate to the 2D method when projected back to 2D. Similarly, the 1D should be a lower dimension projection of the 2D method.

To understand the development of the stabilization method for AdH-SW3, it is important to first grasp the SUPG method for transport. Transport is a simpler system and easier to understand. For example, consider the finite element statement using SUPG on the 1D transport equation. The trial function is linear and piecewise continuous.

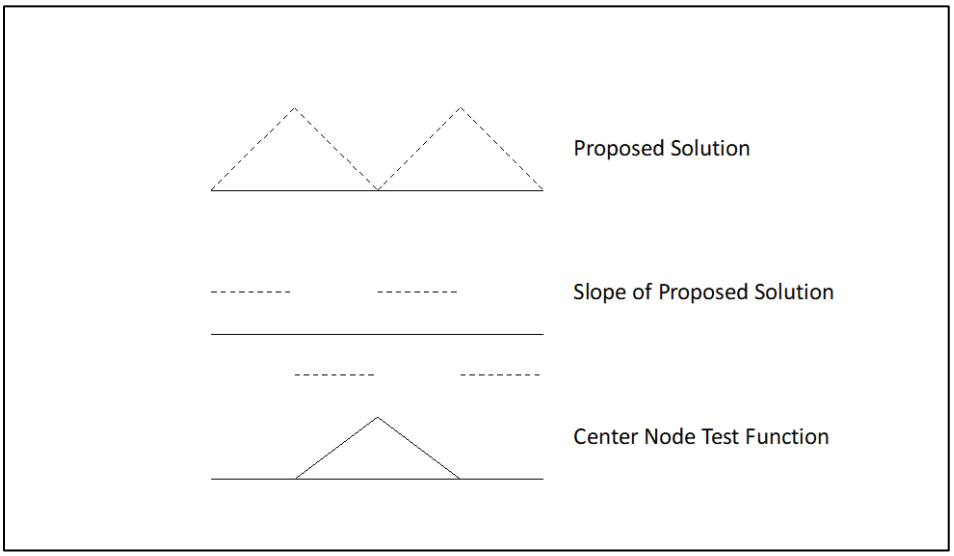
$$\int_{\Omega_h} \phi_i(x)\left(\frac{\partial c_h}{\partial t} + v_{1_h}\frac{\partial c_h}{\partial x}\right)dx + \sum_e \int_{\Omega_e} \alpha l \frac{V_{1_h}}{\left|V_{1_h}\right|}\frac{\partial \phi_i(x)}{\partial x}\left(\frac{\partial c_h}{\partial t} + v_{1_h}\frac{\partial c_h}{\partial x}\right)dx = 0 \tag{43}$$

As a reminder, the subscript h indicates an approximation, for example in terms of concentration it is $c_h \equiv \sum_j \phi_j(x)c_j$. α is a coefficient between 0 and 0.5, and l is an element length. The upper-case letter indicates an element average value. This report will not go into detail why this is stable.

Instead, for general understanding, consider that the standard Galerkin approach for linear, continuous trial functions has a test function that is nominally an *even* function. That is, it is symmetric about node “i”. Spatial solutions that look like a node-to-node oscillation have a spatial derivative that is an *odd* function. An odd function is antisymmetric about node "*i*". The integration of an even function multiplied by an odd function results in zero. That is, the node-to-node oscillation *looks* like a solution to the Galerkin approach. This suggests that the test function needs to be enriched to detect this spurious node-to-node oscillation. This is precisely what is done in SUPG. The perturbation term added to the test function includes the spatial derivative of the trial function. This added perturbation term is an odd function and when scaled properly, provides stability.

Here is a simple example that will illustrate the concept. Consider the simple differential equation, $\frac{dc}{dx} = 0$. If, given a value at a single boundary, this differential equation says that the entire domain is that value given on the boundary. This solution is consistent with the differential equation; the slope in c is 0, or c is a constant. A simple piecewise Galerkin representation, surprisingly, will have difficulty solving this trivial problem. A node-to-node oscillation is a solution to the discrete problem, even though it is not a solution to the original equation. Consider Figure 14.

Figure 14. Demonstration that a proposed oscillatory solution is an actual numerical solution of a Galerkin finite element method.



The top panel in Figure 14 shows a node-to-node oscillation. This is labeled as a “proposed solution.” It is obvious that this is not the true solution. That

is, it is not a flat line of constant value. The representation of this proposed solution in the differential equation is shown in the second panel. This is the slope of the proposed solution. The third panel is the Galerkin test function for the center node. The finite element statement to be satisfied in the numerical method is the following equation.

$$find\ c_h\ for\ all\ \phi_i\ such\ that,$$

$$\int_{\Omega_h} \phi_i\ (x) \frac{dc_h}{dx} dx = 0 \tag{44}$$

This equation means that the test function, shown in the third panel, is multiplied times the slope of the proposed numerical solution, shown in the second panel, and integrated. If this proposed solution is, in fact, a numerical solution, then the result should be zero. Graphically it can be seen that this is true. The integral of the product of these two panels is indeed zero. This node-to-node shows up as a numerical *solution* to the differential equation. It is not a true solution, however. It is a spurious solution. The simple symmetric test function (even function) does not see and cannot control this spurious solution. The SUPG method adds an odd (antisymmetric) perturbation function to the Galerkin test function. If correctly done, this enriched test function sees and controls the spurious mode.

The shallow water equations (in 1D) are a system of equations rather than a single equation with one dependent variable, like the transport equation. The 1D shallow water equations are two equations that have two dependent variables. Each equation includes both dependent variables. For the sake of illustration, consider the linear 1D shallow water equations.

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \boldsymbol{A}_{1_x} \frac{\partial \boldsymbol{Q}}{\partial x} = 0 \tag{45}$$

where

$$\boldsymbol{Q} = \begin{Bmatrix} h \\ v_1 \end{Bmatrix}$$

$$\boldsymbol{A}_{\boldsymbol{1}_x} = \begin{bmatrix} V_1 & H \\ g & V_1 \end{bmatrix}$$

Upper-case letters indicate a constant value. $V_1$is the velocity in the x-direction. The value of acceleration due to gravity, g, is also a constant. The development of this method for Berger and Winant (1991) and Berger (1992) follows from the ideas proposed by Courant, Isaacson, and Rees (1952) for one-sided finite differences. This is generalized in Moretti (1979), Chakravarthy (1979), and Gabutti (1983) as split-coefficient matrix methods. It is also like the generalized flux vector splitting proposed by Steger and Warming (1981). This system of equations can be turned into two transport equations with a change of dependent variables. The stabilization for each of these transport equations is understood. A new variable vector can be created, say $\mathbb{Q}$, such that $\mathbb{Q} = \boldsymbol{PQ} \equiv \begin{Bmatrix} \mathbb{Q}_1 \\ \mathbb{Q}_2 \end{Bmatrix}$. $\boldsymbol{P}$ is chosen so that $\boldsymbol{PA_{1_x}P^{-1}} = \boldsymbol{\Lambda}$. $\boldsymbol{\Lambda}$ is the matrix of the eigenvalues of $\boldsymbol{A_{1_x}}$.

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} V_1 + (gH)^{\frac{1}{2}} & 0 \\ 0 & V_1 - (gH)^{\frac{1}{2}} \end{bmatrix}$$

This results in the two transport equations written in terms of the new variables $\mathbb{Q}$.

$$\boldsymbol{I}\frac{\partial \mathbb{Q}}{\partial t} + \boldsymbol{\Lambda}\frac{\partial \mathbb{Q}}{\partial x} = 0 \tag{46}$$

For convenience, this equation is kept in matrix form. Notice that the first equation contains only the variable $\mathbb{Q}_1$ and the second equation contains only $\mathbb{Q}_2$.

$$\frac{\partial \mathbb{Q}_1}{\partial t} + \lambda_1\frac{\partial \mathbb{Q}_1}{\partial x} = 0$$

$$\frac{\partial \mathbb{Q}_2}{\partial t} + \lambda_2\frac{\partial \mathbb{Q}_2}{\partial x} = 0$$

At this point, the equations are represented in the finite element discretization, and the subscript h is used to indicate the Lagrange polynomial representation. Equation (46) becomes

$$\boldsymbol{I}\frac{\partial \mathbb{Q}_h}{\partial t} + \boldsymbol{\Lambda}\frac{\partial \mathbb{Q}_h}{\partial x} = 0$$

Since the matrix $\boldsymbol{P}$ is composed of constants, the trial (or shape) functions are the same as for the original variables.

The SUPG type stabilization test function used is

$$\psi_i \boldsymbol{I} \equiv \phi_i \boldsymbol{I} + \alpha l_{1d} \frac{\partial \phi_i}{\partial x} \widehat{\boldsymbol{\Lambda}}_h$$

where $\widehat{\boldsymbol{\Lambda}}_h = \begin{bmatrix} \frac{\lambda_{1_h}}{|\lambda_{1_h}|} & 0 \\ 0 & \frac{\lambda_{2_h}}{|\lambda_{2_h}|} \end{bmatrix}$

The equivalent test function in the original variables would be

$$\psi_i \boldsymbol{I} \equiv \phi_i \boldsymbol{I} + \alpha l_{1d} \frac{\partial \phi_i}{\partial x} P_h^{-1} \widehat{\boldsymbol{\Lambda}}_h \boldsymbol{P_h}$$

Here, ϕ_i is used for both the dependent variables. This procedure then extends the SUPG approach to the 1D shallow water system of equations.

This is a method used in HIVEL2D (Stockstill and Berger [1994] or Berger and Stockstill [1995]) and a similar method is used in AdH-SW2 (Berger and Lee (2005). HIVEL2D, of course, is 2D. Extending the 1D approach, above, to 2D can only be done approximately since the x and y coefficient matrices cannot be simultaneously reduced to separate dependent variables. HIVEL2D used unit discharges rather than velocities. AdH-SW2 does not follow this approach but instead uses velocities. AdH-SW2 is used in situations in which the water depth is allowed to go to zero to represent the waterline when wetting and drying. Using unit discharges results in the use of terms in the momentum equation that are divided by depth. This is an obvious problem if the depth shrinks to zero. In terms of the eigenvalue matrices used for the x and y directions, these are the same for unit discharges and velocity dependent variables. That is not surprising since the eigenvalues represent the speed of a free surface wave or the speed that a particle moves.

In HIVEL2D the eigenmatrices were as follows:

$$\mathbf{\Lambda}_{x_h} = \begin{bmatrix} \dfrac{\lambda_{1x_h}}{\left(\lambda_{1x_h}^2 + V_{2_h}^2\right)^{1/2}} & 0 & 0 \\ 0 & \dfrac{\lambda_{2x_h}}{\left(\lambda_{2x_h}^2 + V_{2_h}^2\right)^{1/2}} & 0 \\ 0 & 0 & \dfrac{\lambda_{3x_h}}{\left(\lambda_{3x_h}^2 + V_{2_h}^2\right)^{1/2}} \end{bmatrix}$$

$$\mathbf{\Lambda}_{y_h} = \begin{bmatrix} \dfrac{\lambda_{1y_h}}{\left(\lambda_{1y_h}^2 + V_{1_h}^2\right)^{1/2}} & 0 & 0 \\ 0 & \dfrac{\lambda_{2y_h}}{\left(\lambda_{2y_h}^2 + V_{1_h}^2\right)^{1/2}} & 0 \\ 0 & 0 & \dfrac{\lambda_{3y_h}}{\left(\lambda_{3y_h}^2 + V_{1_h}^2\right)^{1/2}} \end{bmatrix}$$

where

$$\lambda_{1x_h} = V_{1_h} + (gH_h)^{1/2}$$

$$\lambda_{2x_h} = V_{1_h} - (gH_h)^{1/2}$$

$$\lambda_{3x_h} = V_{1_h}$$

$$\lambda_{1y_h} = V_{2_h} + (gH_h)^{1/2}$$

$$\lambda_{2y_h} = V_{2_h} - (gH_h)^{1/2}$$

$$\lambda_{3y_h} = V_{2_h}$$

This SUPG approach approximates *upwinding* along characteristics. For large Froude number flow or very small Froude number flow, this is a very good approximation.

In the early trials of the AdH-SW2 module, this approach was tested. However, the linear iterative solver being used would fail using this method. Instead, AdH-SW2 uses a form of a spectral radius to scale all the eigenvalues. This results in eigenmatrices as follows:

$$\boldsymbol{\Lambda}_x = \frac{1}{(V_1^2 + V_2^2 + gH)^{\frac{1}{2}}} \begin{bmatrix} \lambda_{1x} & 0 & 0 \\ 0 & \lambda_{2x} & 0 \\ 0 & 0 & \lambda_{3x} \end{bmatrix}$$

$$\boldsymbol{\Lambda}_y = \frac{1}{(V_1^2 + V_2^2 + gH)^{\frac{1}{2}}} \begin{bmatrix} \lambda_{1y} & 0 & 0 \\ 0 & \lambda_{2y} & 0 \\ 0 & 0 & \lambda_{3y} \end{bmatrix}$$

There are at least a couple of observations one can note here. First, the scaling used in AdH-SW2 and AdH-SW3 is not as good as that in HIVEL2D. For example, a surge moving in the upstream direction will have an eigenvalue (or wave speed) that represents the potential wave propagating upstream (negative x direction) at a speed of $V_1 - (gH)^{\frac{1}{2}}$. In HIVEL2D, the scaling would result in an upwinding via the Petrov Galerkin perturbation test function that is -1. In AdH, the scaling results in an upwinding much less in magnitude than -1. This could result in more oscillations in water surface and velocity upstream of a surge. The development of AdH-SW2 was under short time constraints, and there was not time to investigate why the iterative solver did not function using the individual eigenvalue scaling used in HIVEL2D. This would be worthwhile to investigate and return to the more robust scaling method.

The second thing to note is that the use of this spectral radius means all the eigenvalues are scaled by a single elemental constant. Therefore, this spectral radius constant can be pulled out, and the original coefficient matrices for the pressure and convection may be used directly. This will be illustrated now for the inviscid shallow water equations.

$$\frac{\partial}{\partial t}\begin{Bmatrix} h \\ v_1 h \\ v_2 h \end{Bmatrix} + \frac{\partial}{\partial x}\begin{Bmatrix} v_1 h \\ v_1 h + \frac{1}{2} g h^2 \\ v_1 v_2 h \end{Bmatrix} + \frac{\partial}{\partial y}\begin{Bmatrix} v_2 h \\ v_1 v_2 h \\ v_2^2 h + \frac{1}{2} g h^2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{47}$$

or in a shorthand, this can be written

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \boldsymbol{\nabla_{2d}} \cdot \boldsymbol{F_2} = \mathbf{0} \tag{48}$$

where, $\boldsymbol{Q}$ is redefined in 2D as

$$\boldsymbol{Q} = \begin{Bmatrix} h \\ v_1 h \\ v_2 h \end{Bmatrix}$$

and $\boldsymbol{\nabla_{2d}}$ is the 2D grad operator.

$$\boldsymbol{F_2} = \begin{Bmatrix} \boldsymbol{F_{2x}} \\ \boldsymbol{F_{2y}} \end{Bmatrix}$$

$$\boldsymbol{F}_{2x} = \begin{Bmatrix} v_1 h \\ v_1 h + \frac{1}{2} g h^2 \\ v_1 v_2 h \end{Bmatrix}$$

$$\boldsymbol{F_{2y}} = \begin{Bmatrix} v_2 h \\ v_1 v_2 h \\ v_2^2 h + \frac{1}{2} g h^2 \end{Bmatrix}$$

This is the conservative form of the shallow water equations. This general form (along with other terms such as viscous effect) is used in AdH-SW2 for the finite element statement in the portion of the test function that is the standard Galerkin part. This is important for enforcing natural boundaries, shock conditions, and enforcing precise conservation.

The nonconservative form shallow water equations are also used in the perturbation portion of the test function. The nonconservative form shallow water equations in 2D are

$$\boldsymbol{E}_{2d} \equiv \frac{\partial \boldsymbol{q}}{\partial t} + \boldsymbol{A_{2x}} \frac{\partial \boldsymbol{q}}{\partial x} + \boldsymbol{A}_{2y} \frac{\partial \boldsymbol{q}}{\partial y} = \boldsymbol{0} \tag{49}$$

where

$$\boldsymbol{A}_{2x} = \begin{bmatrix} V_1 & H & 0 \\ g & V_1 & 0 \\ 0 & 0 & V_1 \end{bmatrix}$$

$$\boldsymbol{A}_{2y} = \begin{bmatrix} V_2 & 0 & H \\ 0 & V_2 & 0 \\ g & 0 & V_2 \end{bmatrix}$$

$$\boldsymbol{q} = \begin{Bmatrix} h \\ v_1 \\ v_2 \end{Bmatrix}$$

The dependent variables h, v_1, and v_2 are represented over space as a combination of elements each of which is interpolated using linear Lagrange polynomials. The distribution is piecewise continuous across elements. (The independent spatial variables are also represented in this manner). For example,

$$h_h(x,y) = \sum_j \phi_j\,(x,y) h_j$$

$$v_{1h}(x,y) = \sum_j \phi_j\,(x,y) v_{1h_j}$$

$$v_{2h}(x,y) = \sum_j \phi_j\,(x,y) v_{2h_j}$$

As before, the subscript h is simply a designation that this is a discretized representation and not the analytic variable.

The nonconservative equation set must be scaled to be dimensionally consistent with the conservative form.

$$\boldsymbol{E}_{H2d_h} \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & H_h & 0 \\ 0 & 0 & H_h \end{bmatrix} \boldsymbol{E}_{2d_h}$$

This matrix multiplies the two momentum equations by the average element depth. The perturbation contribution to the Petrov Galerkin approach for a particular test function "i" then is as follows:

$$\boldsymbol{P}_{2d_i} = \alpha \frac{l_{2d}}{\left(V_{1_h}^2 + V_{2_h}^2 + gH_h\right)^{\frac{1}{2}}} \left(\frac{\partial \phi_i}{\partial x} \boldsymbol{A}_{2x_h} + \frac{\partial \phi_i}{\partial y} \boldsymbol{A}_{\boldsymbol{2y_h}} \right)$$

This can be seen to be a simple system for stabilization. The matrices are scaled by the spectral radius (the magnitude of the largest estimated

eigenvalue). As mentioned previously, a more rigorous method would scale all the eigenvalues individually. This would be more stable but at the expense of more complexity. This more complex method in previous attempts did not work well with the iterative linear solver.

The final Petrov-Galerkin perturbation test function contribution then is

$$\sum_{e} \int_{\Omega_{2d_e}} \boldsymbol{P}_{2d_i} \boldsymbol{E}_{H2d_h} da$$

The entire discrete equation then is

$$\int_{\Omega_{2d}} \left(\phi_i \frac{\partial \boldsymbol{Q}_h}{\partial t} - (\boldsymbol{\nabla}_{2d} \phi_i) \cdot \boldsymbol{F}_{\boldsymbol{2}_h} \right) da + \sum_{e} \int_{\Omega_{2d_e}} \boldsymbol{P}_{2d_i} \boldsymbol{E}_{H2d_h} da \tag{50}$$

$$+ natural\ boundary\ conditions\ = 0$$

This is a shorthand that does not show that each equation's test function may be different at Dirichlet boundary conditions, though the shape of each is identical. At this point, note that the equations are multiplied times the perturbation portion of the test function and integrated over the area of the elements (no element boundary integrals). This is what is meant by the summation over each element rather than an integration over the domain. At first, these may seem like the same thing, but what the summation over the integration of each element infers is that the perturbation portion of the test function is not included in boundary conditions or in defining the jump conditions across elements.

Generally, the 2D method is an extension of the 1D representation. In the original dependent variables, one can see that the stabilization for one equation includes contributions from the other equations. For example, the continuity equation has stabilization terms that are associated with the continuity equation itself but also includes the momentum equation. However, if the dependent variables were the Riemann Invariants, then each equation would include stabilization associated with only that equation. When the stabilization is transformed back to the original dependent variables (velocity and depth), then the apparent intercommunication takes place. This is also true for the original

differential equations. This is precisely the way that one would want to stabilize these equations.

Now consider the 3D shallow water model stabilization. This 3D stabilization must be a higher dimensional representation of the stabilization that has been used in the lower dimensions. The projection of the 3D stabilization to 2D must be a facsimile of the 2D stabilization. This means that the 3D model must be stable in 1D and 2D modes as well as modes that are peculiar to 3D, which shall be called "Reduced Dimension Consistency." That is, the 3D stabilization should also stabilize lower dimension spurious modes as well as those unique to three dimensions. Some items are clear about the 3D stabilization just from basic understanding of hydrodynamics. For an inviscid shallow water equation set, the horizontal velocity is the same over depth. The maximum speed of information movement would be approximately $(\boldsymbol{v} \cdot \boldsymbol{v} + gh)^{\frac{1}{2}}$. Here, the velocity is now in three dimensions, $\boldsymbol{v} = \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix}$.

An important aspect of the 3D module is that the horizontal velocities and the water surface are solved together as a system of equations. The water surface is found by the depth-integrated continuity equation. The vertical velocities are solved purely from continuity. Thus, both the depth-integrated step and the vertical velocity steps are solutions of the continuity equation. It is critical that the two steps be discretely identical. The depth-integrated step then is a direct summation of the individual equations used in the vertical velocity calculation. Failing to be discretely consistent results in a scheme that is not mass conservative. This requirement dictates that the nodes be aligned in vertical columns. By arranging the elements and nodes in vertical columns, the integration over "z" by summing the individual continuity equations is possible. Also, because of the vertical alignment the vertical velocity does not show up in the interior equations when summing over depth. Instead, one has only the kinematic boundary condition at the surface and bed, which produces the water surface elevation instead of the vertical velocity. An additional benefit of the vertical alignment is a direct and less diffusive way to compute the hydrostatic pressure.

An illustration of the vertical alignment of the nodes and the definition of terms is shown in Figure 6. The figure is a 2D representation of an AdH mesh for SW3. The figure is 2D for convenience. The ideas carry through

to three-dimensions. Each vertex represents a node and is labeled i_1, i_2, i_3, . . ., i_{12}. The labels I_1, I_2, and I_3 are the node column numbers. These column *nodes* are the 3D equivalent of the nodes in the normal AdH-SW2 model, that is, the 2D model. If one considers column I_2, it is made up of the column of nodes i_5, i_6, i_7, and i_8. The nonzero test function domain of column I_2 is the entire mesh shown in Figure 6. The test function associated with column node I_2 is the sum of the test functions represented by nodes i_5, i_6, i_7, and i_8. As an aside, for AdH-SW3 the elements do not have to be the same size or distributed evenly in the vertical, though, the nodes must be arranged in a vertical column.

First, consider the continuity equation. The depth integrated continuity equation is solved along with the distributed x and y momentum equations. *Distributed* here means each equation in x and y locations but also the distributed equations in z. (Specifically, they are not vertically integrated.) As a shorthand, the depth integrated continuity associated with column node I_2 would be developed as shown in the following equation.

$$\sum_{i \in I_2} (\text{Eq } 14) = 0$$

In this case, the column node or test function is I_2. Figure 6 shows nodes and test functions $i_5, i_6, i_7, and\ i_8 \in I_2$. The discrete depth integrated continuity equation is then found by summing the contributions from these four nodes/test functions. The details of the continuity equation are now discussed.

The term W_i is now explained.

$$W_i \equiv -\int_{\Omega_h} \boldsymbol{\nabla}\phi_i \cdot \boldsymbol{v_h} d\Omega + \int_{\partial\Omega_{\eta_h}} \phi_i \frac{\partial \eta_h}{\partial t} da_{z_h} - \int_{\partial\Omega_{b_h}} \phi_i \frac{\partial b_h}{\partial t} da_{z_h} + natural\ boundary\ conditions \quad (51)$$

Here, a_z is the area in the x-y plane. The natural boundary conditions include flows in and out of the domain, or a condition of no flow through the boundary. Along the sidewalls, AdH enforces a no-flux condition unless a specified flux is given. The terms involving the temporal derivative of the surface and bottom boundaries arise from the integration by parts. This produces terms related to the flux through the surface and bed. The free surface will move to make sure that no unspecified flux

passes through it. This is accomplished through the kinematic free-surface boundary condition. As discussed earlier, this is the portion of the test function that is applied across element boundaries. As mentioned earlier, the nodes are aligned vertically. This means that the vertical velocity, v_3, does not enter the equations. This is important since this first step is to calculate the horizontal velocity components at every node and the water surface. The vertical velocity is not calculated until the next step. One can see the term $-\int_\Omega \frac{\partial \phi_i}{\partial z} v_{3_h}$ when summed over the vertical column on nodes making up the column test function, the result is zero. This is apparent since ϕ_i is a constant value over the column summation.

Moving to the perturbation contributions, the first term is PWW_i, which is the Petrov-Galerkin stabilization contribution to the depth-integrated continuity equation by the individual distributed continuity equations. PWW_i is defined in Equation (16).

The second summation term is one that is needed for the distributed vertical velocity calculation in a subsequent step. It will be discussed in more detail in that section. Note at this point that this summation will contribute a value of zero to the depth-integrated (depth summed) equation. This is a result of the test function, $\frac{\partial \phi_i}{\partial z}$. When this test function is summed over depth, the value of ϕ_i is a function of x and y but not z, and its derivative with z is then zero. This is a direct result of the nodes vertical alignment.

The first summation is the stabilization term from the continuity equation that contributes to the depth-integrated continuity equation. This has an equivalent term in the 2D SW2 equation set. In fact, these 3D contributions degenerate to the 2D stabilization contributions in what this report terms Reduced Dimension Consistency. One can see this by noting that the perturbation test function is depth independent. It can be pulled outside an integration in "z". The integration in z then just includes the divergence of velocity term. This PWW_i term will then help suppress spurious 2D modes in the same manner that the AdH-SW2 suppression is implemented.

The terms PWM_{x_i} and PWM_{y_i} are the stabilization terms from the x-momentum and y-momentum equations in the continuity equation. These are defined in Equations (17) and (18), respectively. The term within the outermost parentheses in each definition is the associated momentum equation. It is the nonconservative form of the momentum equation. As

discussed earlier in the report, the nonconservative and conservative forms are identical for smooth flow conditions (no shocks present). Since these perturbation-test-function-derived terms apply only within each element, the conditions are always smooth, and the two forms are equivalent. Any non-smoothness is addressed at the element edges. Notice that the perturbation test function is a constant over depth. This results in a very strong reduced dimension consistency. The 3D module will have similar stability as the 2D module for 2D modes.

The individual terms of the perturbation portion of the test function can be shown to be precisely the 2D AdH terms when the velocity is a constant over depth. This will be emphasized by explicitly showing (x, y) after the velocity variables, meaning they can vary horizontally but not in the vertical. That is, the 3D stabilization degenerates to the 2D stabilization when the physical conditions are indeed 2D. To demonstrate this, note that the 3D integrations that sum over each element will first be summed over the vertical. The 3D test functions with subscript i will be summed to become the 2D test function with subscript I.

This leads to the

$$\sum_{i\in I} PWW_i \rightarrow \frac{l_{2d}}{\left(V_{1_h}^2(x,y)+V_{2_h}^2(x,y)+gH_h\right)^{\frac{1}{2}}}\left(V_{1_h}(x,y)\frac{\partial \phi_I}{\partial x}+V_{2_h}(x,y)\frac{\partial \phi_I}{\partial y}\right)$$
$$\left\{\int_{\Omega_{2d_e}}\left(\frac{\partial h_h}{\partial t}+h_h\frac{\partial v_{1_h}(x,y)}{\partial x}+v_{1_h}(x,y)\frac{\partial h_h}{\partial y}+h_h\frac{\partial v_{2_h}(x,y)}{\partial y}+v_{2_h}(x,y)\frac{\partial h_h}{\partial y}\right)d\Omega_{2d_e}\right\}$$

The arrow "→" is used to mean that degenerates into 2D. This is saying the perturbation part of the Petrov Galerkin test function (for the impact of the continuity equation stabilization on the continuity equation) looks just like the 2D contribution under the circumstances of a velocity that is independent of depth. The same can be demonstrated for the terms $PWM_{x_i}$and PWM_{y_i}. The point is that the successful stabilization produced for the 2D AdH-SW2 module should carry over to the 3D module.

Now consider the momentum equations. These are just the two horizontal momentum equations for x and y. (The third momentum component in the z direction is not present since the vertical inertia terms are assumed to be negligible. This is the hydrostatic assumption.)

The x-momentum equation is then given by Equation (19), and the y-momentum equation by Equation (23).

The momentum equations are slightly different. The continuity equation in 3D is still depth integrated. The individual momentum equations are not. However, to demonstrate the 2D analogue of the 3D stabilization, the elements will be summed over the vertical. For simplicity, this will be illustrated using the term $PM_x W_i$

$$\sum_{i \in I} PM_x W_i \rightarrow \sum_e \frac{\alpha l_m g H_h}{\left(V_{1_h}^2(x,y) + V_{2_h}^2(x,y) + V_{3_h}^2(x,y,z) + gH_h\right)^{\frac{1}{2}}} \frac{\partial \phi_I}{\partial x} \int_{\Omega_{2d_e}} \left(\frac{\partial h_h}{\partial t} + h_h \frac{\partial v_{1_h}(x,y)}{\partial x} + v_{1_h}(x,y) \frac{\partial h_h}{\partial y} + h_h \frac{\partial v_{2_h}(x,y)}{\partial y} + v_{2_h}(x,y) \frac{\partial h}{\partial y} \right) d\Omega_{2d_e}$$

This is saying that this stabilization in 3D degenerates to its 2D equivalent. This is not a perfect projection since the l_m term is somewhat different than the l_{2d} term, and it varies by element in the vertical column. Another slight difference is that the vertical velocity is not zero. However, it is stable for the 2D modes. The equations are also stabilized for 3D modes. The momentum equations include terms involving the vertical velocity and length scales that are derived from 3D.

The tolerance of the iterative Newton method should be considered in the implementation of mass conservation. Keep in mind that Step 1 is solved in an iterative fashion because the equations are nonlinear. A Newton type method addresses the nonlinearity and results in linear matrix problem that is solved repetitively to drive the residual to nearly zero. The original nonlinear equations are then satisfied to some tolerance. Notice that the Galerkin-related component of the test function for conservation of mass is linear. If this were the only part of the conservation of mass equation, then the conservation of mass discrete equation would be satisfied to machine precision at every iteration. To understand this the reader can form an equation set composed of one linear equation and one nonlinear equation each sharing two variables. Each iteration of this contrived set will satisfy the linear equation while the nonlinear equation will only be satisfied to some tolerance.

Here, for example, is a two-equation/two-unknown equation set:

$$x + y = 0$$
$$3x^2 + y^2 - 4 = 0$$

The first equation is linear, and the second is nonlinear. Using a Newton-type method to converge this system yields the following linearized system:

$$\begin{bmatrix} 1 & 1 \\ 6x_k & 2y_k \end{bmatrix} \begin{Bmatrix} \Delta x \\ \Delta y \end{Bmatrix} = \begin{Bmatrix} -R_{1_k} \\ -R_{2_k} \end{Bmatrix}$$

Here, the new iterate is found by solution of this linear equation set and using $x_{k+1} = x_k + \Delta x$ and $y_{k+1} = y_k + \Delta y$. Here, the subscript k indicates the iterate number. This table gives the values of each solution iterate and the residuals. The calculations use a starting set of values of $x_0 = 2$ and $y_0 = 2$. Table 1 shows the results. The fourth column contains the residuals for the linear equations, and the fifth column contains the residuals for the nonlinear equations.

Table 1. The values of the unknowns *x* and *y* for each iteration *k*.

k	x_k	y_k	R_{1_k}	R_{2_k}
1	5/2	-5/2	0	21
2	71/20	-71/20	0	164/400

Table 1 shows the values of the unknowns x and y for each iteration k. The residual for the first (linear) equation is shown in the column identified as R_{1_k}. The residual for the second (nonlinear) equation is shown in the column identified as R_{2_k}.

After the initial guess, all the residuals for the linear equation are zero. The nonlinear equation converges toward zero but is not zero.

In the discrete conservation of mass equation solved in AdH-SW3, there are some nonlinear terms. These are the PWM_{x_i} and PWM_{y_i} terms. Notice that these are a fraction of the discrete momentum equations (Equation (19) and Equation (23)). Still, the nonlinearity of the discrete conservation of mass equation (Equation (14)) is much milder than those of the discrete momentum equations. By the time the momentum equation residuals are iterated to some tolerance, the residual of the conservation of mass equation is orders of magnitude smaller than the tolerance.

Additionally, the Petrov-Galerkin perturbation terms (PWW_i, PWM_{x_i}, and PWM_{y_i}) all cancel out if summed over an element. The test function is composed of a gradient of the Galerkin test function over some spatial dimension, the Galerkin test function sums to a constant over an element, and the coefficients are elemental constants. The nonlinear effects of these terms over an element are zero. This means that elementally the Galerkin portion of the test function is all that remains. The elemental equation is composed solely of C_{d_i}, which is linear. Elemental mass conservation is assured. However, since the discrete problem may not be solved exactly (non-zero residual in the iterative solution), the flux jump across elements will also likely be small, but not zero. Therefore, the tolerance should be tight. However, keep in mind that mass conservation will be solved much more precisely than momentum.

A discussion of the way that pressure is calculated and used within AdH-SW3 is warranted. Density is calculated from salinity, temperature, and, possibly, sediment concentrations. The concentration of each of these variables is available at the nodes and interpolated linearly within the element. The equation of state used in AdH-SW3 is a linear relationship between density and the concentration variables. Pressure in AdH-SW3 is hydrostatic. That is, the pressure is composed of the weight of the water above a location per unit area plus atmospheric pressure. The integration from the free surface down to this location yields the pressure. The pressure then is an integration of density and gravity in the z direction. The result is that pressure is quadratic in z. AdH-SW3, therefore, uses a quadratic basis for pressure. The value of pressure is determined at the vertex nodes (element corners) and at additional nodes located at the mid-point along each element edge. This results in 10 nodes for tetrahedral elements in 3D and 6 nodes on vertical plane triangular surface elements.

The pressure at each quadratic basis node is calculated at the beginning of each Newton iteration. It is difficult to include the pressure calculation within the numerical differentiation step within the Newton Method. The calculation on a single element at some depth would need to utilize information from all the elements above in its column. This can be quite computationally expensive for a largely unstructured mesh code. Therefore, the pressure and the perturbed pressure due to changes in water surface are calculated at the beginning of each Newton iteration. This calculation is done in a column-structured part of AdH-SW3.

5.5 Step 2. Vertical velocity calculation

The explanation in this step is focused on two items:

1. The continuity (conservation of water mass) equation is a first-order differential equation. Therefore, only one boundary condition can be applied if the boundaries are defined. This means that while both surface and bed kinematic boundary conditions are applied in the depth integrated step (Step 1), only the one kinematic boundary condition is applied in Step 2. Step 2 must be absolutely discretely consistent with Step 1 to be conservative.

2. The calculation of the vertical velocity is made by summing all the individual discrete water mass conservation equations, beginning at the free-surface boundary condition. The summations proceed down each column of nodes. As described above, AdH-SW3 uses a quasi-unstructured mesh (some structure in the vertical direction), and it is implicit. To make this type of calculation work, an additional term is added to the equations. This term creates a difference-type stencil that allows the calculations to not be influenced by values below the vertical velocity being calculated. (Actually, there is a choice. Either use the free-surface boundary condition and proceed downward through the column of test functions or choose the bed boundary condition and proceed upward through the column. In AdH-SW3, the free surface is used.)

The vertical velocity equation (Equation (27)) is the previously mentioned continuity equation for the individual distributed test functions. This solution uses a free surface boundary condition. The free-surface and bed boundary conditions cannot both be applied as this is merely a first-order differential equation. Trying to do so would over-specify the differential equation. Instead, AdH-SW3 relies on the fact that the water surface elevation is determined by both boundary conditions in Step 1. AdH-SW3 then requires that the continuity equation in the calculation for the water surface (depth-integrated equation) and the individual continuity equations used to calculate the vertical velocity are identical finite element statements. This is critical. By doing this, AdH-SW3 can sum the individual distributed finite element continuity statements beginning at the surface to create the vertical velocities. When AdH-SW3 reaches the bed, moving down the column of test functions, the solution will automatically satisfy the bed kinematic boundary condition. The last term in the PWW_i (see Equation (16)) is not actually a stabilization term. It is a

term that is added to force the summation process from surface to bottom in an unstructured finite element mesh. AdH-SW3 uses a solution to the differential equations all solved at once to mimic what should be a sum of all the elements above a nodal location to result in the vertical velocity there. The negative sign in the term forces a one-sided difference stencil. The result is that the method does not *need* the information below a point. It only needs the surface kinetic boundary condition and the nodal contributions above this node. If one had chosen to use the bed kinematic boundary condition instead of the free surface, then the sign on this integral would have been positive. If one considers finite difference stencils, one can see that with strong enough upwinding the difference method for a first-order differential equation becomes one-sided. When summed over the column, as it is in the depth-integrated continuity equation, the full depth test function, ϕ_I, is not a function of z. This extra upwinding term make zero contribution in the depth-integrated continuity equation.

5.6 Step 3. Transport

The stabilization used in transport follows the same principles explained in the hydrodynamic steps. Although, for the transport equation, only a single concentration field must be calculated at a time, rather than a system of coupled equations like Step 2. The additional item that will be addressed here are the $\sum_e \ C_h PWW_i + \sum_e \ C_h PWM_{x_i} + \sum_e \ C_h PWM_{y_i}$ terms from Equation (29).

Notice that these amount to the Petrov-Galerkin perturbation terms from the water mass conservation equation but multiplied times each elemental average concentration of the constituent that is being calculated. These terms are an extension from a long-term practice used in AdH-SW2.

The terms were added in AdH-SW2 by the author because of observations of the behavior of calculated concentrations of a self-contained lake. The lake had no inflow or outflows. The water surface was perturbed and allowed to slosh. The concentration field was initiated at a constant value. As the water in the lake sloshed, the concentration should have remained fixed at the originally assigned constant concentration. Instead, small wiggles or oscillations in concentrations appeared. On average, these were correct but oscillated about this true constant value. How this happens is illustrated next.

The expansion of the conservative form 2D shallow water transport equation is

$$\frac{\partial hc}{\partial t} + \boldsymbol{\nabla}_{2d} \cdot (\boldsymbol{v}hc) = c\left(\frac{\partial h}{\partial t} + \boldsymbol{\nabla}_{2d} \cdot (\boldsymbol{v}h)\right) + h\left(\frac{\partial c}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{2d} c\right) \quad (52)$$

For this equation, $\boldsymbol{v}$ is the 2D velocity vector. From Equation (52), one can see that the first term within the parentheses on the right side of the equation is the conservation of water mass for the shallow water equations. The second term in parentheses is transport of the constituent. For a concentration that is a constant, the last term will be zero. The first term in parentheses will be zero if water mass is conserved. For the discrete finite element shallow water constituent mass conservation equation, these terms must be computed precisely as they are in the shallow water mass conservation equation. As such, the discrete constituent mass conservation equation must include the Petrov-Galerkin perturbation test function contribution to water mass conservation. These must be multiplied by the elemental constant value of constituent concentration.

For the AdH-SW3 discrete equations, this is precisely what is being done in 3D. Each water mass stabilization term is included but multiplied by the elemental constant, C_h. There should be an investigation to see if an elemental constant for concentration is necessary for the multiplier or could the distributed value of concentration be used instead. If the starting concentration is a constant, then a constant would result automatically in the discrete equations, even if the linearly distributed value of c_h were used. This is an area that could use further investigation.

6 Further Investigations and Challenges

Throughout this report, several items were flagged to indicate where additional investigation could potentially improve AdH-SW3. Also, some items were given that would be improvements in the model. These are left to other interested engineers and scientist to evaluate and resolve. These items were discussed earlier in the report, and this location is referenced with each.

1. AdH-SW3 could incorporate a temporal difference that includes the change in time-step size. The method is shown in the section "Time Advancement" of the chapter "The Discrete Model." This modification would need to be made for water surface, velocity, constituents, and the mesh movement.
2. Improvement of the manner that sidewalls are handled in AdH-SW3 would be an important investigation. As discussed in section "Computation Sequence" of chapter "The Discrete Model." The sidewalls are handled by solving the momentum equations for a direction that is parallel to the boundary. The method sometimes gives erratic results if the sidewall angle changes abruptly. As discussed in this section, something more like the natural boundary condition for the sidewalls used in AdH-SW2 would be better. This is not directly applicable to AdH-SW3 since there is no pressure term at each individual node over depth to change to force momentum and mass conservation. Since AdH-SW3 is hydrostatic, the only pressure variable that is available is the change in the water surface elevation. The pressure cannot vary over depth to satisfy conservation as a natural boundary would enforce. There likely would need to be a node-based constraint on the flow but using the same natural boundary approach.
3. The 3D as well as the 2D stabilization could be improved if the approach used by the HIVEL2D model would function with the iterative solver used in AdH-SW3 and AdH-SW2. HIVEL2D is an ancestor code of the AdH-SW2 module in that the stabilization was patterned after that of HIVEL2D. HIVEL2D scales the test function upwinding for each eigenvalue. HIVEL2D used a direct solver and functioned well. The advantage of scaling each eigenvalue is that the control of oscillations due to shocks and rapid changes is much better. The oscillations are smaller. Unfortunately, the iterative solver used in AdH would not converge using this approach. Instead, AdH-SW2 and AdH-SW3 scale all the eigenvalues by a single *spectral radius* value.

Individual eigenvalue scaling would be a great benefit and worth further study. A discussion is given in section "Step 1. Horizontal momentum and depth-integrated continuity" of chapter "Discussion of the Discrete Model."

4. In section "Step 2. Vertical velocity calculation" of chapter "Discussion of the Discrete Model," there is a discussion about the calculation of the individual vertical velocities at each node over a column proceeding from the water surface to the bed. This is done in a manner to be consistent with the horizontal velocity and depth-integrated continuity equation calculations. Presently, AdH-SW3 calculates the vertical velocity by using a water surface kinematic boundary condition to find the other vertical velocities. Since the method is fundamentally consistent with the horizontal velocity and depth-integrated continuity, the bed kinematic boundary condition is automatically satisfied. There is an additional term that is added to these calculations to give a direction to the calculations. It is also possible to reverse the sign of this additional term and use the bed kinematic boundary condition and not the water surface kinematic boundary condition. It is not obvious if one is superior to the other. This could be investigated further.
5. In section "Step 3. Transport" of chapter "Discussion of the Discrete Model," a method is described that controls oscillations resulting from the hydrodynamic mass conservation (see Equation (52)). The coefficient in this method uses an elemental constant for concentration. It is not apparent that this is necessary. Instead, one could investigate if using the normal linear distribution of concentration could be used.

The most important item to address is the sidewall boundary approach. The stabilization issue listed as "3" and the time-step listed as "1" would then be the next most important.

7 Conclusions

The overall theme of the AdH-SW3 development is one of design based upon the type of problem to be solved and experience with previous model development. AdH-SW3 is configured to be able to address stratification issues peculiar to navigation channels in estuaries. AdH-SW3 is an unstructured approach, and as such the mesh can be aligned in horizontal layers in a navigation channel. This prevents the steep side slopes from causing salinity to leak from the channel artificially. This requires the model user be knowledgeable and create this type of mesh in that environment, but AdH-SW3 can easily handle this condition.

In the computational development, the overall approach is one of consistency in all formats. First, AdH-SW3 is locally mass conservative about each element. This is a result of the use of the conservation form of the differential equations and the understanding the model must be consistent with the discrete nodal fluxes.

Additionally, AdH-SW3 attempts to be consistent with the test function. This means that the entire differential equation is integrated against the entire test function. This is not totally possible. For example, since the variables such as velocity and concentration are linear, they become zero when a second-order spatial derivative is applied to them. In the case of the turbulent diffusion terms, this becomes necessary when applying the perturbation part of the test function, but it is an important goal to try to meet since this makes the scheme a higher-order method. This means the numerical solution converges to a higher order as the mesh/time-step is refined.

AdH-SW3 is Reduced Dimension consistent as well. This means that the 3D model is consistent with the 2D stabilization of AdH-SW2. AdH-SW3 stabilizes the oscillatory 2D modes that the 2D model faces but also stabilizes the 3D modes. The stabilization scheme in AdH-SW3 reverts to the 2D stabilization scheme of AdH-SW2 when the 3D solution is projected down to the 2D. This is beneficial since the AdH-SW2 scheme has been very successful. AdH-SW3 builds on that approach but also stabilizes additional 3D spurious modes.

One very important consistency in AdH-SW3 is the two first split step calculations. In the first step, the horizontal velocity vectors at each node

and the water surface elevations are calculated. The water surface elevation is found by summing the individual finite element equations over the water column. This is a vertical line of nodes and test functions. This summation forms the depth-integrated conservation of water mass equation. There is a free-surface and a bed kinematic boundary condition as well. Together, these determine the location and movement of the free surface. The next step involves calculating the vertical velocity at each node in the column and over the entire mesh. AdH-SW3 uses the identical mass conservation equations and a single kinematic boundary condition to determine all the vertical velocities. This works only if the same discrete conservation of water mass equations is used in both steps, that is, in the solution for the free surface and the subsequent calculation of vertical velocity.

The variables that make up AdH-SW3 such as concentration are linearly interpolated within each element. AdH-SW3 is a hydrostatic code, which means that the weight of the water above a point completely determines the pressure. When transported variables such as temperature and salinity are used, they impact the density. The pressure approximation then must be at least quadratic. That is, in fact, what is done in AdH-SW3. The model is consistent with the interpolation of variables.

Finally, the report suggests possible improvements in the model. These typically will involve some preliminary investigation. These suggestions are included throughout the report. The most important one of these is to address the manner that the sidewalls boundary conditions are implemented. AdH-SW3 cannot rely only on the natural boundary conditions like AdH-SW2 does. Since there is only one water surface elevation per nodal column, all the column cannot be forced to conserve momentum. One should be able to apply constraints to the wall to enforce the no-momentum flux condition at each node in the column. Then there would be the additional equation to solve for the constraint. However, this deserves further investigation.

References

Berger, R. C. 1992. *Free-Surface Flow over Curved Surfaces*. PhD dissertation. Austin, TX: University of Texas at Austin.

Berger, R. C., and S. E. Howington. 2002."Discrete Fluxes and Mass Balance in Finite Elements." *ASCE, J. Hydr. Engr*. 128(1): 87–92. DOI: 10.1061/(ASCE)0733-9429(2002)128:1(87).

Berger, R. C., and L. M. Lee. 2005. *Multidimensional Numerical Modeling of Surges Over Initially Dry Land*. ERDC/CHL TR-04-10. Vicksburg, MS: US Army Corps of Engineers, Engineer Research and Development Center.

Berger, R. C., and R. L. Stockstill. 1995. "Finite Element Model for High Velocity Channels." *ASCE, Journal of Hydraulic Engineering* 121(10): 710–716.

Berger, R. C., and E. H. Winant. 1991. "One-Dimensional Finite Element Model for Spillway Flow." *Hydraulic Engineering, Proceedings, 1991 National Conference, ASCE*, Nashville, TN, July 29–August 2, 1991. Edited by Richard M. Shane. New York: 388–393.

Chakravarthy, S. R. 1979. *The Split-Coefficient Matrix Method for Hyperbolic Systems of Gasdynamic Equations*. PhD dissertation. Ames. IA: Department of Aerospace Engineering, Iowa State University.

Courant, R., E. Isaacson, and M. Rees. 1952. "On the Solution of Nonlinear Hyperbolic Differential Equations." *Communications on Pure and Appl. Math.* 5: 243–255.

Gabutti, B. 1983. "On Two Upwind Finite Difference Schemes for Hyperbolic Equations in Non-Conservative Form." *Computers and Fluids* 11(3): 207–230.

Hicks, F. E., and P. M. Steffler. 1992. "Characteristic dissipative Galerkin scheme for open-channel flow." *Journal of Hydraulic Engineering, ASCE* 118(2): 337–352.

Hirsch, C. 1988. *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*. Chapter 6, "Finite Volume Method and Conservative Discretizations." Hoboken, NJ: John Wiley & Sons.

Hughes, T. J. R., and A. N. Brooks. 1982. "A Theoretical Framework for Petrov-Galerkin Methods with Discontinuous Weighting Functions: Applications to the Streamline-Upwind Procedures." *Finite Elements in Fluids*. Edited by R. H. Gallagher. London: J. Wiley and Sons.

Hughes, T. J. R., G. Engel, L. Massei, and M. G. Larson. 2000. "The Continuous Galerkin Method is Locally Conservative." *J. Comput. Phys.* 163(2): 467–488.

Johnson, B. H., R. H. Heath, B. B. Hsieh, K. W. Kim, and H. L. Butler. 1991. *Development and Verification of a Three-Dimensional Numerical Hydrodynamic, Salinity, and Temperature Model of Chesapeake Bay: Volume 1, Main Text and Appendix D*. Technical Report HL-91-7. Vicksburg, MS: US Army Corps of Engineers, Waterways Experiment Station.

King, I. P. 1982. *A Finite Element Model for Three-Dimensional Flow*. For USAE Waterways Experiment Station, Vicksburg, MS. Lafayette, CA: Resource Management Associates.

Moretti, G. 1979. "The λ-Scheme." *Computers in Fluids* 7(3): 191–205.

Stagg A., J. Hallberg, and J. Schmidt. 2000. "A Parallel, Adaptive Refinement Scheme for Tetrahedral and Triangular Grids." *Parallel and Distributed Processing. IPDPS 2000*. Lecture Notes in Computer Science, vol 1800. Edited by Rolim J. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45591-4_69.

Stockstill, R. L., and R. C. Berger. 1994. *HIVEL2D: A Two-Dimensional Flow Model for High-Velocity Channels*. Technical Report REMR-HY-12. Vicksburg, MS: USAE Waterways Experiment Station.

Tezduyar, T. E., and Y. J. Park. 1986. "Discontinuity-Capturing Finite Element Formulations for Nonlinear Convection-Diffusion-Reaction Equations." *Computer Methods in Applied Mechanics and Engineering* 59(3): 307–325. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/0045-7825(86)90003-4. http://www.sciencedirect.com/science/article/pii/0045782586900034.

Thompson, J. F. (Ed.). 1982. "Numerical Grid Generation." *Proceedings of a Symposium on the Numerical Generation of Curvilinear Coordinate Systems and Their Use in the Numerical Solution of Partial Differential Equations*, April 1982, Nashville, TN.

Acronyms and Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
ADH	Adaptive Hydraulics/Hydrology
AdH-SW2	Adaptive Hydraulics/Hydrology 2D Shallow Water model
AdH-SW3	Adaptive Hydraulics/Hydrology 3D Shallow Water model
SUPG	Streamline upwind Petrov-Galerkin

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED *(From - To)*
June 2022.	Final Report	FY21–FY22

4. TITLE AND SUBTITLE
Foundational Principles in the Development of AdH-SW3, the Three-Dimensional Shallow Water Hydrodynamics and Transport Module within the Adaptive Hydraulics/Hydrology Model

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

R. C. Berger

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Coastal and Hydraulics Laboratory
US Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

8. PERFORMING ORGANIZATION REPORT NUMBER
ERDC/CHL TR-22-9

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army Corps of Engineers, Engineer Research and Development Center
Vicksburg, MS 39180-6199

10. SPONSOR/MONITOR'S ACRONYM(S)
ERDC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
MIPR W74RDV12944824

14. ABSTRACT
This report details the design and development of the three-dimensional shallow water hydrodynamics formulation within the Adaptive Hydraulics/Hydrology model (AdH-SW3) for simulation of flow and transport in rivers, estuaries, reservoirs, and other similar hydrologic environments. The report is intended to communicate principles of the model design for the interested and diligent user. The design relies upon several layers of consistency to produce a stable, accurate, and conservative model. The mesh design can handle rapid changes in bathymetry (e.g., steep-sided navigation channels in estuaries) and maintain accuracy in density-driven transport phenomena (e.g., thermal, or saline stratification and intrusion of salinity).

15. SUBJECT TERMS
Computer simulation, Hydraulic models, Hydrodynamics, Hydrology, Numerical analysis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jennifer N. McAlpin
Unclassified	Unclassified	Unclassified	SAR	73	19b. TELEPHONE NUMBER *(Include area code)* 601-634-2511