Contents lists available at ScienceDirect

Coastal Engineering

journal homepage: www.elsevier.com/locate/coastaleng

An exact reanalysis technique for storm surge and tides in a geographic region of interest

John Baugh [a,*], Alper Altuntas [a], Tristan Dyer [a], Jason Simon [b]

[a] Department of Civil, Construction, and Environmental Engineering, North Carolina State University, Raleigh, NC, USA
[b] Department of Civil and Environmental Engineering, University of California, Berkeley, CA, USA

ARTICLE INFO

Article history:
Received 2 July 2014
Received in revised form 1 December 2014
Accepted 3 December 2014
Available online 7 January 2015

Keywords:
Hurricane
Storm surge
Subdomain modeling
ADCIRC

ABSTRACT

Understanding the effects of storm surge in hurricane-prone regions is necessary for protecting public and lifeline services and improving resilience. While coastal ocean hydrodynamic models like ADCIRC may be used to assess the extent of inundation, the computational cost may be prohibitive since many local changes corresponding to design and failure scenarios would ideally be considered. We present an exact reanalysis technique and corresponding implementation that enable the assessment of local *subdomain* changes with less computational effort than would be required by a complete resimulation of the full domain. So long as the subdomain is large enough to fully contain the altered hydrodynamics, changes may be made and simulations performed within it without the need to calculate new boundary values. Accurate results are obtained even when subdomain boundary conditions are forced only intermittently, and convergence is demonstrated by progressively increasing the frequency at which they are applied. Descriptions of the overall methodology, performance results, and accuracy, as well as case studies, are presented.

1. Introduction

Coastal storms and hurricanes produce surge, flooding, and wave actions that have damaging effects on the built and natural environment. A key limiting factor in moving from the science of storm surge modeling to its practical application in engineering infrastructure assessment is one of scale: storm surge models necessarily operate over large parts of the globe because that is the scale at which storms and hurricanes operate. However, from a critical infrastructure perspective, assessment of performance and overall resilience happens over a much smaller geographic region, perhaps dealing with individual components, such as levees and seawalls, and their collective behavior. And yet surge loads, as well as other hurricane effects, are necessary to simulate effects on infrastructure at a more local level.

From an engineering perspective, assessment of infrastructure and its resilience necessitates many such large-scale simulations for a single hurricane event: this need is due to the nature of engineering analysis and design. First, one must consider different designs, arrangements, configurations, and materials, each time subjecting the system to a given storm load. Second, there are numerous failure scenarios to consider even for a single infrastructure system, such as levees: for each breach location and type, and for all such breaches in their various combinations, the impact of flooding and other damage must be assessed in order to make collective risk and reliability judgments. These quantitative results can then be used to improve the design and layout of critical infrastructure components and overall system-wide resilience.

To address the particularly important and computationally limiting issue of differences in scale, we present a workflow enabled by a new boundary condition type that combines water surface elevation, velocity, and wet/dry status for unstructured, finite element grids. As in reanalysis techniques for structural systems (Arora, 1976; Kassim and Topping, 1987), our *subdomain modeling* approach for storm surge simulation allows new results to be determined after a local modification to a grid with less effort than would be required to run the entire simulation again. Components of the workflow include a graphical user interface for extracting a geographic region of interest from a grid, the implementation of a new boundary condition type in ADCIRC (Luettich et al., 1992), a widely used storm-surge modeling code, and a low-cost verification step for comparing the results of simulations conducted on an extracted subdomain with those of the original, full domain, both before and after local modifications. The approach is exact in the sense that subdomain runs produce the same results as those that would be obtained by an equivalent simulation on a full domain, so long as the subdomain is large enough to fully contain the altered hydrodynamics.

To demonstrate the value of this approach, consider the array of possible levee failures simulated by (Simon, 2011) in a hypothetical coastal community, as shown in Fig. 1, with flooded areas outlined in white over a local street network. The process begins with an initial large-scale simulation of a storm event followed by subsequent local

[*] Corresponding author.
E-mail address: jwb@ncsu.edu (J. Baugh).

http://dx.doi.org/10.1016/j.coastaleng.2014.12.003
0378-3839/© 2014 Elsevier B.V. All rights reserved.

simulations, each with varying local topographies representing possible levee failure scenarios (which might be determined from other engineering analysis techniques). While the initial simulation requires a thousand CPU-hours of run-time on our setup, a subsequent local simulation takes orders of magnitude less time, since run times are proportional to grid size. Of course, as with any modeling exercise, working with subdomains requires some judgment in extracting regions large enough to contain the changes, yet small enough to minimize redundant computations. In practice, subdomains of various sizes and geometries can be extracted simultaneously from a full run and verified both before and after modifications are made.

The following sections provide background on large-scale and regional approaches for tide and surge simulation, and then move on to the subdomain modeling approach and its realization as a new boundary condition type combining water surface elevation, depth averaged water velocity, and wet/dry status (Altuntas, 2012). The approach is shown to be both accurate and well posed via testing on small, benchmark problems, as well as on large-scale simulations with subdomains of various sizes and geometries along the North Carolina coast. In later sections, we describe a user interface that is designed and implemented to support subdomain modeling, and then conclude with some general observations and ideas for future work.

2. Background

Blain et al. (1994) demonstrate the necessity and effectiveness of large-scale approaches for tide and surge modeling. They examine combinations of domain size and boundary conditions for their effects on computed storm surge characteristics in a study of Hurricane Kate (1985) along the Florida coast. For domains, the authors consider a small, semi-circular region surrounding Panama City, a larger region including the Gulf of Mexico, and a much larger one encompassing the entire western North Atlantic Ocean, the Caribbean Ocean, and the Gulf of Mexico. Combined with those domains, two different open ocean boundary forcing functions are considered: a still water boundary condition where water elevation is set equal to the mean sea level, and a boundary condition that partially accounts for meteorological forcing by imposing an inverted barometer effect. The authors conclude that small domains, whose boundaries are in regions where surge effects appear, are inadequate because those effects on the boundaries cannot be known, and hence enforced, a priori. Even somewhat larger domains, like the Gulf of Mexico, may cause difficulties if boundary conditions are to capture the effects of resonant modes, which cannot easily be quantified due to interactions with neighboring basins. Since the open ocean boundaries of the largest domain are far enough removed from the continental shelf and the Gulf of Mexico basin, the influence of the boundary condition specification is minimal, and such domains are found by the authors to be the most practical and effective.

Dietsche et al. (2007) use Hurricane Hugo (1989) to evaluate the effect of incorporating inundation areas and inland flooding in a computational domain. They present four different grids around Winyah Bay, South Carolina, along with a large-scale grid encompassing the western North Atlantic Ocean. In the study, the open ocean boundary of the large-scale grid is elevation-forced using harmonic data corresponding to tidal constituents. They observe that integrating coastal floodplains can significantly alter storm surge predictions, and that using a large-scale domain helps to capture storm surge propagation better toward the shallow regions. The authors conclude that enforcing spatially variable storm surge hydrographs on the boundaries of local grids improves storm surge prediction. In his MS thesis, Dietsche (2004) notes, however, that the use of hydrographs alone may not be sufficient, and that flux terms may be necessary for near-inlet boundaries to achieve reliable results.

Salisbury and Hagen (2007) perform a numerical parameter study to determine the influence tidal inlets have on open coast storm surge hydrographs. Four idealized inlet-bay configurations are presented with model output recorded on five points along each of eight semicircular arcs encompassing the inlet, and with recording stations located

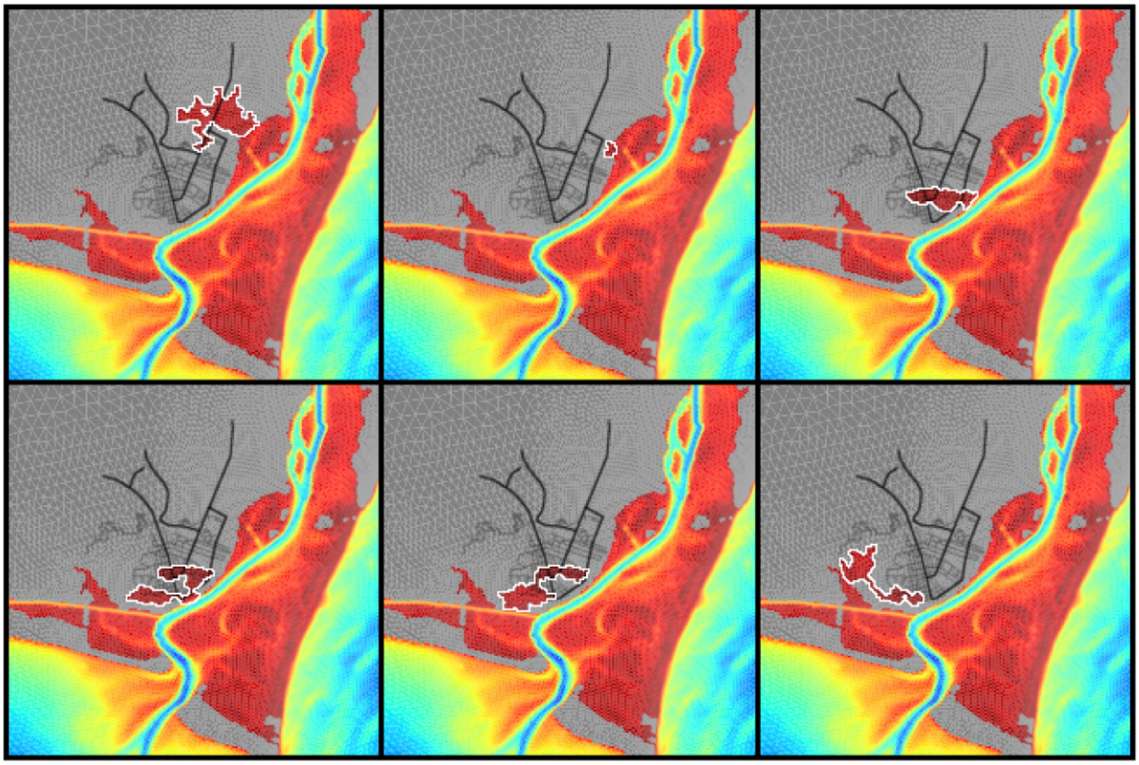


Fig. 1. Maximum water depths under various levee failure scenarios. (Flooding outlined in white, cooler colors are deeper).

from 1 to 15 km from the inlet. Additionally, Hurricane Ivan is simulated on the Pensacola Pass-Escambia Bay system in order to validate results from the idealized inlet-bay tests. In their study, the authors conclude that the effect of incorporating inlet-bay systems on open coast storm surge hydrographs is negligible. “Taking this [conclusion] into account,” they observe, “a large-scale model domain would not need to include the inlet-bay system in the computational domain to generate open coast boundary conditions for a local inlet-based model (a significant time-saving benefit for coastal modelers).” It should be noted that these results apply specifically to open ocean hydrographs since comparisons are made at points kilometers away from the inlet systems in the open ocean.

Funakoshi et al. (2008) describe one-way and two-way coupling procedures that combine the hydrodynamic and wave models ADCIRC and SWAN, along with their application to storm tide elevation calculations in the St. Johns River of Northeastern Florida during Hurricane Floyd (1999). The authors compare the median relative errors and RMS errors of twelve different model results and conclude that “these quantitative errors suggest that one must use [elevation] boundary conditions derived from a large-domain modeling approach in driving a local-domain model which places its open-ocean boundary near the inlet of the estuary.”

Bacopoulos et al. (2009) use ADCIRC to simulate a 122-day period, from June 1 to September 30, 2005, on three distinctly sized domains. The time period chosen contains a significant meteorological event, Hurricane Ophelia. All three domains represent the St. Johns River with the same high level of detail while containing differing extents of the Atlantic Ocean. The largest domain encompasses the Atlantic Ocean as far as the 60d W Meridian, including the entirety of the Gulf of Mexico and the Caribbean Sea. The intermediate domain extends into the Atlantic Ocean far enough to represent the continental shelf. The smallest domain has a minimal extension into the ocean, and aims to contain only the river itself and its immediate inlet. The three domains are run with different combinations of inclusion/exclusion of tidal forcings, land-boundary inflows, wind fields and pressure fields. The oceanic boundary conditions are described by tidal forcings only. As well, the two smaller domains are considered with a hydrograph boundary condition derived from the largest domain, which implicitly includes large-scale tidal and meteorological forcings on the boundary. The authors find that the size of the domain is important to the extent that remote meteorological forcings are important to the dynamics of the St. John's River. The largest domain significantly outperforms the two smaller domains for standard tidal boundary conditions when compared to four National Ocean Service gaging stations located along the St. John's River. However, when the hydrograph boundary condition is used for the two smaller domains, all three domains “virtually mirror” each other.

2.1. ADCIRC

The Advanced Circulation model (ADCIRC) is a parallel, unstructured-grid finite-element hydrodynamic code used by the U.S. Army Corps of Engineers (USACE), the Federal Emergency Management Agency (FEMA), and others to simulate storm surge and tides along the East coast of the United States and elsewhere (Tanaka et al., 2011; Westerink et al., 2008). ADCIRC supports three dimensional (3D) and two dimensional depth integrated (2DDI) analyses (Luettich and Westerink, 2004). The 2DDI formulation, used in this study, is derived from the vertically integrated shallow water equations using the generalized wave continuity equation (GWCE) formulation (Luettich and Westerink, 2004).

Three primary routines drive the physics of the model and are executed in succession at each timestep: the GWCE routine, the wet/dry routine, and the momentum equation routine (Tanaka et al., 2011). The GWCE routine, which finds the free surface elevation at each node in the domain for the current timestep, has the ability to use either a spatially implicit or explicit method, employing an iterative Jacobi Conjugate Gradient method or a lumped mass matrix method, respectively. In this study, the implicit method is used. The wetting and drying routine determines the wet or dry status of each node in the domain via a series of checks based on elevation, velocity, wet/dry status in the previous timestep, and wet/dry status of neighboring nodes (Luettich and Westerink, 1999). The momentum routine finds the x and y velocity at each node in the domain for the current timestep by solving the shallow water equations explicitly using a lumped mass matrix. An extensive description of the mathematical formulation of ADCIRC is provided by Luettich and Westerink (2004).

When run in a parallel processing environment, ADCIRC decomposes its physical domain into subdomains that are approximately equal in node and element quantity, with each subdomain being placed on a dedicated processor. At each timestep, information pertinent to the current state of the physics being modeled is communicated between neighboring subdomains using message passing. By using “ghost nodes,” each subdomain receives information at every node on its boundary from an adjacent subdomain (or from a boundary condition defined for the domain in cases where a subdomain shares a boundary with the full domain), and only solves for nodes that are inside the subdomain. The physical information communicated includes water surface elevation, depth averaged water velocity, flux per unit width, and wet/dry status. The implicit version of the GWCE routine, when in use, also employs message passing for a global summation to find the residual norm of the solution.

3. Motivation and approach

As a product of modeling that necessarily requires judgment, storm surge simulations may need to be modified and repeated for a variety of reasons. In cases where a large-scale model has been validated, but where a local geographic change is under consideration, it is possible to eliminate the repetitive computations of an additional full scale simulation in areas unaffected by the change. The most straightforward application of this idea is to parametric studies of storm surge effects. By varying one or more local properties of a physical domain over a meaningful range of values, the effects of those changes can be simulated in a geographic region of interest with much less effort.

Another area of application for subdomain modeling is in calibrating the local parameters of a storm surge model to match field data. If high-water marks or buoy data are available in a region of interest, for instance, and the local model parameters or grid properties are deemed inadequate in some way, they can be repetitively varied until the model matches the data.

Finally, iterative design scenarios are enabled by subdomain modeling, allowing the differential coefficients of responses —such as water surface elevation and velocity, with respect to infrastructure or other design parameters— to be obtained more efficiently. Such applications might include the design of protective structures of varying sizes, geometries, and layouts, and would typically also account for failure states of the designs both individually and in their various combinations. In this scenario, an engineer might explore design space in an ad hoc manner, or do so using a formal search procedure within an optimization framework based on, for instance, an appropriate metaheuristic.

In such cases, modeling changes to be considered might be topographic, such as adding protective structures and raising roadway grades or shifting alignments; roughness related, such as restoring wetlands and marshes or making land-use changes to simulate new development; or other parameter adjustments that in some way reflect a physical characteristic of interest. Effects to be assessed would draw on the type of output ordinarily produced by ADCIRC, such as surge levels and locations as a function of time, and depth averaged water velocities, along with post-processing tools to estimate flooding on street and highway networks, costs to communities using depth-damage relationships like those produced by the U.S. Army Corps of Engineers

(2012), and susceptibility to scour, dune washout, and roadway damage in areas of interest.

3.1. Comparison with nesting

To allow extraction of a geographic region of interest, interface conditions between a full domain and subdomain are needed, and these may be formulated in a variety of ways. Indeed, there is a substantial body of work in both ocean and atmospheric modeling with one-way nesting (Spall and Robinson, 1989), two-way nesting of interacting models (Debreu and Blayo, 1993), and full coupling based on domain decomposition techniques (Cailleau et al., 2008). Driving these and similar efforts is the recognition that smaller spatial and temporal scale processes may be inadequately resolved in coarse, large-scale models. In such applications, it is well known that discrepancies between the more refined interior solution of a nested model and its boundary conditions may cause reflections, giving rise to relaxation and other techniques such as sponge layers that artificially adjust parameters in the nested model's interior proximal to its boundary.

The primary concerns for specifying appropriate boundary conditions, then, are well-posedness (Kreiss, 1970), stability (Elvius and Sundstrom, 1973), and artificial wave reflections occurring at the boundary (Engquist and Majda, 1977). Effective and accurate methods such as adaptive mesh refinement, damping, and radiation methods have been extensively studied in an effort to avoid the errors associated with the specified boundary conditions (Marchesiello et al., 2001). In one of the early studies, Elvius and Sundstrom (1973) present two sets of stable boundary conditions for the integration of the limited-area shallow water equations, and point out that, although easy to implement, prescribing all quantities along the boundary may lead to a spatially short-wave, quasi-geostropic error component due to the discrepancy between the boundary conditions obtained from the coarse grid and the solutions of the fine limited-area grid.

By way of contrast, subdomain modeling, as presented here, is not an attempt to capture these fine-scale hydrodynamic processes by nesting high resolution models, but is rather an approach to accommodate local modifications of a grid that constitute “what if” type scenarios, without the need to perform an entire large-scale simulation again and again. As a result, formulation of the interface conditions is simplified, and a one-way hand off of boundary information is sufficient to obtain exact results, as we later show.

Like the domain decomposition solver used in Parallel ADCIRC (Tanaka et al., 2011), our approach makes use of water surface elevation, wet/dry status, and depth averaged water velocity along the interface. A new type of boundary condition, simultaneously incorporating all three quantities, is then realized by:

- working with the existing non-periodic elevation boundary condition formulation in ADCIRC, which specifies nodal elevations in the implicit GWCE formulation,
- incorporating the ability to force wet/dry status on boundary nodes while the wetting and drying routine executes, and
- taking advantage of the explicit nature of the momentum equation solver to assign boundary velocities outright.

A flowchart of the modified timestepping procedure in ADCIRC is shown in Fig. 2. Original steps are shaded gray, and new or modified steps are left unshaded.

3.2. Well posedness

Consistent with derivations and proofs in the following section, numerical experiments confirm that boundary conditions combining water surface elevation and depth averaged water velocity are viable in the sense of Drolet and Gray (1988) and Spall and Robinson (1989), i.e., they are effective at the space and time scales of interest. With respect to our approach, then, we define the following intuitive requirements for effectiveness:

Locality of topographic and other changes: So long as the subdomain is large enough to fully contain the altered hydrodynamics, modifications may be made within it without the need to calculate new boundary values.

Sufficient forcing frequency of boundary conditions: While the state of a boundary node can be recorded and forced at each timestep, doing so may be space prohibitive, and also unnecessary; storing and using nodal state data every n timesteps instead allows the tradeoff between space requirements and accuracy to be explored and convergence demonstrated.

Though only a subset is presented here, simulations of subdomains of regular and irregular geometries ranging in size from about 20 to 2000 mile^2 in areas along the North Carolina coast, including Cape Fear, Wrightsville Beach, North Topsail Beach, Cape Lookout, and Hatteras Island, confirm that the approach is well posed and that subdomain solutions converge —as the frequency of forcing boundary conditions is increased— to those that would be obtained by a full run.

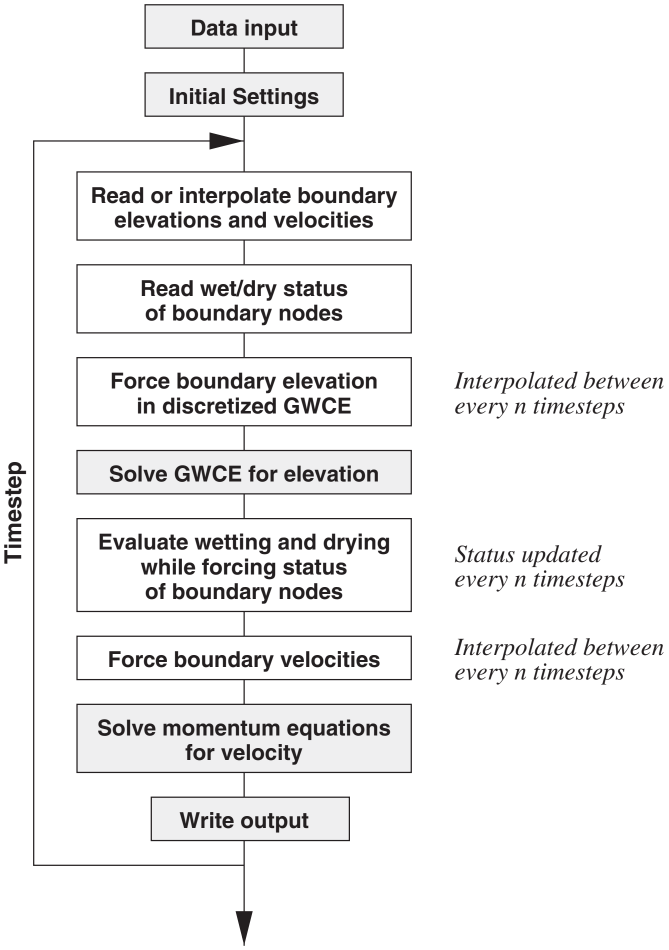

Data input
Initial Settings
Read or interpolate boundary elevations and velocities
Read wet/dry status of boundary nodes
Force boundary elevation in discretized GWCE
Interpolated between every n timesteps
Solve GWCE for elevation
Evaluate wetting and drying while forcing status of boundary nodes
Status updated every n timesteps
Force boundary velocities
Interpolated between every n timesteps
Solve momentum equations for velocity
Write output
Timestep

Fig. 2. Flowchart of the modified timestep loop for subdomain modeling. (Original steps in gray, new ones left unshaded).

4. A new boundary condition type

The realization of subdomain modeling in ADCIRC, or in any other spatially discrete numerical approach, requires a boundary condition type incorporating the quantities described; we consider them in the order in which they appear in the timestep loop: water surface elevation, wet/dry status, and velocity. In each case, we describe computational procedures that enforce them, and show how the temporal and spatial data dependencies at the interface may be resolved so that subdomains can be extracted while preserving interior computations that correspond exactly to those seen in a full domain run. The procedures are derived by considering unmodified subdomains and their relationship to the original, full domain, with the recognition that, for modified subdomains, a verification step must be performed to ensure that local changes have only local effects within the subdomain.

4.1. Prescribing water surface elevation

In ADCIRC, the GWCE and momentum equations are temporally discretized in a three-level and two-level scheme, respectively, leading to a decoupled solution of the two equations, which allows for a sequential solution procedure (Westerink et al., 1992). The equations are then spatially discretized using a standard Galerkin finite element scheme that operates on irregular linear triangles (Tanaka et al., 2011), yielding a linear system of n equations, where n is the number of nodes:

$$[C]\{\zeta\} = \{L\}$$

The coefficient matrix C can be interpreted as a discrete characterization of the geometry and the connectivity of the computational grid. The load vector L determines the forcing term at each node due to wind stress, lateral stress, tidal constituents, bottom friction, etc., and is assembled first by looping through each element, and then by summing up contributions from elements to obtain the terms for each node. The vector ζ contains the water surface elevation changes for each node at a timestep, and is computed using a Jacobi Conjugate Gradient iterative solver (Luettich and Westerink, 2004).

The matrix C is sparse, symmetric, and positive-definite, and depends on the geometry of the elements, wet/dry states, the connectivity of the nodes, the weighting factor of the previous timestep, and the model parameter τ_0. Each element in the grid contributes to the global system, which is assembled at the first timestep and is recomputed if any wetting or drying occurred in the previous timestep. The contribution of an individual element is defined by a local 3 × 3 coefficient matrix together with a vector m^k that maps local to global indices (i.e., $i = m^k_{i'}$). Terms are defined as follows for element k with local nodes i' and j':

$$c^k_{i',j'} = \frac{A_k}{12\Delta t}\left(\frac{1}{\Delta t}+\frac{\overline{\tau}_{0k}}{2}\right)\phi_{i',j'} + \frac{g\overline{h}_k\alpha_1}{4A_k}\left(b_{i'}b_{j'}+a_{i'}a_{j'}\right) \qquad (1)$$

where

A_k	area of elemet of k
$\overline{\tau}_{0k}$	average τ_0 over element k
$\phi_{i',j'}$	$\begin{cases} 2 & \text{if } i' = j' \\ 1 & \text{otherwise} \end{cases}$
α_1	weighting factor of the previous timestep
$\overline{h}_k$	average bathymetric water depth over element k
a_1	$x_3 - x_2,\ a_2 = x_1 - x_3,\ a_3 = x_2 - x_1$
b_1	$y_2 - y_3,\ b_2 = y_3 - y_1,\ b_3 = y_1 - y_2$
$x_{i'}, y_{i'}$	horizontal coordinates of node i'

A diagonal term on the jth row of the matrix corresponds to the coefficient involving node j itself, and a non-zero off-diagonal term on the jth row and ith column corresponds to the coefficient involving node j and an adjacent node i that are incident on the same element, provided that nodes i and j are both wet. Similarly, the jth row of the ζ vector corresponds to the change in the water surface elevation of node j, and the jth row of the L vector corresponds to the right-hand side of the GWCE equation for node j.

A dry node is excluded from the system of equations by zeroing out the load vector term and the off-diagonal terms in the coefficient matrix, and setting the diagonal term to the root mean square of all the diagonal terms.

4.1.1. Water surface elevation for subdomain boundaries

For subdomain modeling, we wish to prescribe the water surface elevation of boundary nodes using data obtained from an initial full domain run, so we begin by grouping nodes in the full domain according to type: *external* (e), *boundary* (b), and *internal* (i). Doing so leads to the following partitioned system of equations:

$$\begin{bmatrix} C_{ee} & C_{eb} & 0 \\ C_{be} & C_{bb} & C_{bi} \\ 0 & C_{ib} & C_{ii} \end{bmatrix} \begin{bmatrix} \zeta_e \\ \zeta_b \\ \zeta_i \end{bmatrix} = \begin{bmatrix} L_e \\ L_b \\ L_i \end{bmatrix}$$

Since an external node is not connected to any internal node, and vice versa, C_{ei} and C_{ie} are zero block matrices. Therefore, the final equation is simply:

$$C_{ib}\zeta_b + C_{ii}\zeta_i = L_i \qquad (2)$$

This relationship can be used as follows: after performing a full run, the quantities of the product $C_{ib}\zeta_b$ are known and saved, allowing us to use either the original or modified values of C_{ii} and L_i to find the elevation changes internal to the subdomain, ζ_i, the unknowns of interest. It should be clear that when C_{ii} and L_i are unmodified, the values obtained for ζ_i in a subdomain run are equivalent to those originally obtained. When C_{ii} and L_i associated with a subdomain have been modified, we must rely on a post-processing verification step, as noted. Alternatively, one might imagine employing backward error analysis or other techniques in an attempt to predict, at each timestep, whether a modified subdomain is in that sense valid; we leave this for future work.

4.2. Prescribing wet/dry status

The order of the GWCE system of equations is determined by the number of "wet" nodes in the domain. In ADCIRC, both nodes and elements have wet/dry states, as well as other temporary states, that are set by an algorithmic process at each timestep. Additionally, elements themselves may be either active or inactive. Because neighboring node and element states are interrelated, care must be taken within the algorithm if subdomains are to behave in a manner consistent with the original full domain run.

We begin with the following definitions and primary state variables used by the wetting and drying algorithm.

Final wet/dry status: the wet/dry status of a node i at a given timestep, which is read and used by the GWCE on completion of the algorithm. This final status is maintained by the Boolean variable W_i, which is true when the node is "finally wet."

Temporary wet/dry status: the wet/dry status of a node i, which is updated and tested during the evaluation of the algorithm. This temporary status is maintained by another Boolean variable, W^t_i, which is true when the node is "temporarily wet."

Intermediate wet/dry status: the temporary wet/dry status of a node i immediately on completion of the second step of the algorithm, described below.

Active element: a wet element with three wet nodes.

Inactive element: a dry element, or a wet element with at least one dry node.

Landlocked node: a node connected only to inactive elements.

The wetting and drying algorithm in ADCIRC works by checking four main criteria in sequence at each timestep, as shown in Algorithm 1. First, nodes with less than the minimum water column height H_0 are made dry. Second, steady state velocities are used to determine whether the wet front should propagate across an element. Third, elements with two "barely wet" nodes are made dry, causing water to build up before it is allowed to flow. Fourth, wet nodes that are landlocked are made dry. Finally, after all of the criteria are checked and temporary states assigned, the final wet/dry states of the nodes are assigned. Data dependencies introduced by these steps are characterized in Fig. 3.

Algorithm 1. Wetting and drying algorithm

```
0. Initialization (start with all elements being wet)
for e in elements do
    make e wet
1. Nodal Drying (consider nodes with water column height $H$ less than $H_0$ to be dry)
for i in nodes do
    if $W_i$ and $H_i < H_0$ then
        $W_i \leftarrow$ false, $W_i^t \leftarrow$ false
2. Nodal Wetting (propagate wetness across element if velocity is sufficient)
for e in elements do
    if e has exactly 2 wet nodes and $V_{ss}(e) > V_{min}$ then
        Let j be the remaining dry node
        $W_t(j) \leftarrow$ true
for i in nodes do
    if not $W_i$ and i has adjacent receiving barrier nodes then
        $W_i^t \leftarrow$ true
3. Elemental Drying (allow water to build up in a "barely wet" element)
for e in elements do
    find nodes i and j of e that have the largest values of water surface elevation, $\eta_i$ and $\eta_j$
    if $H_i$, $H_j < 1.2H_0$ then
        make element e dry
4. Nodal Drying (make landlocked nodes dry)
for i in nodes do
    if $W_i^t$ and i is connected only to inactive elements then
        $W_i^t \leftarrow false$
5. Final Update Step
for i in nodes do
    $W_i \leftarrow W_i^t$
```

4.2.1. Wet/dry status for subdomain boundaries

We again wish to make use of data from an initial full domain run, in this case to prescribe the wet/dry status of boundary nodes. To do so, the final wet/dry states from the full run are recorded and then used in subdomain runs. If the water surface elevations of the subdomain nodes match those of the full domain, which we ensure in the GWCE and momentum solvers, it can be shown that recording only final states and enforcing them on boundary nodes at the completion of steps 2 and 4 of the algorithm is sufficient for correctness, as described below. The proof is presented in pieces that collectively exhaust the set of possible cases where a discrepancy could occur in the wet/dry status of both internal and boundary subdomain nodes.

Lemma 4.1. *The intermediate wet/dry state of a node is wet if its final wet/dry state is wet.*

Proof. A finally wet node cannot be temporarily dry after the second step. Therefore, if the final wet/dry state of a node is wet, its temporary wet/dry state at the second step must have been wet. □

Lemma 4.2. *If it is determined in the full domain run that the final wet/dry state of a subdomain boundary node is wet, setting its intermediate state in the subdomain run using the final wet/dry state obtained from the full domain run (wet) does not cause a discrepancy in the evaluation of wetting and drying in the subdomain grid.*

Proof. Since Lemma 4.1 ensures that the intermediate wet/dry state of a node is wet if its final wet/dry state is wet, there is no case in which a wet node is temporarily dry after the second step. Therefore, all intermediate states of all variables of a wet boundary node in a subdomain run match the intermediate states of the corresponding node in a full domain run. □

Lemma 4.3. *If the intermediate wet/dry state of a node is wet and its final wet/dry state is dry, then the node is landlocked.*

Proof. After the intermediate state is determined, the only criterion checked that can cause a temporarily wet node to be dry, is the landlocking criterion. □

Lemma 4.4. *If a subdomain boundary node is dry and is not landlocked, setting its intermediate wet/dry state using the final wet/dry state (dry) does not cause a discrepancy.*

Proof. From Lemma 4.3, the only situation in which a dry node is temporarily wet after the second step is when the node is landlocked, so if the node is dry and is not landlocked, it is dry also after the second step of the algorithm. Therefore, using the final wet/dry state (dry) to enforce the intermediate wet/dry state of the subdomain boundary node causes the intermediate states of subdomain run to match the corresponding intermediate states of the full domain run. □

Lemma 4.5. *The only way that changing the intermediate wet/dry state of a node can have an effect on the final wet/dry state of an adjacent node is if doing so changes the active/inactive state of the elements incident on them.*

Proof. Within the wetting and drying algorithm, the only criterion that considers the intermediate wet/dry state of adjacent nodes after they have been set is step 4, and that criterion does so only by testing the active/inactive state of an element. □

Lemma 4.6. *If a temporarily wet node is landlocked (and therefore finally dry), its intermediate wet/dry state has no effect on its neighbors' wet/dry states.*

Proof. The only wet/dry criterion in which the intermediate wet/dry states of *neighbors* are tested is the landlocking criterion. In this criterion, the active/inactive states of surrounding elements of a node are checked. If all are inactive, the temporarily wet node is regarded as landlocked and made finally dry. If it is already determined in the full domain run that the surrounding elements of a subdomain boundary node are inactive, setting the intermediate wet/dry state of this boundary node to temporarily dry (instead of temporarily wet) does not change the active/inactive state of the elements incident on it, and therefore, from Lemma 4.5, the final wet/dry state of the adjacent nodes will not be affected. □

Theorem 4.7. *Provided the water surface elevations of the subdomain nodes are the same as the corresponding full domain nodes, enforcing both intermediate and final wet/dry states of subdomain boundary nodes using the corresponding final wet/dry states obtained from the full domain run results in the same final wet/dry states of the subdomain grid nodes as the final wet/dry states of the corresponding full domain nodes.*

Proof. The first three criteria of the wetting and drying algorithm depend only on the water surface elevations of the nodes themselves and the elevations of their neighbors. Since it is assumed that the subdomain elevations are identical to the full domain elevations, the progress of the wetting and drying algorithm for the internal subdomain nodes is the same as for the corresponding nodes in the full domain run until the final criterion. At this stage, absent any intervention, boundary nodes would potentially have erroneous wet/dry states, since some of their neighbors are excluded from the subdomain. To correct the final wet/dry states of the boundary nodes, the final wet/dry states obtained from the full domain run are used to enforce them. Additionally, to be able to faithfully determine the wet/dry states of internal nodes adjacent to the boundary at the final step, the intermediate wet/dry states of the boundary

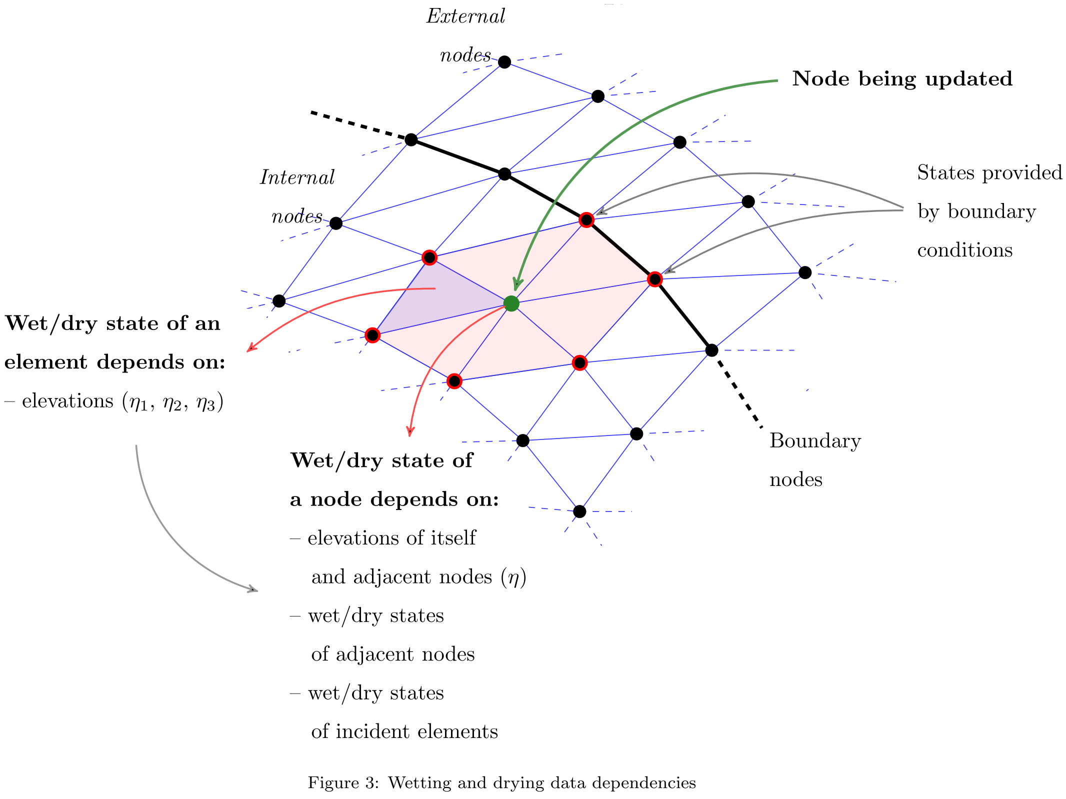


Fig. 3. Wetting and drying data dependencies.

nodes are forced using the final wet/dry states obtained from the full domain run. Taken together, Lemmas 4.2, 4.4, and 4.6 exhaust the set of possible cases where a discrepancy could occur in the wet/dry states of subdomain nodes, allowing us to conclude that forcing both intermediate and final wet/dry states of subdomain boundary nodes using final states obtained from a full run is sufficient to guarantee correctness. □

4.3. Prescribing velocity

After the GWCE is solved and the wetting and drying algorithm completes, ADCIRC solves the momentum equations. It is here that we use the velocities of boundary nodes from an initial full domain run as the final step in ensuring that a subdomain behaves consistently with its full domain counterpart.

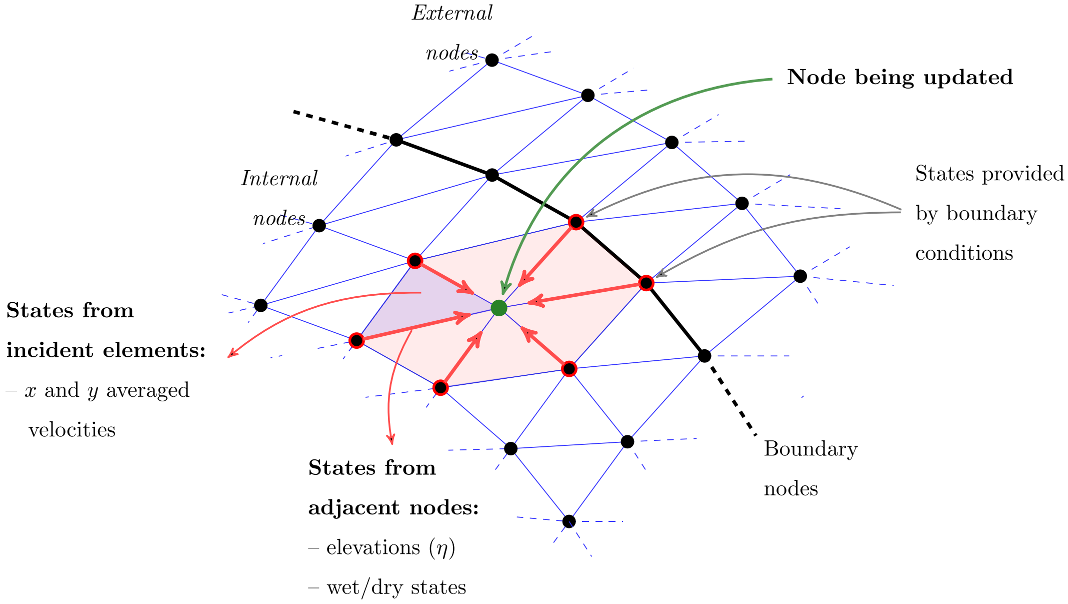


Fig. 4. Momentum equation data dependencies.

The x and y velocities of node i in the current timestep, s, are denoted by the vector quantity $\mathbf{v}_i^s$. Its value in the subsequent timestep $s + 1$ is a function of the wet/dry state of incident elements as well as advective, barotropic pressure gradient, and lateral viscous terms. The resulting temporal and spatial dependencies, illustrated in Fig. 4, can then be characterized as follows:

$$\mathbf{v}_i^{s+1} = f\left(\mathbf{v}_i^s, \mathbf{v}_j^s, \mathbf{v}_k^s, \ldots\right) \tag{3}$$

where j, k, … are nodes adjacent to i, the node being updated. In addition, other quantities required to compute $\mathbf{v}_i^{s+1}$ from neighboring nodes include the water surface elevations at the previous and current timesteps and wet/dry states.

Unlike with wetting and drying, a simplifying aspect of solving the momentum equations explicitly is that a velocity $\mathbf{v}_i^{s+1}$ is computed once, stored, and never updated. Therefore we need only ensure that the boundary nodes have appropriate velocities when new velocities in the interior of a subdomain are computed.

To implement the approach, ADCIRC is modified so that, during a subdomain run, the velocities associated with boundary nodes are not computed but are instead set to values obtained during the full run. Then, velocities in the interior of the subdomain are computed using the same boundary values previously obtained and stored.

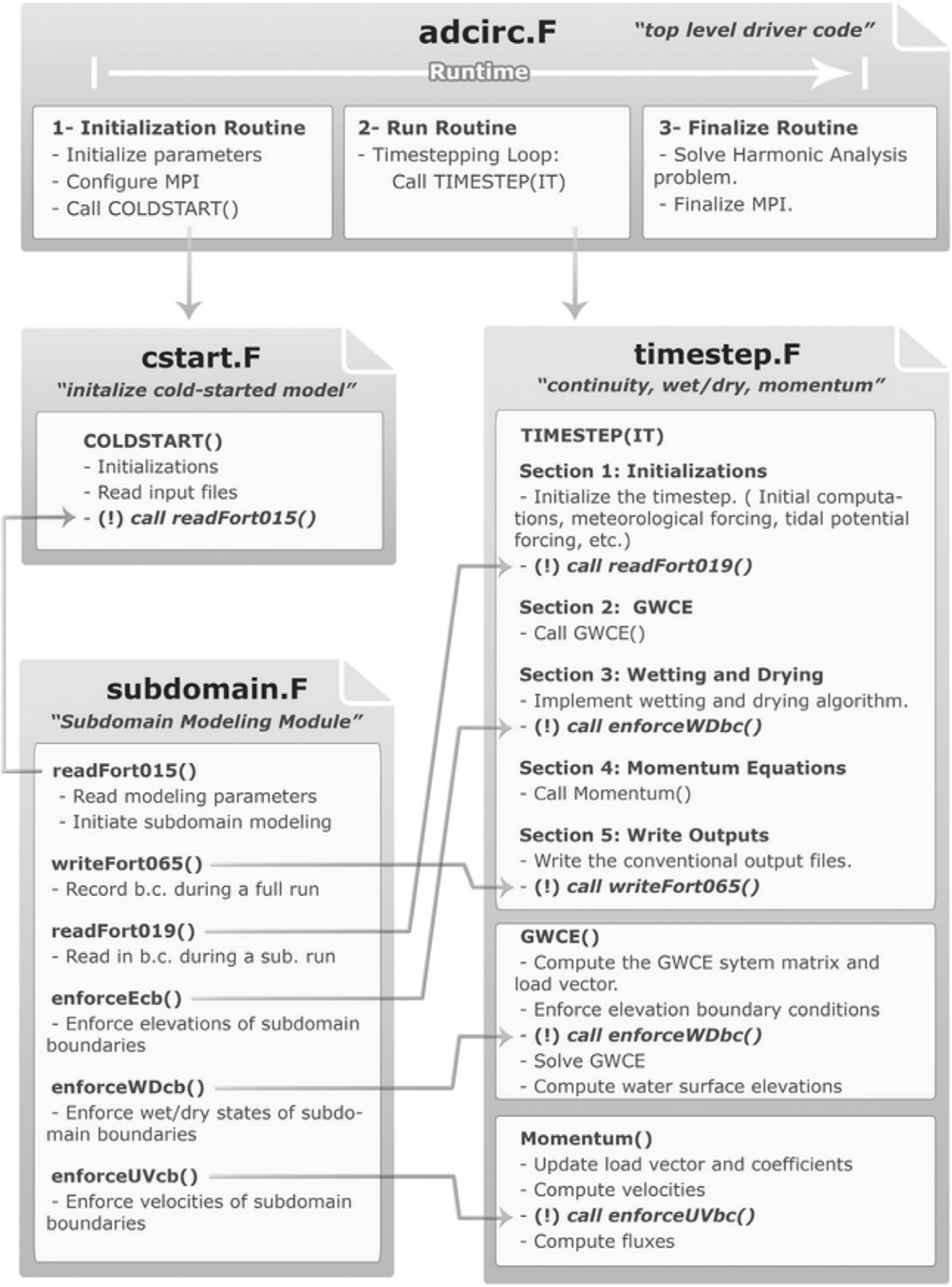

adcirc.F
"top level driver code"
Runtime
1- Initialization Routine
- Initialize parameters
- Configure MPI
- Call COLDSTART()
2- Run Routine
- Timestepping Loop:
Call TIMESTEP(IT)
3- Finalize Routine
- Solve Harmonic Analysis problem.
- Finalize MPI.
cstart.F
"initalize cold-started model"
COLDSTART()
- Initializations
- Read input files
- (!) call readFort015()
subdomain.F
"Subdomain Modeling Module"
readFort015()
- Read modeling parameters
- Initiate subdomain modeling
writeFort065()
- Record b.c. during a full run
readFort019()
- Read in b.c. during a sub. run
enforceEcb()
- Enforce elevations of subdomain boundaries
enforceWDcb()
- Enforce wet/dry states of subdomain boundaries
enforceUVcb()
- Enforce velocities of subdomain boundaries
timestep.F
"continuity, wet/dry, momentum"
TIMESTEP(IT)
Section 1: Initializations
- Initialize the timestep. (Initial computations, meteorological forcing, tidal potential forcing, etc.)
- (!) call readFort019()
Section 2: GWCE
- Call GWCE()
Section 3: Wetting and Drying
- Implement wetting and drying algorithm.
- (!) call enforceWDbc()
Section 4: Momentum Equations
- Call Momentum()
Section 5: Write Outputs
- Write the conventional output files.
- (!) call writeFort065()
GWCE()
- Compute the GWCE sytem matrix and load vector.
- Enforce elevation boundary conditions
- (!) call enforceWDbc()
- Solve GWCE
- Compute water surface elevations
Momentum()
- Update load vector and coefficients
- Compute velocities
- (!) call enforceUVbc()
- Compute fluxes

Fig. 5. Outline of the subdomain modeling implementation in ADCIRC.

5. Implementation and work flow

Subdomain modeling is implemented within ADCIRC's timestep loop, with extensions appearing in two new source files: *prep/subprep.F* and *src/subdomain.F*. The subroutines declared in these additional source files are called within the original source files *adcprep.F*, *timestep.F*, and *cstart.F*. The locations of the calls made to these subroutines during a subdomain modeling run are shown in Fig. 5. An input file is created from a setup run that includes water surface elevation, *x* and *y* velocities, and wet/dry status of each node at a specified reporting interval. While ADCIRC ordinarily reports water surface elevation and velocity, further changes allow more user control over the quantities output for any nodes of interest, i.e., those that may later comprise one or more subdomains, as well as their wet/dry states, which would otherwise not be reported. For nodes that are dry, topobathymetric heights above mean sea level are used for their water surface elevations. Also, since it is not possible to interpolate between binary states, a node's wet/dry status remains unchanged until the time at which a new status has been recorded.

5.1. Work flow

The construction of a subdomain model in ADCIRC consists of four main steps. First, certain subdomain input files and control files are generated. Second, a full domain ADCIRC run is performed. Third, a subdomain boundary conditions file is generated using full domain output files. Finally, a subdomain ADCIRC run is performed.

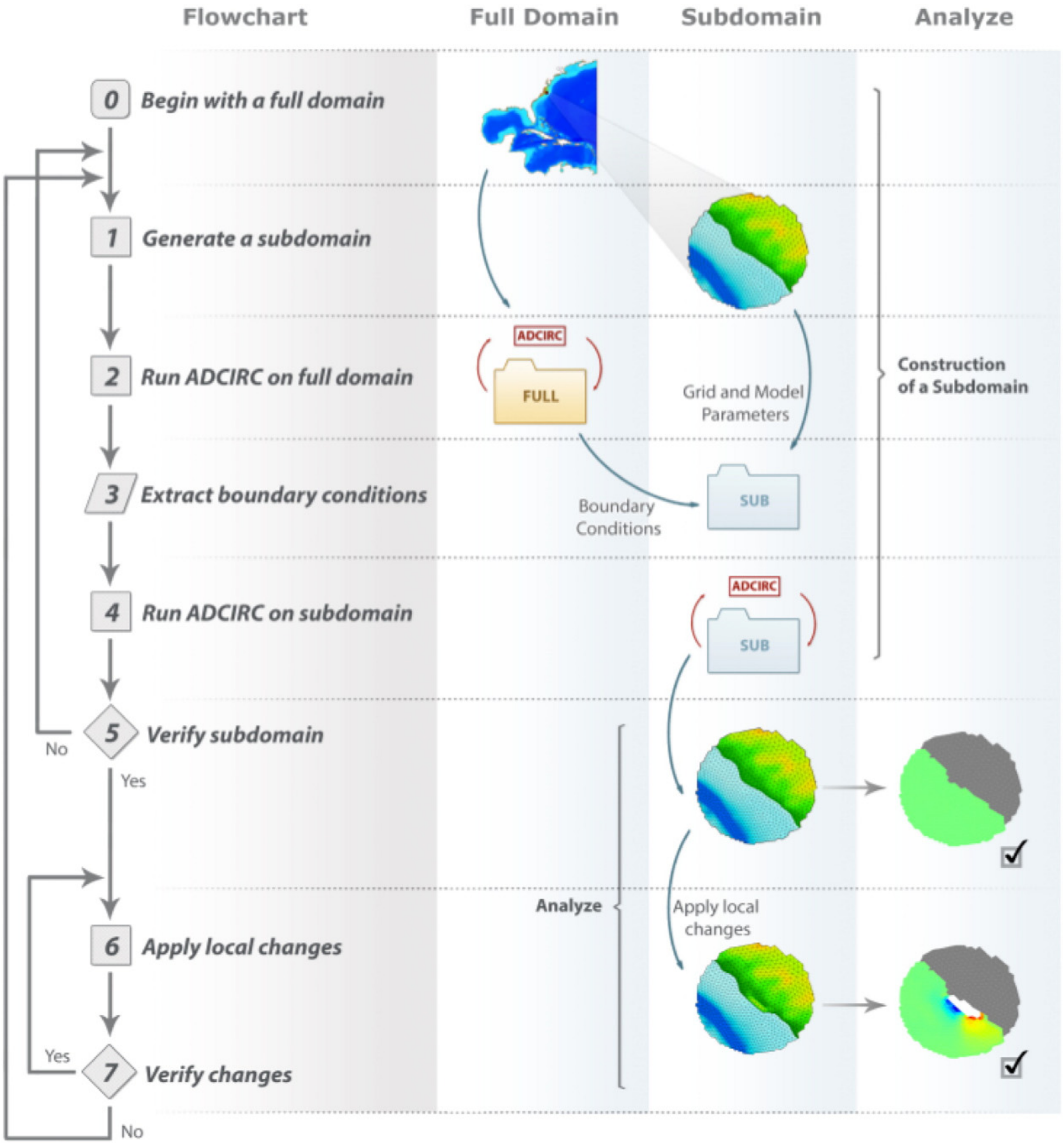


Fig. 6. Work flow for subdomain modeling.

Following the construction of a subdomain model, various case studies can be carried out in three main steps. First, the subdomain grid is verified, i.e., it is ensured that results of the subdomain and full domain runs match. Second, various local changes are applied to the verified subdomain grid. Finally, the subdomain grid with local changes is verified, i.e., it is ensured that the effects of the changes do not propagate to the subdomain boundaries.

Fig. 6 summarizes the main steps of the subdomain modeling approach, which is facilitated by a graphical user interface, called SMT, that allows geographic regions of interest to be specified and extracted (Dyer, 2013). An introduction to SMT is presented in Section 8. A user guide (Altuntas et al., 2013) offers additional detail, along with the required input and generated output files.

An alternative work flow is suggested by the idea that subdomain models might be extracted from previously archived simulations, since output from existing studies might be available and local changes made without rerunning an entire simulation. Although wet/dry states would not be available for subdomain boundary nodes in such a case, one could allow them to be recomputed as needed by ADCIRC since the wet/dry algorithm is executed after the GWCE routine. Altuntas (2012), however, describes cases in which the wet/dry states of nodes and elements just outside of a subdomain are needed by ADCIRC's wet/dry algorithm to faithfully reproduce wet/dry states within a subdomain. Alternatively, instead of trying to recalculate wet/dry states, one might instead attempt to recover and make use of them from the reported water surface elevations of archived simulations. Though not further considered here, such an approach could be viable, we believe, in certain contexts.

5.2. Tidal forcing

Tidal models in ADCIRC are constructed by specifying tidal potential forcing that sets the effects of the tidal constituents inside the conputational domain, and tidal boundary forcing that sets the effects of the tidal constituents along the boundary. Tidal potential forcing must be specified for both full domain and subdomain runs. However, tidal boundary forcing is required only for full domain runs, since the elevation boundary conditions for the subdomain grids, which are obtained from the full domain runs, already include the effect of the tidal constituents.

6. Test cases

The following sections demonstrate results on both a simple test case and large-scale simulations of historical storms in the western North Atlantic Ocean. Results were obtained using modifications for subdomain modeling made to ADCIRC v.50, which was built using PGI Fortran 11, GCC 4.5 and MPICH2 1.3.1. All simulations were executed on a single server rack containing 4 AMD Opteron 6168 (12 core) processors for a total of 48 cores and 64 gigabytes of RAM.

6.1. Quarter annular harbor with tidal forcing

The quarter annular test case, first developed by Lynch and Gray (1978), is a simple example used to evaluate the performance of numerical approaches for solving hydrodynamic problems (Westerink et al., 1994). The problem consists of a radially symmetric mesh with an open ocean boundary along an outer arc of radius 152,400 m, and land boundaries along the other edges, including an inner arc of radius 60,960 m. The bathymetric depth varies quadratically from 3.048 m along the inner arc to 19.05 m along the outer. A tidal force is applied at the open ocean boundary; there are no wind or pressure forces. The mesh consists of 63 nodes, and a 20-node subdomain is defined within it, as shown in Fig. 7.

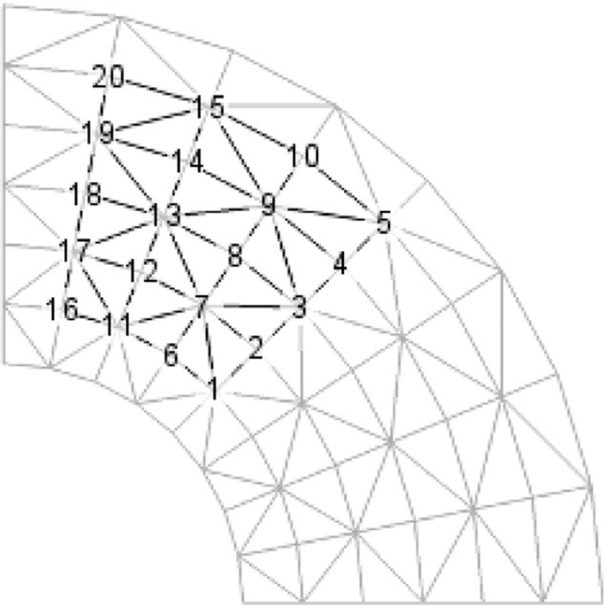


Fig. 7. The quarter annular problem (gray) and a subdomain (black).

After performing a simulation on the full domain as originally defined (with absolute convergence criteria for the GWCE solver set to 10^{-5}), boundary conditions obtained from the water surface elevations and depth averaged water velocities are forced along the boundary of the subdomain in a subsequent simulation at every timestep. Quantities within the interior of the subdomain match those originally obtained with good accuracy: The absolute errors in maximum water surface elevations and maximum velocities are on the order of 10^{-7}. Accordingly, reducing the absolute convergence criteria for the GWCE solver from 10^{-5} to 10^{-7} further decreases the order of absolute errors in maximum water surface elevations and maximum velocities from 10^{-7} to 10^{-9}.

6.2. Western North Atlantic storms

Turning to large-scale domains, the simulations that follow make use of NC FEMA Mesh Version 9.92 and control files provided by the ADCIRC development group, with a domain that encompasses the

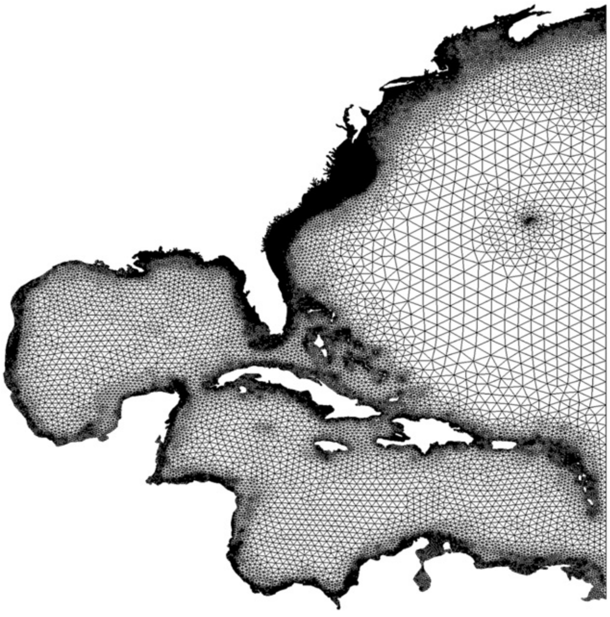


Fig. 8. Finite element mesh of the western North Atlantic Ocean.

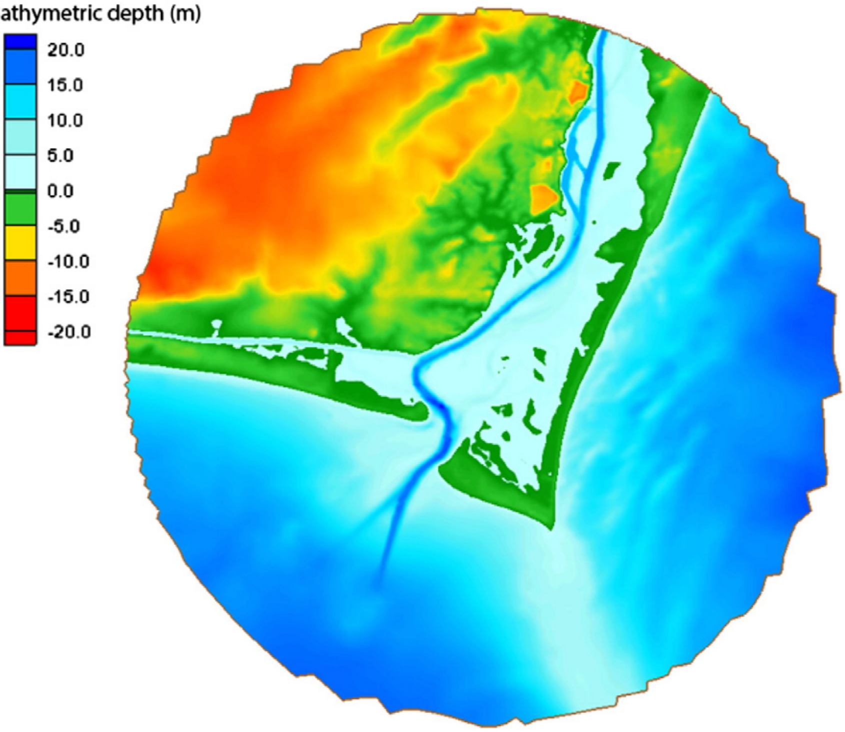


Fig. 9. Cape Fear terrain: a subdomain with a radius of 10.8 mi.

western North Atlantic Ocean, the Caribbean Sea, and the Gulf of Mexico. The mesh, shown in Fig. 8, consists of 620,089 nodes and 1,224,714 elements, and has a steady open ocean boundary condition along its eastern edge; the water surface is specified as constant at mean sea level. External land boundaries along the North American coastline have no normal flow and free tangential slip. Islands inside the domain are represented by internal land boundaries. Nodes in the mesh have attributes that include surface directional effective roughness length, a value for Manning's n at the sea floor, a surface canopy coefficient, and a primitive weighting in the continuity equation.

Wind and pressure fields used to model hurricanes follow the Oceanweather (OWI) wind format, which describes pressure and velocity on a rectangular grid at regular time intervals. Data values are interpolated in time to synchronize with the model time and interpolated in space to describe the meteorological state at the underlying mesh's computational nodes. Garratt's formulation is used to compute wind stress from wind velocity (Garratt, 1977; Luettich and Westerink, 2010).

When performing the 2DDI runs below, meteorological data values are sampled every 1800 s. Time weighting factors for timesteps $s + 1$, s, and $s - 1$ are 0.35, 0.30, and 0.35, respectively. The wet/dry algorithm operates using a minimum water depth of 0.1 m and a minimum velocity for wetting of 0.01 m/s. A quadratic bottom friction law is specified, finite amplitude terms with wetting and drying of elements are enabled, and both the spatial and time derivative portions of the advective terms are included. Tidal constituents are not included.

6.2.1. Hurricane Fran (1996) and Cape Fear

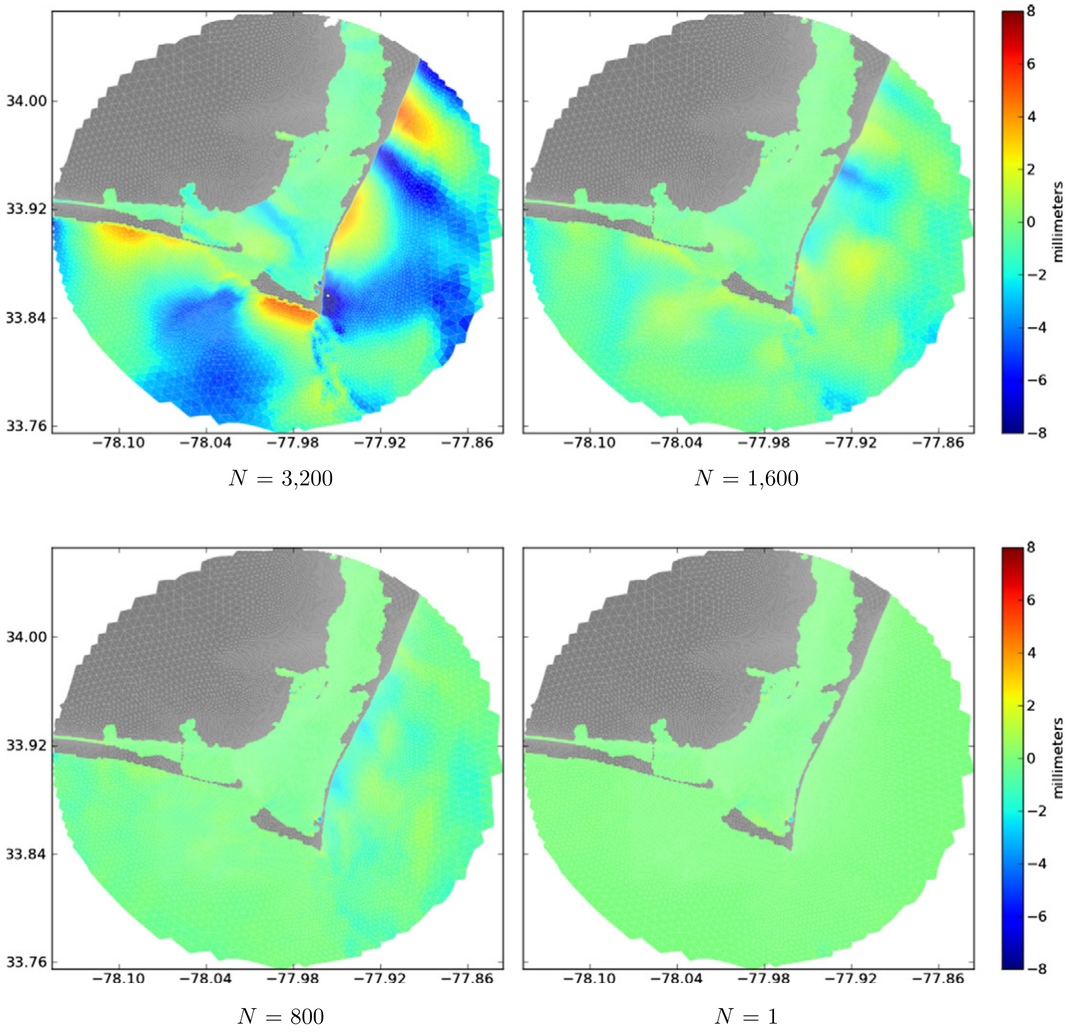
From the western North Atlantic mesh we extract a circular subdomain in the Cape Fear area of North Carolina covering 366 sq mi and consisting of 28,643 nodes, 56,983 elements, and 302 boundary nodes, as shown in Fig. 9. Then, a 3.9-day simulation of Hurricane Fran is performed on the original mesh using a timestep of 0.5 s, which generates the boundary conditions for the Cape Fear subdomain. Performing a full run requires 1080 CPU-hours of computation time, whereas the same simulation on the extracted subdomain requires only 28 CPU-hours, about 2.6% of the original time.

To demonstrate convergence, several simulations of Hurricane Fran are performed on the Cape Fear subdomain with boundary conditions forced at varying time intervals. Fig. 10 shows differences in maximum water surface elevations between the original and subdomain runs, measured in millimeters, when the forcing interval N is incrementally reduced from 3200 down to 1 timestep, i.e., from every 26 min 40 s to every 0.5 s. Areas where elevations are overpredicted show up as positive differences in the plots. The coarse temporal resolution of surface elevations along the boundary when $N = 3200$ yields underpredicted values, as one would expect of hydrograph peaks clipped by interpolation; these effects are gradually reduced and become imperceptible at $N = 1$.

Because forcing intervals can be varied, the tradeoff between space requirements and accuracy may also be explored. From the full run, boundary condition data are generated in a form that requires minimal computational effort in post-processing to produce the input files needed for subdomain runs. Writing data at every timetep, when N is set to 1, generates about 8.4 GB of output, whereas only about 11 MB of output are generated when N is set to 800. For the Cape Fear subdomain and in other experiments, we observe that a forcing interval of $N = 100$ provides an accuracy nearly on par with the smallest intervals, and does so with substantially reduced data requirements, just 86 MB in this case. That $N = 100$ yields accurate results is not surprising, since it corresponds to the case where hydrograph values, for instance, are enforced on the boundaries with exact values every 50 s and interpolated ones in the intervals.

As an example of making a local, topographic change, we consider an area on the Cape Fear River, shown in Fig. 11, near the intake canal of the Brunswick Nuclear Plant (not shown). The area is zoned for heavy industry and military but is undeveloped, and is ecologically sensitive given drainage from it into marsh and wetlands. To show that what-if studies of a variety of types could be performed in the area, we arbitrarily raise the topography and add a 2.5-mile protective structure that prevents flooding. We then make use of the prior boundary conditions, unchanged, and repeat a simulation of Hurricane Fran.

As shown in Fig. 12, changes of this nature, and more meaningful ones that are perhaps less substantial, alter the hydrodynamics locally



N = 3,200

N = 1,600

N = 800

N = 1

Fig. 10. Cape Fear, Hurricane Fran: maximum elevation differences at various forcing intervals (N).

but the effects are contained within the subdomain at the space and time scales of interest. Along the boundary, water surface elevations are within a millimeter of their original values, and depth averaged velocities are within a millimeter per second with minor exceptions. Comparisons such as these between modified and unmodified subdomains serve as a preliminary post-processing check on whether a subdomain is of sufficient size, position, and geometry. Comments on the adequacy of this check are offered in the discussion following the example below.

6.2.2. Hurricane Isabel (2003) and Cape Hatteras

From the same western North Atlantic mesh, we extract an elliptic subdomain in the Cape Hatteras area of North Carolina covering 353 sq mi and consisting of 8031 nodes, 15,911 elements, and 150 boundary nodes, as shown in Fig. 13. Then, a 5.5-day simulation of Hurricane Isabel is performed on the original mesh using a timestep of 1.0 s, which generates the boundary conditions for the Cape Hatteras subdomain. Relative to the Cape Fear subdomain, this one covers about the same area but nodal spacing is less dense, the simulation

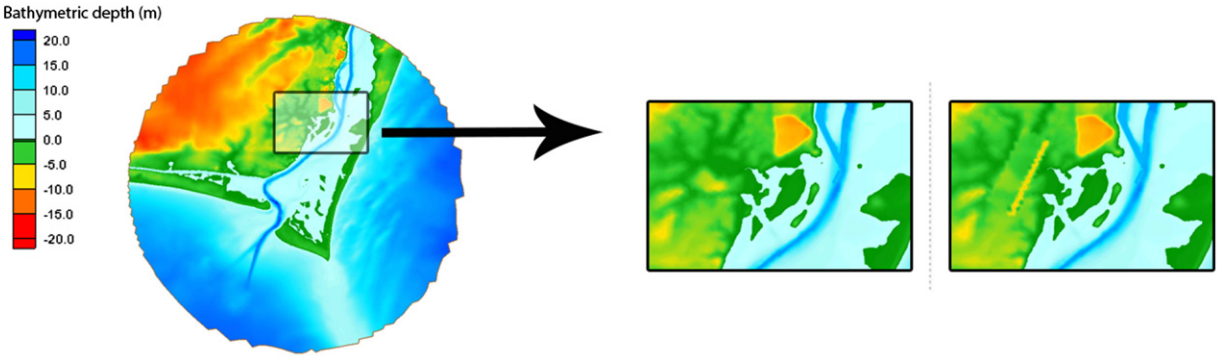


Fig. 11. Introducing a change in the Cape Fear subdomain.

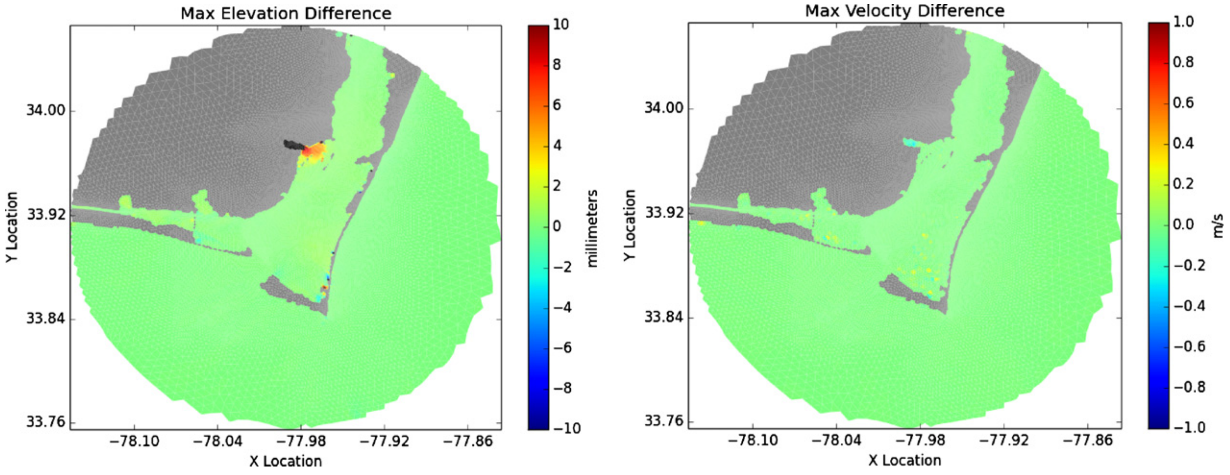


Fig. 12. Cape Fear: extent of altered hydrodynamics.

time and step size are both greater, and water boundaries surround the entire subdomain.

In simulating Hurricane Isabel on the subdomain, we choose to force boundary conditions every 100 timesteps. Fig. 14 shows maximum differences between the original and subdomain runs, with water surface elevations measured in millimeters. Again, positive quantities indicate areas where elevations are overpredicted. The largest difference in water surface elevation in a node is 11 mm, and 99.98% of the nodes have differences less than a millimeter. Depth averaged water velocities are also compared, and 99.93% of the nodes have differences less than 0.1 m/s.

In terms of computation time and space requirements, a 5.5-day simulation of Hurricane Isabel using a timestep of 1.0 s takes 800 CPU-hours and produces 52 MB of boundary condition data from the original full run. To perform the same simulation on the Cape Fear subdomain requires only 20 CPU-hours.

As an example of making a local, topographic change, we consider an area where an inlet formed during Hurricane Isabel, shown in Fig. 15. While not intended to be a precise exercise in hindcasting, the subdomain mesh is refined and terrain lowered to create three separate channels akin to the ones that formed during that event. Then, keeping the boundary conditions from before, a simulation of Hurricane Isabel is performed on the modified subdomain.

A comparison of water surface elevations between the original and modified subdomains in Fig. 16 shows that water surface elevations along the boundary are within a millimeter of their original values, but depth averaged velocities have greater disparity near the boundaries, particularly along the northern perimeter. If these are judged to be insufficiently small, the subdomain can be resized to better contain the changes. Hence, Fig. 17 shows differences in velocities for an enlarged subdomain covering 520 sq mi and consisting of 10,736 nodes, 21,263 elements, and 208 boundary nodes. For this enlarged subdomain, differences in depth averaged velocities are within a millimeter per second around the boundary.

6.2.2.1. Discussion. In subdomain modeling, each storm event is fully simulated initially, with subsequent simulations performed on a smaller domain, thereby eliminating calculations that would fall outside the sphere of influence of any changes that have been introduced. Needless to say, such changes are expected to be comparatively small, and from an engineering perspective many are. The failure of a levee or rebuilding of a stretch of dunes is unlikely to create effects 10 or 20 miles up the coastline, and yet these are precisely the kinds of what-if scenarios needed for infrastructure design and damage prediction.

Of the examples presented, the change introduced in the Cape Hatteras subdomain is the more substantial. And yet, even there, one finds stability between the original and enlarged subdomains with respect to their interior solutions, which are nearly identical except along the northern area. In principle, one might argue that a complete vetting of this or any change requires another simulation of the modified full domain, as we have done for the examples presented. In a design scenario, since many runs are likely to be performed on a subdomain as changes are made, performing a full domain run on the final configuration, as a check, should not be prohibitive. As a preliminary check, we observe that significant solution gradients near the boundary, such as those in the smaller Hatteras subdomain, are a good indicator that a subdomain is too small. This diagnostic approach coincides well with the intuition that incremental increases in the size

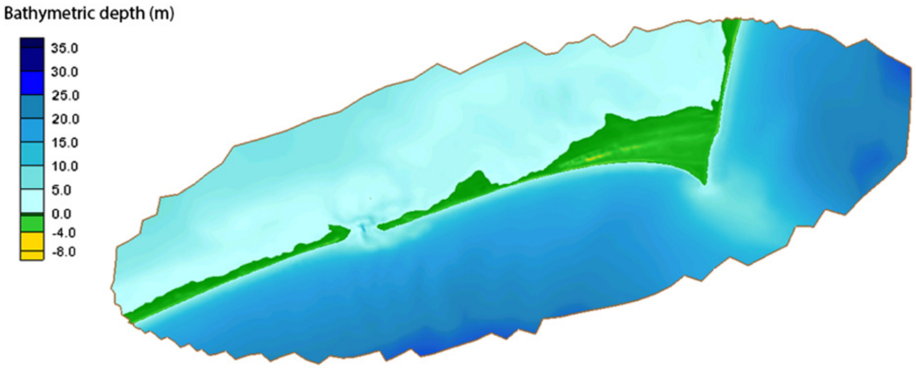


Fig. 13. Cape Hatteras terrain: a subdomain with a length of 33.7 mi and width of 13.4 mi.

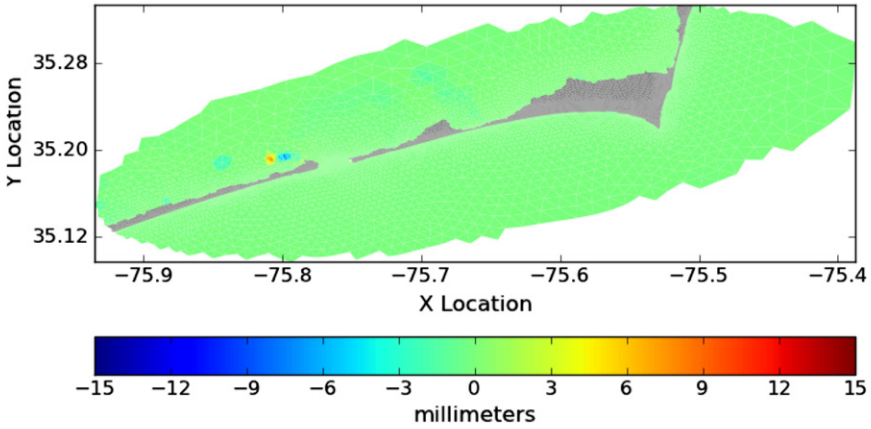


Fig. 14. Cape Hatteras, Hurricane Isabel: maximum elevation differences.

of a subdomain should leave the interior solution unchanged, and this, too, can be implemented in practice as an additional check on the quality of a modified subdomain to gain confidence while undertaking a particular type of modeling activity.

7. User interface and case study.

A graphical user interface, called the ADCIRC Subdomain Modeling Tool (SMT), streamlines the pre- and post-processing requirements of the subdomain modeling technique (Dyer, 2013). The tool automates steps 1 through 4 of the workflow outlined in Fig. 6, and allows the user to visualize and interact with ADCIRC meshes. To fully leverage the power of the subdomain modeling technique, SMT also provides project management, so that a user can quickly and simply create and run a large number of subdomains from a single full domain without having to worry about file management.

SMT has the goal of being a tool capable of managing the large datasets associated with ADCIRC and subdomain modeling in a highly responsive graphical environment. To achieve this goal, SMT is implemented in C++ and OpenGL, yielding a high performance backend for file I/O and interactive graphics display. Additionally, a novel range search algorithm – operating on a quadtree data structure containing a large-scale mesh – enables users to create arbitrary shapes and instantly extract the nodes falling within the shape as a subdomain. The tool is open source and can be obtained by contacting the authors.

7.1. Introduction to the SMT interface

Fig. 18 introduces the SMT window frame, which is split into two main sections. The right side displays an interactive view of the currently selected ADCIRC mesh and provides panning, zooming, and selection functionality. It is this panel that the user interacts with when selecting nodes to be included in a subdomain and when editing the nodal properties of subdomain meshes. The left side displays a stack of modules, each of which provides tools that aid the user in performing the individual steps of the subdomain modeling procedure. The modules included are:

Project Explorer: displays the project structure and allows the user to change the currently visible mesh.
Create New Subdomain: provides selection tools, allowing the user to identify nodes using various drawing shapes (e.g., rectangle, circle, and polygon), either individually or in combination. Automates the file creation process on completion.
Edit Subdomain: provides tools used to edit all nodal properties in subdomains.
ADCIRC: provides a front-end for running ADCIRC and ensures that domains are run in the proper order.
Analyze Results: provides plotting and drawing tools that can be used to compare results between domains (a feature still under development).

7.2. SMT case study

The following case study demonstrates the effectiveness of the subdomain modeling technique by using SMT to create and run several subdomains. In this particular example, we consider a modeler whose work is focused on two specific areas: a small portion of the Cape Fear River near Silver Lake in Wilmington, NC, and the Masonboro Inlet, just south of Wrightsville Beach, NC.

In the case of the Silver Lake region, the modeler wishes to vary Manning's n along the river bank over a number of ADCIRC simulations to determine the materials ecologically best suited to lowering water velocities during a hurricane event. The modeler needs to run at least ten different simulations, each with a different value of Manning's n in the region of interest. In the case of the Masonboro Inlet, she wishes to examine the effects of several possible beach nourishment projects on surge heights and water velocities through the inlet during a storm

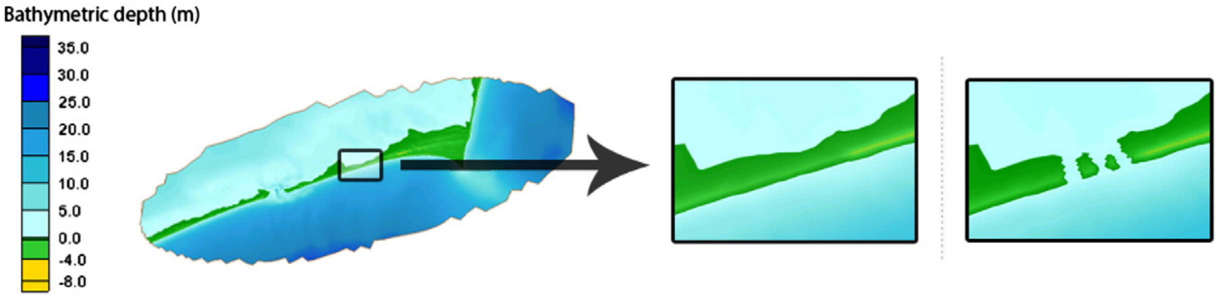


Fig. 15. Introducing a change in the Cape Hatteras subdomain.

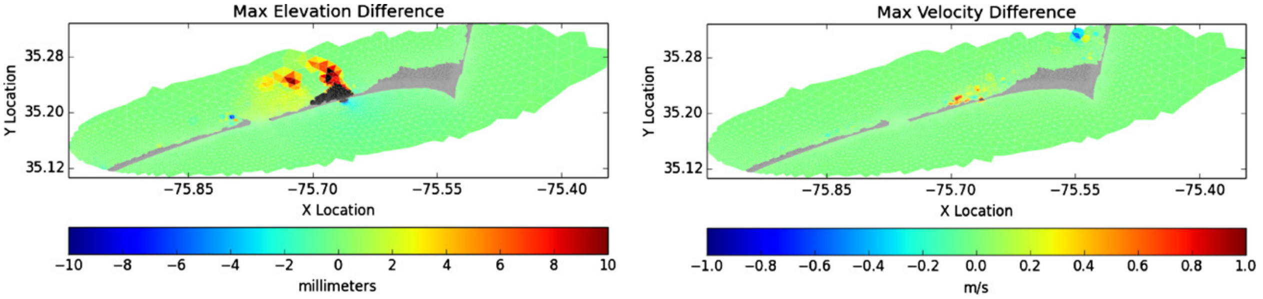


Fig. 16. Cape Hatteras: extent of altered hydrodynamics.

event. This study involves running at least five different ADCIRC simulations, each with minor changes made to nodal elevations on the northern beach of the inlet.

7.2.1. Creating the subdomains

Working with subdomains in SMT starts with a new project and the selection of a full domain, as previously shown in Fig. 18, at which point the modeler is ready to begin creating subdomains. Choosing an appropriate size is a matter of judgment, though subdomains of varying shapes and sizes can be created and extracted simultaneously from a full run if the extent of hydrodynamic effects of modifications are difficult to anticipate.

In the Silver Lake region, the modeler is only varying the value of Manning's *n* and does not expect the resulting changes in water velocity to propagate far, allowing her to create a very small subdomain, as shown in Fig. 19a. Once the subdomain has been created, any number of duplicates can be added to the project; Fig. 19b shows eleven of them, one left unmodified for purposes of verification, and ten others named according to the new value of Manning's *n* that will be used in the area of interest.

For the Masonboro Inlet study area, the modeler anticipates more substantial changes and a broader extent of their effects, and therefore chooses a larger subdomain size, as shown in Fig. 19c. After creating all duplicate subdomains, including one for verification, the modeler now has the full set listed within the Project Explorer shown in Fig. 19d.

7.2.2. Running ADCIRC and Editing subdomains

With all subdomains created, the modeler is now ready to perform the full domain ADCIRC run, which records the boundary conditions for both the Silver Lake and Masonboro Inlet study areas. During the run, the modeler may begin making the necessary changes to the subdomain models using the tools provided in the 'Edit Subdomain' module.

Once the full domain run has completed, any of the subdomain runs can be performed, typically beginning with unmodified ones for verifying that an appropriate recording frequency has been used. SMT allows modelers to select a set of subdomain runs to be performed, as shown in Fig. 20. When running a subdomain, SMT applies the appropriate boundary conditions from the full domain run.

A summary of the three ADCIRC domains being used in this case study is presented in Table 1. The computational benefits of using the subdomain modeling technique can be seen very clearly when considering the number and sizes of the ADCIRC runs required by the modeler. Without using it, almost 20,000 CPU-hours would be required to perform the 18 runs, in contrast with the approximately 1128 CPU-hours actually used. The time cost to the modeler in setting up and extracting subdomains is minimal: about 15 min for each of the two study areas presented using SMT. After-the-fact checks on subdomain quality can likewise be performed in minutes using the computational tools provided for comparing simulation results.

8. Conclusion and future work

Storm surge modeling can be used to assess the effects of hurricanes on coastal regions and to determine the weaknesses of civil infrastructure systems, to formulate actions and countermeasures, and to implement better designs for enhanced resilience. Because such simulations are costly, however, employing them as part of an iterative design strategy may be impractical without some modification.

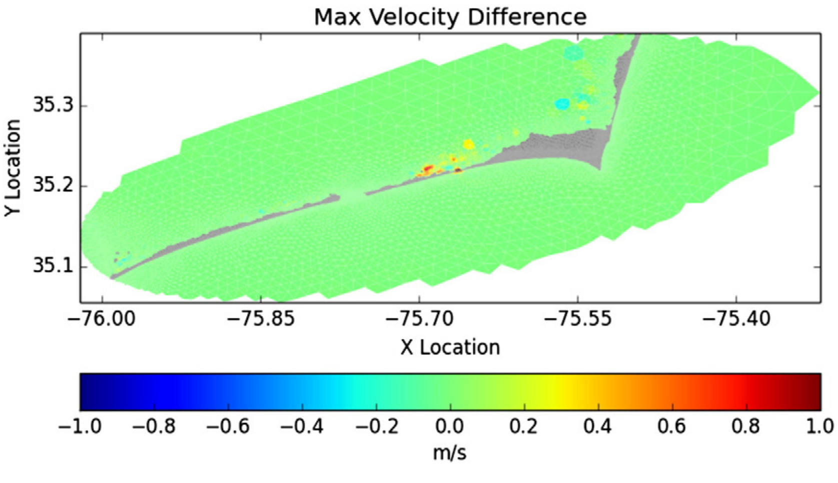
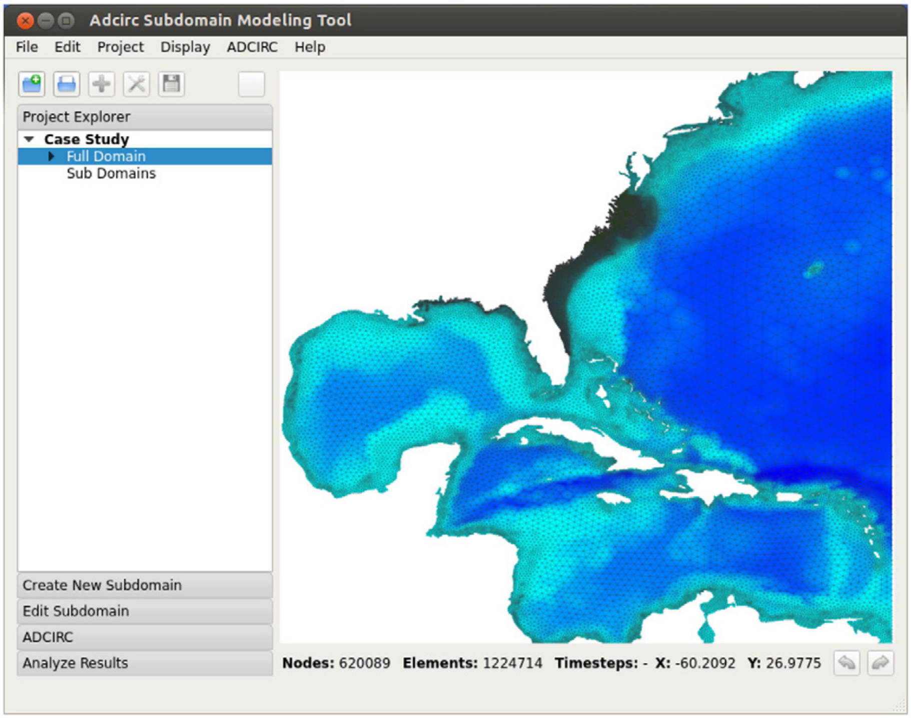


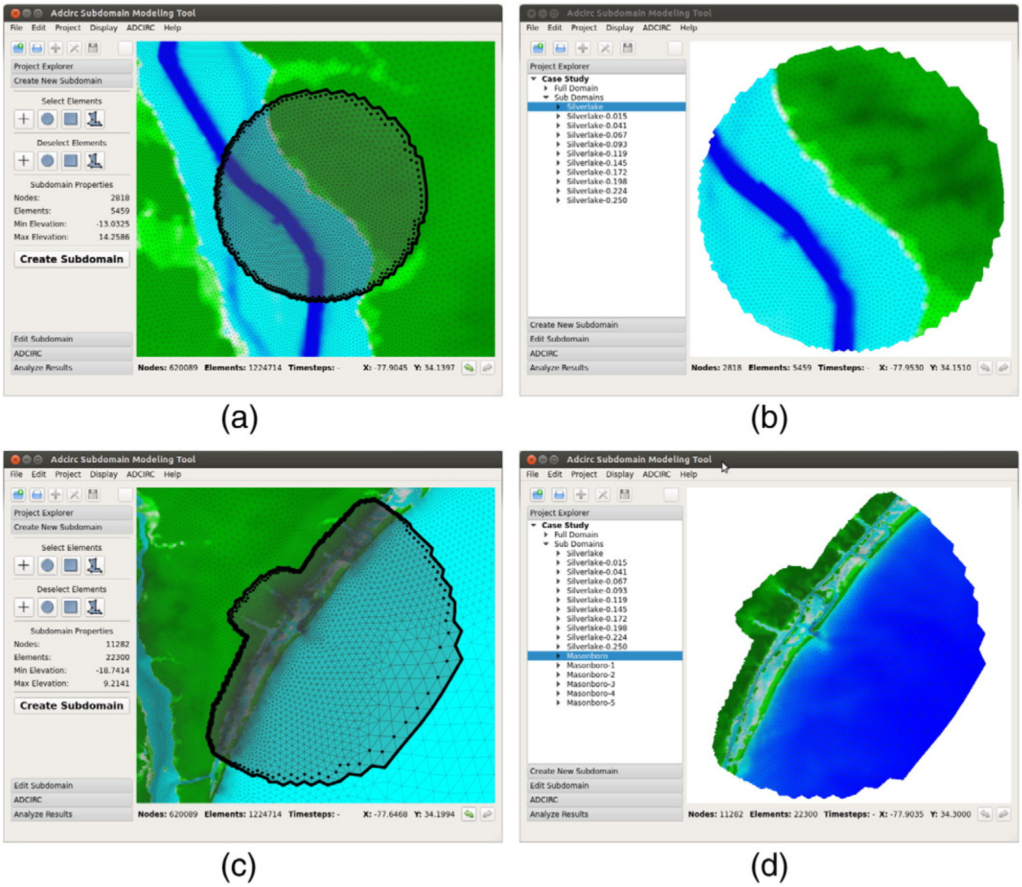
Fig. 17. Cape Hatteras enlarged subdomain with a length of 41.0 mi and width of 16.1 mi.



Case Study
Full Domain
Sub Domains

Nodes: 620089 Elements: 1224714 Timesteps: - X: -60.2092 Y: 26.9775

Fig. 18. SMT view of the full domain used in the newly created Silver Lake project.



(a)

Nodes: 2818
Elements: 5459
Min Elevation: -13.0325
Max Elevation: 14.2586

Nodes: 620089 Elements: 1224714 Timesteps: - X: -77.9045 Y: 34.1397

(b)

Case Study
Full Domain
Sub Domains
Silverlake
Silverlake-0.015
Silverlake-0.041
Silverlake-0.067
Silverlake-0.093
Silverlake-0.119
Silverlake-0.145
Silverlake-0.172
Silverlake-0.198
Silverlake-0.224
Silverlake-0.250

Nodes: 2818 Elements: 5459 Timesteps: - X: -77.9530 Y: 34.1510

(c)

Nodes: 11282
Elements: 22300
Min Elevation: -18.7414
Max Elevation: 9.2141

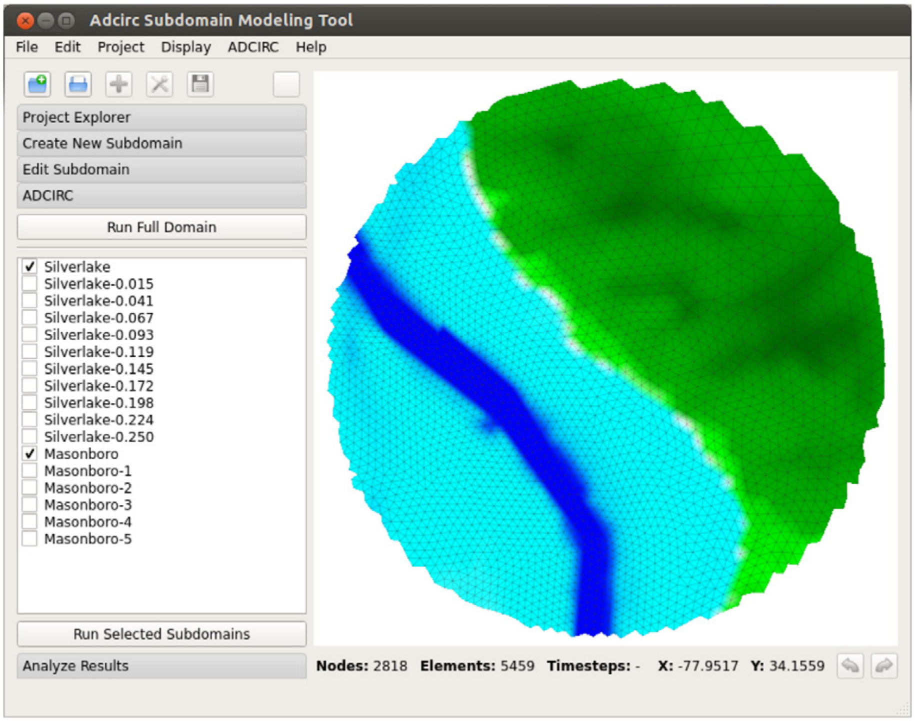
Nodes: 620089 Elements: 1224714 Timesteps: - X: -77.6468 Y: 34.1994

(d)

Case Study
Full Domain
Sub Domains
Silverlake
Silverlake-0.015
Silverlake-0.041
Silverlake-0.067
Silverlake-0.093
Silverlake-0.119
Silverlake-0.145
Silverlake-0.172
Silverlake-0.198
Silverlake-0.224
Silverlake-0.250
Masonboro
Masonboro-1
Masonboro-2
Masonboro-3
Masonboro-4
Masonboro-5

Nodes: 11282 Elements: 22300 Timesteps: - X: -77.9035 Y: 34.3000

Fig. 19. (a) Selecting subdomain nodes for the Silver Lake region. (b) Creating eleven copies of the Silver Lake subdomain. (c) Selecting subdomain nodes for Masonboro Inlet. (d) All subdomain copies created for both study areas.



☑ Silverlake
□ Silverlake-0.015
□ Silverlake-0.041
□ Silverlake-0.067
□ Silverlake-0.093
□ Silverlake-0.119
□ Silverlake-0.145
□ Silverlake-0.172
□ Silverlake-0.198
□ Silverlake-0.224
□ Silverlake-0.250
☑ Masonboro
□ Masonboro-1
□ Masonboro-2
□ Masonboro-3
□ Masonboro-4
□ Masonboro-5

Nodes: 2818 **Elements:** 5459 **Timesteps:** - **X:** -77.9517 **Y:** 34.1559

Fig. 20. Verifying the unchanged subdomains before running the modified subdomains.

We describe an approach that allows model changes within a geographic region of interest without requiring a full-scale simulation of a storm event for each change under consideration. The use of water surface elevations, depth averaged velocities, and wet/dry status are shown to be practical and effective as boundary conditions in engineering scale models, where the intent is to accommodate model changes corresponding to infrastructure improvements or failure scenarios. To validate a subdomain model, a straightforward check can be performed against the original simulation, and upon making a local change, another check can be performed to gain confidence that a subdomain is large enough to contain the altered hydrodynamics.

For all of its advantages, subdomain modeling does introduce minor burdens in the form of new modeling concerns that require some judgment. One must choose a size and shape for each subdomain, along with a forcing interval, *N*, that determines how frequently boundary conditions are written during a full run. On the former, it is clear that subdomain boundaries should be located far enough from modifications so that their effects are not felt on those boundaries. What may be less clear is the anticipated spatial extent of those effects. In such cases we recommend that a small number of subdomains of varying size and potentially shape be defined and extracted over the region, since doing so comes at no increase in computational cost. Then, data files are on hand for either larger or smaller subdomains as needed. On forcing interval, an appropriate choice depends on timestep size so that the combination yields interpolated values that reasonably match elevation and velocity time histories, as well as wet/dry changes. The only tradeoff here is with respect to file sizes and available storage.

Although implemented in ADCIRC, in principle the approach could be realized within other numerical modeling schemes, whether based on finite element methods or otherwise. To extract subdomains, a first step is identifying the temporal and spatial data dependencies introduced by the algorithms operating over a grid. In the case of the GWCE, a linear algebraic relationship constitutes global dependencies at a node that can be used to factor out effects external to the subdomain. The wetting and drying algorithm, though locally applied to a node and its neighbors, includes a series of state changes that complicate the extraction. The momentum equations, on the other hand, also have only local data dependencies but the nodal state updates are trivial to incorporate.

While not presented here, additional enhancements to ADCIRC by the authors allow models to be trimmed temporally as well as spatially, while still allowing local changes in a subdomain (Altuntas, 2012). The approach further reduces run time, often dramatically, and draws on ADCIRC's hot-start feature, which is ordinarily used to restart a simulation in the event of a computer failure during a run (Westerink et al., 1994).

Other extensions of the approach include a coupling between ADCIRC and SWAN (Booij et al., 1999; Ris et al., 1999) that makes use of wave information along the subdomain boundary. SWAN, a third-generation wave model that solves the spectral action balance equation, allows modelers to enforce the boundary of its computational domain with any of the three available boundary condition files: TPAR files and 1D or 2D spectra files. A typical TPAR file contains significant wave height, wave period, peak direction, and directional spread for each forcing time. Nonstationary spectra files contain either energy or variance density of a location that are used to enforce wave spectra on a boundary. While both types of boundary conditions may be used for subdomain modeling, 2D spectra files appear to offer better accuracy. Assessment and application of the approach are ongoing.

Finally, beyond supporting what-if scenarios, we note that subdomain modeling might also be used to facilitate formal optimization strategies for coastal development, such as resilience models that take into account impact from storm surge and probability of occurrence (Niedoroda et al., 2008; Piper, 2009). Ultimately, one can imagine it enabling a full-featured design environment that supports a suite of analysis tools for joint-cognitive decision making. Work is proceeding in these and other directions.

Table 1
Comparison of the three domains used in this case study.

Domain	# Nodes	# Elements	% Full domain	# Runs	CPU-hours/run
Full domain	620,089	1,224,714	100.0	1	1,080
Masonboro	11,282	22,300	1.8	6	19.4
Silver Lake	2,818	5,459	0.4	11	4.32

Acknowledgments

The authors thank Rick Luettich and Brian Blanton for their helpful feedback and for providing the base files used in this study, including the NC FEMA meshes and wind files. We also thank Julie Rutledge for refinements of the mesh at Cape Hatteras, and Yoonhee Park for motivating the Silver Lake case study from the perspective of a landscape architect.

This material is based upon work supported by the Coastal Hazards Center of Excellence, a US Department of Homeland Security Science and Technology Center of Excellence under Award Number 2008-ST-061-ND 0001 and a US DHS HS-STEM Career Development Grant.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DHS.

References

Altuntas, A., 2012. Downscaling Storm Surge Models for Engineering Applications. (Master's thesis). North Carolina State University, Raleigh, NC.

Altuntas, A., Simon, J.S., Baugh Jr., J.W., 2013. Subdomain ADCIRC v. 50 User Guide. Technical Report. North Carolina State University, Raleigh, NC (CE-308-13).

Arora, J., 1976. Survey of structural reanalysis techniques. J. Struct. Div. 102, 783–802.

Bacopoulos, P., Funakoshi, Y., Hagen, S.C., Cox, A.T., Cardone, V.J., 2009. The role of meteorological forcing on the St. Johns River (Northeastern Florida). J. Hydrol. 369, 55–70.

Blain, C., Westerink, J., Luettich, R., 1994. The influence of domain size on the response characteristics of a hurricane storm surge model. J. Geophys. Res. Oceans (1978–2012) 99, 18467–18479.

Booij, N., Ris, R., Holthuijsen, L.H., 1999. A third-generation wave model for coastal regions: 1. Model description and validation. J. Geophys. Res. Oceans (1978–2012) 104, 7649–7666.

Cailleau, S., Fedorenko, V., Barnier, B., Blayo, E., Debreu, L., 2008. Comparison of different numerical methods used to handle the open boundary of a regional ocean circulation model of the Bay of Biscay. Ocean Model. 25, 1–16.

Debreu, L., Blayo, E., 1993. Two-way embedding algorithms: a review. Ocean Dyn. 58, 415–428.

Dietsche, D., 2004. Storm Tide Simulations for Hurricane Hugo (1989): On the Significance of Including Inland Flooding Areas. (Master's thesis). University of Central Florida, Orlanda, Fla.

Dietsche, D., Hagen, S.C., Bacopoulos, P., 2007. Storm surge simulations for Hurricane Hugo (1989): on the significance of inundation areas. J. Waterw. Port Coast. Ocean Eng. 133, 183–191.

Drolet, J., Gray, W.G., 1988. On the well posedness of some wave formulations of the shallow water equations. Adv. Water Resour. 11, 84–91.

Dyer, A.T., 2013. An Interface for Subdomain Modeling using a Novel Range Search Algorithm for Extracting Arbitrary Shapes. (Master's thesis). North Carolina State University, Raleigh, NC.

Elvius, T., Sundstrom, A., 1973. Computationally efficient schemes and boundary conditions for a finemesh barotropic model based on the shallowwater equations. Tellus 25, 132–156.

Engquist, B., Majda, A., 1977. Absorbing boundary conditions for numerical simulation of waves. Proc. Natl. Acad. Sci. 74, 1765–1766.

Funakoshi, Y., Hagen, S.C., Bacopoulos, P., 2008. Coupling of hydrodynamic and wave models: case study for Hurricane Floyd (1999) hindcast. J. Waterw. Port Coast. Ocean Eng. 134, 321–335.

Garratt, J.R., 1977. Review of drag coefficients over oceans and continents. Mon. Weather Rev. 105, 915–929.

Kassim, A.A., Topping, B.H.V., 1987. Static reanalysis: a review. J. Struct. Eng. 113, 1029–1045.

Kreiss, H.O., 1970. Initial boundary value problems for hyperbolic systems. Commun. Pure Appl. Math. 23, 277–298.

Luettich, R.A., Westerink, J.J., 1999. Elemental wetting and drying in the ADCIRC hydrodynamic model: upgrades and documentation for ADCIRC Version 34.XX. Technical Report. Dept. of the Army, U.S. Army Corps of Engineers. Waterways Experiment Station, Vicksburg, MS (http://www.adcirc.org).

Luettich, R.A., Westerink, J.J., 2004. Formulation and numerical implementation of the 2D/3D ADCIRC finite element model version 44.xx. http://www.adcirc.org.

Luettich, R.A., Westerink, J.J., 2010. User's manual v49: A (parallel) advanced circulation model for oceanic, coastal and estuarine waters. http://www.adcirc.org.

Luettich, R.A., Westerink Jr., J.J., Scheffner, N.W., 1992. ADCIRC: an advanced three-dimensional circulation model for shelves, coasts and estuaries, report 1: theory and methodology of ADCIRC-2DDI and ADCIRC-3DL. Technical Report DRP-92-6. Dredging Research Program. U.S. Army Engineers Waterways Experiment Station, Vicksburg, MS.

Lynch, D.R., Gray, W.G., 1978. Analytic solutions for computer flow model testing. ASCE J. Hydraul. Div. 104, 1409–1428.

Marchesiello, P., McWilliams, J.C., Shchepetkin, A., 2001. Open boundary conditions for long-term integration of regional oceanic models. Ocean Model. 3, 1–20.

Niedoroda, A.W., Resio, D.T., Toro, G., Divoky, D., Reed, C., 2008. Efficient strategies for the joint probability evaluation of storm surge hazards. Proceedings of the ASCE Conference on Solutions to Coastal Disasters, ASCE, Turtle Bay, Hawaii, pp. 242–255.

Piper, B.E., 2009. A Decision Framework for Improving Resilience of Civil Infrastructure Systems Considering Effects of Natural Disasters. (Master's thesis). North Carolina State University, Raleigh, NC.

Ris, R., Holthuijsen, L., Booij, N., 1999. A third-generation wave model for coastal regions: 2. Verification. J. Geophys. Res. Oceans (1978–2012) 104, 7667–7681.

Salisbury, M.B., Hagen, S.C., 2007. The effect of tidal inlets on open coast storm surge hydrographs. Coast. Eng. 54, 377–391.

Simon, J.S., 2011. A Computational Approach for Local Storm Surge Modeling. (Master's thesis). North Carolina State University, Raleigh, NC.

Spall, M.A., Robinson, A.R., 1989. A new open ocean, hybrid coordinate primitive equation model. Math. Comput. Simul. 31, 241–269.

Tanaka, S., Bunya, S., Westerink, J.J., Dawson, C., Luettich, R.A., 2011. Scalability of an unstructured grid continuous Galerkin based hurricane storm surge model. J. Sci. Comput. 46, 329–358.

U.S. Army Corps of Engineers, 2012. Flood risk management manual. Washington, D.C. National Economic Development (NED) Manual.

Westerink, J.J., Luettich, R.A., Baptista, A.M., Scheffner, N.W., Farrar, P., 1992. Tide and storm surge predictions using finite element model. J. Hydraul. Eng. 118, 1373–1390.

Westerink, J.J., Luettich, R.A., Feyen, J.C., Atkinson, J.H., Dawson, C., Roberts, J.J., Powell, M.D., Dunion, J.P., Kubatko, E.J., Pourtaheri, H., 2008. A basin- to channel-scale unstructured grid hurricane storm surge model applied to southern Louisiana. Mon. Weather Rev. 136, 833–864.

Westerink, J.J., Luettich, R.A., Scheffner, N.W., 1994. ADCIRC: an advanced three-dimensional circulation model for shelves, coasts, and estuaries, report 2: user's manual for ADCIRC-2DDI. Technical Report. Department of the Army. Waterways Experiment Station, Vicksburg, MS (DRP-92-6).