OCSMesh and an end-to-end workflow for fully automatic mesh generation with application to compound flood studies

Felicio Cassalho[1] [2], Soroosh Mani[1], Fei Ye[3], Y. Joseph Zhang[3], Saeed Moghimi[1]

[1]Coast Survey Development Laboratory, NOAA, Silver Spring, MD 20910, USA
[2]Ocean Associates, Incorporated, Arlington, VA 22207, USA
[3]Virginia Institute of Marine Science, William & Mary, Gloucester Point, VA 23062, USA

Abstract

Despite great efforts towards automation and reproducibility, mesh generation remains a major 'gray area' in coastal ocean modeling. In this manuscript we introduce the NOAA Office of Coast Survey (OCS) mesh generation tool 'OCSMesh', an end-to-end data-driven, object-oriented, unstructured mesh generation Python tool that uses geometry and sizing function objects to define mesh domain and resolution, primarily targeting SCHISM applications. OCSMesh leverages SCHISM's greater flexibility and robustness with skewed elements to deliver faithful representations of the DEM (Digital Elevation Model) at different modeling scales (i.e., creek-to-ocean). The main novelties OCSMesh brings are its relentless pursuit of fidelity to the underlying DEM (e.g., capable of resolving arbitrarily small channels with no restriction, which is essential for compound flood studies), greater efficiency, and active and continued development. We demonstrate OCSMesh applications with two examples. In the first, we reproduce a cross-scale mesh used by a SCHISM-based 3-dimensional operational forecast system. Results show that the OCSMesh-based model can reproduce the skill of the operational model while taking a fraction of the time needed for mesh development. The second example shows how the highly efficient OCSMesh mesh merging process can be used in a relocatable forecasting scenario. We show that within reasonable time (< 2-hours) we are able to not only create a high-resolution mesh for a large river basin, but also merge it into the coarser operational mesh. With that approach we can deliver more realistic flood predictions for the area of interest, which are essential for timely flood hazard mitigation.

Keywords: cross-scale coastal modeling, open-source, Python, SCHISM, triangulation, STOFS3D.

1. Introduction

Unstructured grid (UG) models have been increasingly used in cross-scale coastal ocean modeling studies. This trend is largely attributed to UG's greater flexibility and accuracy in capturing the terrain by enabling localized refinement or coarsening as a function of the regions' relevance and project-specific resolution requirements (Ye et al., 2019). Mesh generation is the most laborious and consequential step during the development of an ocean circulation model. This step is often performed by humans with the aid of software based on a graphical user interface and guided by a highly subjective decision-making process that fails to promote automation, objectivity, and reproducibility (Conroy et al., 2012; Roberts et al., 2019). Several *automated* unstructured mesh generation packages have been developed in order to address these limitations (Gorman et al., 2008; Conroy et al., 2012; Remacle and Lambrechts, 2018; Roberts et al., 2019; Trotta et al., 2021). However, despite these efforts, mesh generation is still regarded as an 'art' as it requires almost unachievable trade-offs between accuracy (i.e., faithful representation of topographic and bathymetric features), computational cost (i.e., mesh resolution in terms of the node and element count), and mesh 'quality' (e.g., regularity of triangles or other types of cells). Given that computational costs and model stability are generally non-negotiable requirements, this trilemma is usually resolved at the expense of mesh accuracy.

Automated mesh generation packages are often designed envisioning its application towards a particular modeling system, and as a result, they tend to compensate for the limitations of these models. For instance, to accommodate the numerical instabilities inherent in many ocean models, modern automated mesh generation packages are designed with advanced features to ensure equilateral triangulation and smooth transitions in mesh resolution (Gorman et al., 2008; Conroy et al., 2012; Roberts et al., 2019). While these features are crucial for the numerical stability of targeted models, an excessive focus on smoothness and 'mesh quality' inevitably leads to the creation of unnecessarily larger meshes (Ye et al., 2023). This in turn increases computational costs, thus further skewing the mesh generation trilemma by sacrificing both accuracy and costs in favor of model stability. Another common feature in those packages is Digital Elevation Models (DEMs) smoothing before and after the interpolation of topobathy data into the mesh. This common practice, prevalent in the literature, would fundamentally alter the characteristics of the DEM and thus introduce system-wide changes in the simulated processes that modelers would then attempt to erroneously compensate during model calibration (Ye et al., 2018; Cai et al. 2020; Zhang et al., 2024). For example, most published works in coastal ocean modeling focus on the choice of model, algorithm, and parameters, but glossed over the critical role played by mesh representation of the DEM, therefore making the analysis dubious at best (Zhang et al. 2024). Therefore, we believe that the development of an objective, defensible, reproducible, and accuracy-focused automated mesh generation package is best based on numerically robust ocean models that free the mesh generation process from these mesh 'quality' error compensations.

SCHISM is a community-supported, open-source model that stands out for its ability to seamlessly simulate 3D baroclinic circulation across creek-to-ocean scales using highly efficient semi-implicit finite-element/finite-volume methods for solving a wide range of hydrodynamic processes and biogeochemical applications (Zhang and Baptista, 2008; Cai et al. 2021). SCHISM also strives to faithfully ingest the observed bathymetry by allowing skewed horizontal mesh elements as well as mixed-element meshes (triangular and quadrangular), thus waiving the need for bathymetry smoothing and other types of artificial mesh manipulations (Zhang et al., 2016; Zhang et al. 2024). In fact, judicious use of skewed elements is particularly advantageous in nearshore regime due to the numerous competing meshing constraints. For example, defense structures like levees and jetties are usually located in close proximity to navigation channels. The combination of these horizontal mesh tolerances with highly configurable vertical grid schemes leads to extremely powerful 'polymorphism' (a single SCHISM grid allows 1D/2D/3D configurations in different regions) that can effectively support a myriad of real-world applications (Zhang et al., 2015; Zhang et al. 2016). To fully leverage SCHISM's robustness, the mesh generation process must be re-oriented towards two *in silico* oceanography principles described in Zhang et al. (2024): i) the underlying terrain is a first order model forcing, and as such should not be smoothed beyond the resolution of the unstructured mesh; and ii) the mesh resolution and domain extent should be dependent on the simulated processes (and thus, is not a consequence of the numerical instabilities of the targeted model).

In this paper we introduce the NOAA Office of Coast Survey (OCS) mesh generation tool 'OCSMesh', an automated unstructured mesh generation Python package targeting SCHISM-based applications. OCSMesh was designed to deliver faithful representation of the DEM by leveraging SCHISM's greater flexibility and efficiency with skewed triangular and optional quadrilateral elements. Different from other mesh generation packages, the main novelties

OCSMesh brings are its relentless pursuit of fidelity to the underlying DEM (e.g., capable of resolving arbitrarily small channels with no restriction, which is essential for compound flood studies), greater efficiency, and active and continued development. The software features a suite of specialized functions specifically designed to translate SCHISM's mesh generation requirements into continuous cross-scale meshes that extend from small creek systems to expansive oceanic domains. OCSMesh's capability of individually generating and seamlessly merging different mesh domains (i.e., river systems, floodplain, and ocean) ensures that the generated meshes are well-suited for integration within operational modeling frameworks and relocatable ocean modeling platforms, enhancing the effectiveness and flexibility of simulations across contrasting spatial scales. OCSMesh is freely distributed via GitHub[1] and can easily be installed on different operating systems. In fact, a few user groups have successfully utilized this tool in their own mesh development work (Mani et al., 2022; Sun et al., 2022; Martins et al., 2024).

This manuscript is structured as follows: Section 2 describes the application cases, including the model setup and short synoptic history of the major events during the simulation; Section 3 describes the OCSMesh main functions and end-to-end mesh generation workflow; Sections 4 and 5 present the outcomes of the two application cases; Section 6 presents the final discussions and conclusions.

2. Description of Application Cases

Floods can be driven by different types of forcings of fluvial (riverine flood wave), pluvial (direct precipitation), and coastal (storm surges) origins. If these all occur simultaneously and in close progression of each other they can exacerbate flood impacts (Wahl et al., 2015). State-of-the-art models have demonstrated their capacity to combine features of traditional hydrology and ocean modeling through the development of seamlessly integrated creek-to-ocean model meshes, which leads to more accurate representations of the compounding effects of coastal floods (Ye et al., 2020). Here we developed two cross-scale examples to demonstrate 1) OCSMesh mesh generation workflow and mesh merging processes, and 2) its deployability during major

[1] OCSMesh GitHub Repo; URL: https://github.com/noaa-ocs-modeling/OCSMesh, last accessed in Mar 2025.

compound-flooding events. These examples were set up based on the Surge and Tide Operational Forecast System Three-Dimensional Component for the Atlantic Basin (STOFS3D-Atlantic[2]) model, and the outputs from the OCSMesh-based model were compared to STOFS3D-Atlantic's outputs.

The first example shows how OCSMesh performs the very laborious and error-prone task of creating cross-scale meshes. More specifically, this example demonstrates how OCSMesh generates a size function-based floodplain mesh, then triangulates a continuous drainage network (creek, rivers, and channels) derived from representations of the thalwegs and river banks in the DEM, and finally merges them into STOFS3D-Atlantic ocean mesh (Figure 1). The goal of this example is to assess whether OCSMesh can reasonably develop a seamless creek-to-ocean mesh for the entire US East Coast in a timely manner and produce model estimates comparable to that of STOFS3D.

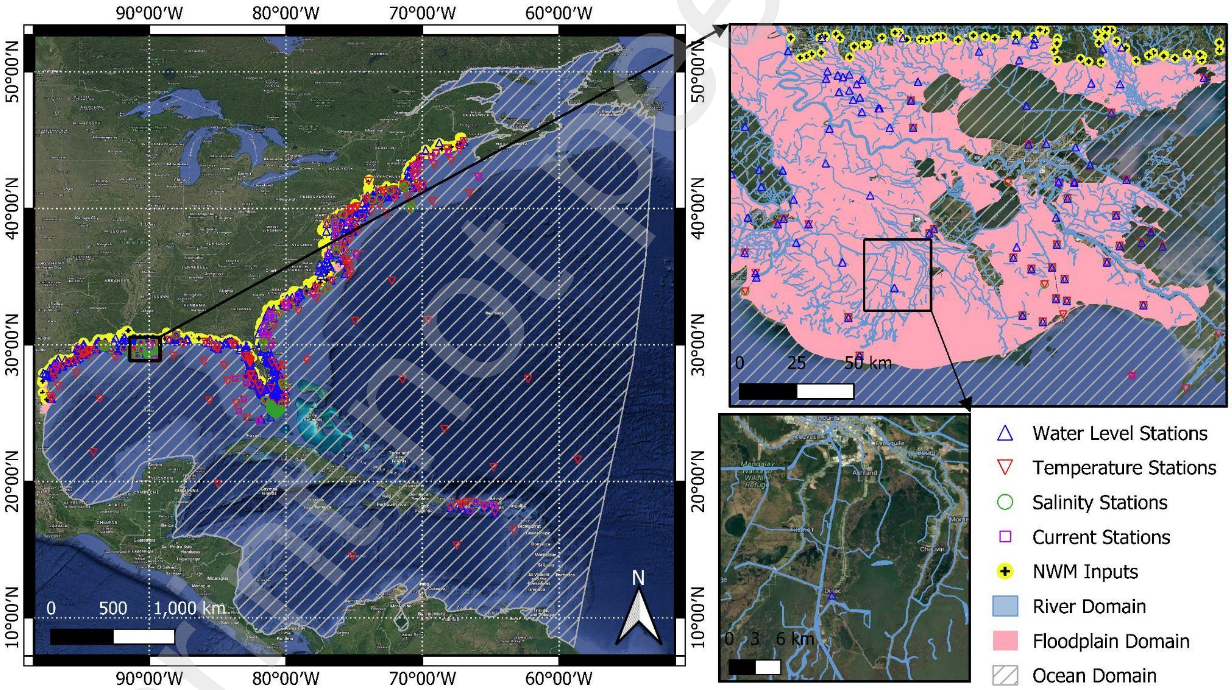

Figure 1. Geographical domain of the STOFS3D-Atlantic mesh, *in-situ* data locations used for model output comparison, and National Water Model-based river heads.

[2] STOFS-3D-Atlantic: 3-D Surge and Tide Operational Forecast System for the Atlantic Basin; URL: https://polar.ncep.noaa.gov/estofs/, last accessed in Mar 2025.

The second example demonstrates the software's potential deployability within a relocatable operational modeling framework during an emergency scenario. A frequent request from users when approaching coastal flood events is high-resolution data, which can be obtained by deploying a separate nested high resolution numerical model into larger-scale ocean forecasts (Trotta et al., 2021). Here we used a different approach where a high-resolution compound flood mesh for the Pearl River Basin between the states of Mississippi and Louisiana was created using OCSMesh and merged into the STOFS3D-Atlantic mesh with the objective of conserving computational costs (by increasing resolution only where necessary) while downscaling processes in the area of interest. Our goal with this example is to evaluate whether OCSMesh's mesh generation and merging processes can be efficiently used in relocatable cases by seamlessly merging the high-resolution mesh generated for a target area into a baseline mesh in a timely manner (between forecast cycles).

Modeled total water levels, water temperature, salinity, and surface velocities were compared at 1220 stations, including 744 water level, 308 temperature, 115 water salinity, and 53 currents gauges corresponding to the location of existing CO-OPS, USGS, and NDBC stations (Figure 1). The location of these stations was used for creating point-based model outputs for further skill assessment (Section 4.2.2).

2.1. Baseline Model for Comparison

In its operational version STOFS3D-Atlantic uses NOAA's High-Resolution Rapid Refresh (given high-priority) and Global Forecast System (given low-priority) as surface meteorological forcing. The global operational Real-Time Ocean Forecast System (G-RTOFS) is used for subtidal three-dimensional boundary forcing of water temperature, salinity, and currents. Subtidal water level forcing is taken from Copernicus gridded Absolute Dynamic Topography (ADT) satellite altimetry. River heads are forced using 7446 inlet points from the NOAA's National Water Model (NWM) (Cosgrove et al., 2024) (Figure 1). The model open boundary is tidally forced using the finite element solution FES2014 global ocean tide atlas (Lyard et al., 2021). Our hindcast version only differs from the operational STOFS3D in terms of atmospheric forcing, which was derived from the fifth generation ECMWF atmospheric reanalysis (ERA5) (Hersbach et al., 2020), and in terms of three-dimensional open boundary forcing based on the data-assimilative HYbrid Coordinate Ocean Model (HYCOM) (Chassignet et al., 2007). All hindcast simulations in this paper were hotstarted using the HYCOM 3D fields. A detailed description of the STOFS3D-Atlantic setup adopted in this study is provided by Cui et al. (2024). Note that the operational version of 'STOFS3D-Atlantic' differs from the hindcast version shown in this paper in that the former has been continuously run for a few years, with each forecast hotstarted from the previous forecast cycle.

The STOFS3D-Atlantic mesh, in its current version, has a total of 2,973,769 nodes and 5,728,020 elements. Mesh resolution varies from 100 meters near river banks and parts of the coastline to 1,000 meters overland (Figure 2 a, b, c). Horizontal resolutions in the continental shelf vary between 1,000 and 4,000 meters while in the deep-water regions resolutions are mostly within the 5,000 and 6,000 meters range.

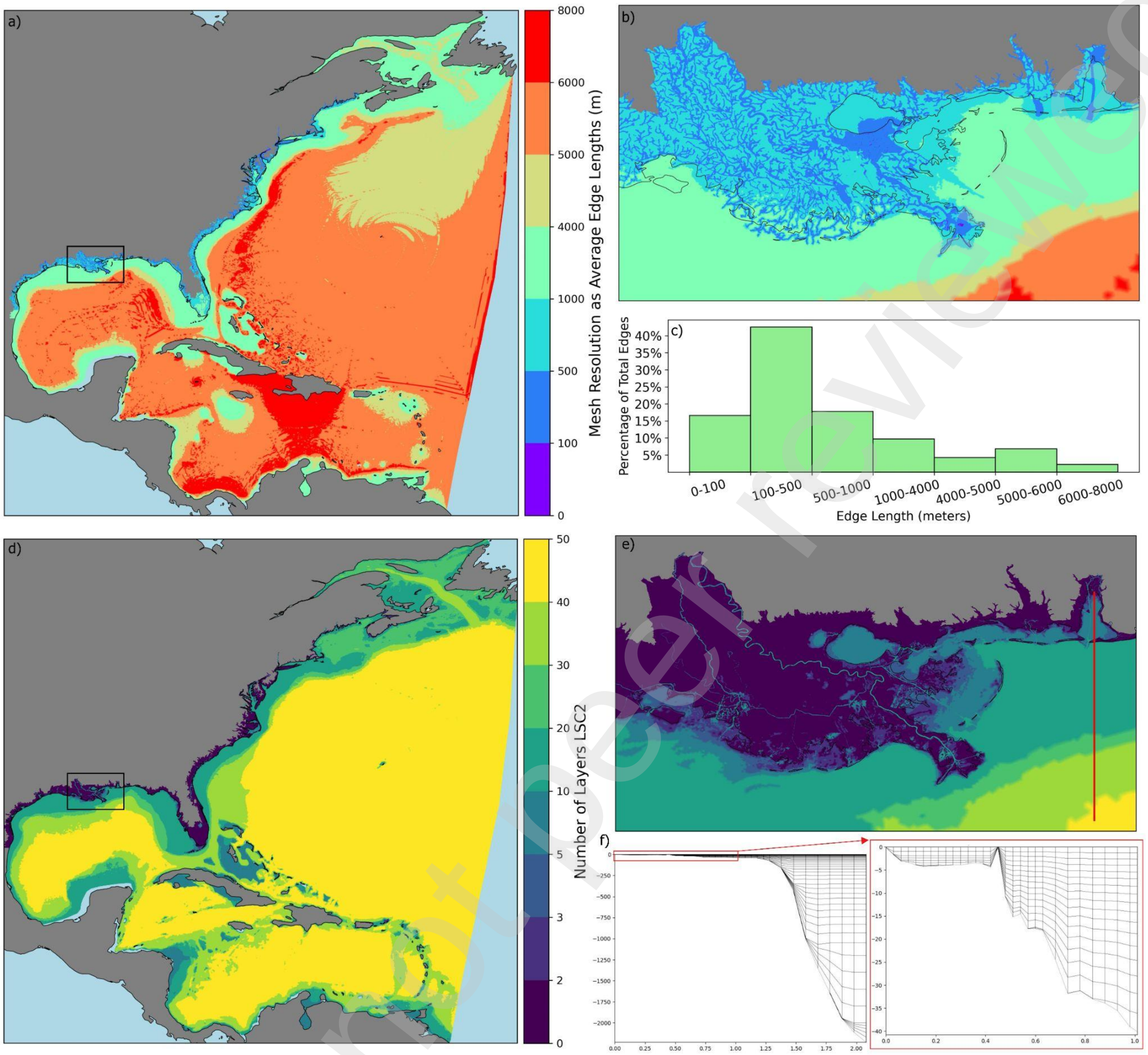

Figure 2. STOFS3D-Atlantic horizontal (a) mesh resolution, where b) is a zoom-in view of the coastal Louisiana and c) is the histogram of mesh resolution distribution; d) is the number of vertical layers in the vertical grid and e) is a zoom-in view of the on coastal Louisiana, f) is the vertical grid configuration along a typical transect shown in (e).

The vertical grid uses the Localized Sigma Coordinates with Shaved Cell (LSC^2) vertical coordinate system for cross-scale applications (Zhang et al., 2015), which allows for each grid node to have its own sigma discretization by treating the mismatch in the number of vertical layers between neighboring nodes as 'shaved' cells at the bottom. As shown in Figure 2 (d, e, and f), the STOFS3D-Atlantic vertical grid has a large number of layers (up to 50) in the deep-ocean areas, and transitions to 2 vertical levels (surface and bottom, i.e., 1 layer) over the floodplain. This discretization allows SCHISM to treat the floodplain in 2D depth-averaged mode, while seamlessly transitioning to 3D configuration over the river and ocean domains. Figure 2f shows the vertical discretization of a transect from upstream Mobile Bay - Alabama to the continental shelf (Figure 2e).

For efficiency, classical first-order Upwind finite volume transport scheme was used for upstream/overland regions, where salinity and temperature gradients are small. In major rivers, estuaries, and the ocean, where 3D processes are important, we used a third-order Weighted Essentially Non-Oscillatory (WENO) for transport as it better preserves baroclinic gradients compared to lower-order schemes (Ye et al., 2019). Bottom friction was set as dependent on the water depth, with coefficients varying from 0.0025 (in estuaries and oceans) to 0.01 (in the watersheds). Relaxation of tracers was applied to cells within 1.5 degrees of the open boundary. Smagorinsky-like filter was applied over pre-defined ocean zones to avoid spurious oscillations. The entire STOFS3D-Atlantic setup was automated and can be found on the SCHISM GitHub repository[3]. For details on the aforementioned parameters, refer to the setup driver script[4] and to Cui et al. (2024).

It is worth noting that the goal of this manuscript is not to improve STOFS3D-Atlantic setup and parameterization but instead to closely replicate it with an OCSMesh-generated horizontal mesh. Therefore, assessing the model setup and calibration is beyond the scope of this work.

2.2. Case-study Timeframe

The model runs include most of the 2021 hurricane season for a period between July 01 to September 30. The first 45 days of simulations were discarded from the analysis to avoid direct influence from the HYCOM hotstart as well as to allow plenty of time for the NWM flows to propagate from the river heads to the river mouths. During the August 15 to September 30 timeframe, six major hurricanes passed through the STOFS3D-Atlantic domain, with three making landfall within the CONUS (Figure 3).

[3] SCHISM GitHub Repo; URL: https://github.com/schism-dev/schism, last access in Mar 2025.
[4] STOFS 3D Atlantic driver script; URL: https://github.com/schism-dev/schism/blob/master/src/Utility/Pre-Processing/STOFS-3D-Atl-shadow-VIMS/Pre_processing/stofs3d_atl_driver.py, last accessed in Mar 2025.

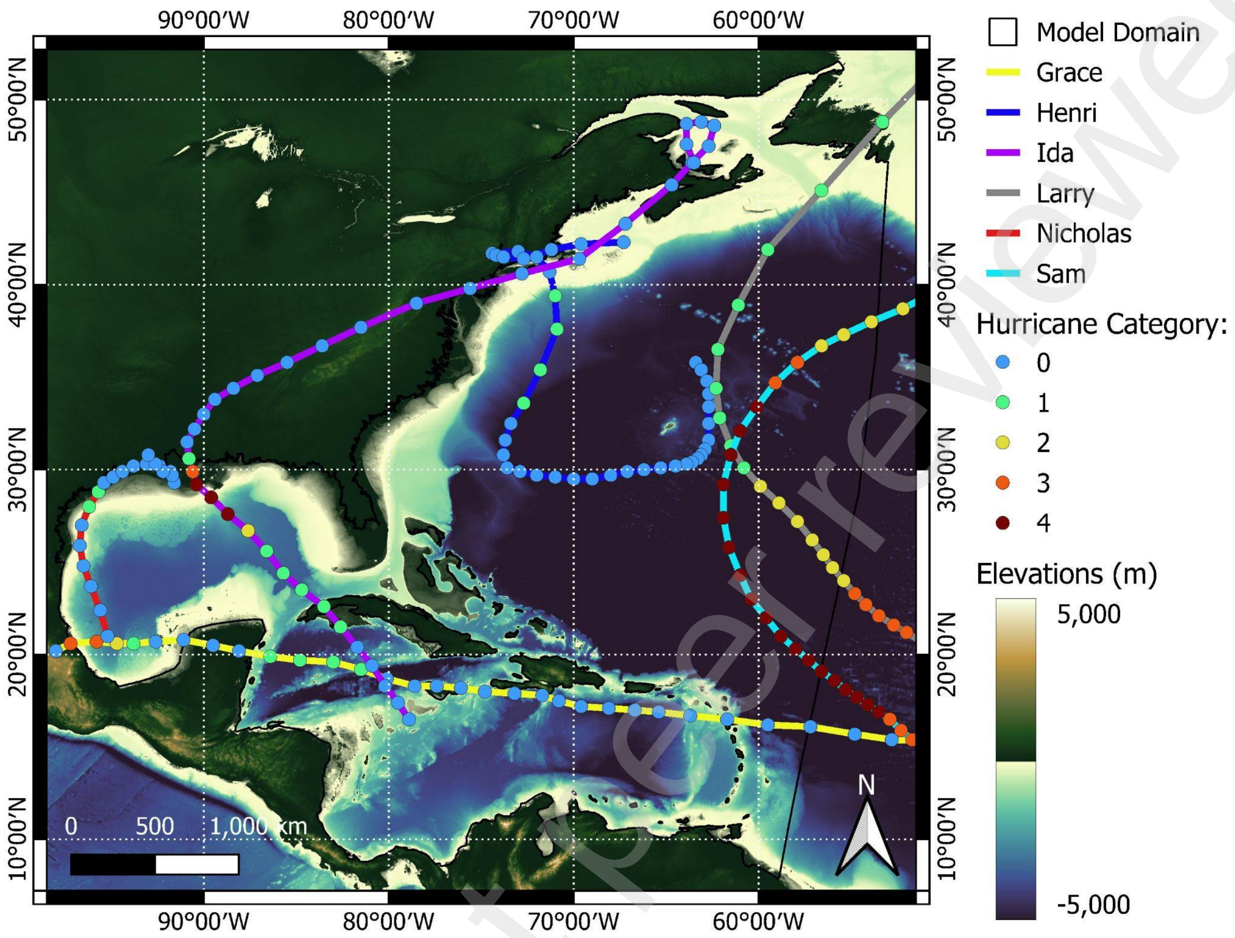

Figure 3. Hurricanes that crossed the model domain between Aug-15 and Sep-30, along with their respective paths and categories (Saffir-Simpson Hurricane Wind Scale).

Hurricane Grace moved across the Caribbean before making landfall in the Yucatan Peninsula as a Category 1 hurricane on August 19, eventually regaining strength over the Gulf and making landfall on the Mexican coast on August 21 as a Category 3 storm (Reinhard et al., 2022). Henri was a Category 1 hurricane that made landfall as a tropical storm on August 22. After making landfall in Rhode Island, Henri followed an unusual track moving west before moving east and dissipating over the Atlantic (Pasch et al., 2022). Hurricane Ida made landfall at Port Fourchon, Louisiana on August 29 as a Category 4 hurricane, causing above ground water levels of more than 4 meters along the east bank of the Mississippi River and delivering more than 250 mm of precipitation over portions of southeastern Louisiana and Mississippi (Beven et al., 2022). Hurricane Larry reached Category 3 right before entering the STOFS3D-Atlantic domain. Larry then passed through east of the Bermuda Islands as a Category 1 hurricane and made landfall in southeastern Newfoundland on September 11 (Brown et al., 2021). Hurricane Nicholas made landfall on September 14 in the Matagorda Peninsula - Texas as a Category 1 hurricane. Although its formation can be traced back to August 28 over the coast of Africa, it was not until September 13, while over the warm waters and unstable atmospheric conditions of the Gulf, that Hurricane Nicholas gained strength. Nicholas is reported to have brought heavy rainfall and caused major flooding across the southeastern United States (Latto and Berg, 2022). Lastly, Hurricane Sam was a high-intensity hurricane that passed eastward of Bermuda before weakening as it moved northeastwards (Pasch and Roberts, 2022). Sam reached its maximum strength over the Atlantic on September 26 (before entering the STOFS3D-Atlantic domain).

3. OCSMesh Design and Application

3.1. Mesh Engines

The NOAA Office of Coast Survey (OCS) mesh generation tool ‘OCSMesh’ is a data-driven, object-oriented, unstructured mesh generation Python tool that uses geometry and sizing function objects to define mesh domain and resolution, primarily targeting SCHISM applications. OCSMesh is mainly designed for DEM-based workflows (although it also supports shape or mesh-based workflows) and is highly flexible to accommodate skewed triangular and quadrilateral elements. OCSMesh relies on the Python wrapper for the JIGSAW library (JIGSAW-Python[5]) as the main meshing engine (Engwirda, 2014). It also utilizes the Triangle[6] mesh engine (Shewchuk, 2002) through some of its utility functions for meshing constrained areas (Section 3.4).

The JIGSAW engine performs mesh generation based on a hybrid solution of conventional Delaunay triangulation. The algorithm is based on the Frontal-Delaunay paradigm, being more theoretically robust than advancing-front techniques and outperforming Delaunay-refinement techniques (Engwirda, 2014). This unstructured mesh generation and tessellation library can be applied to planar, surface, and volumetric triangulation domains. OCSMesh relies on the JIGSAW-

[5] JIGSAW-Python Library; URL: https://github.com/schism-dev/schism, last accessed in Mar 2025.
[6] python wrapper around Jonathan Richard Shewchuk's two-dimensional quality mesh generator; URL: https://github.com/drufat/triangle, last accessed in Mar 2025

Python package for the C++ JIGSAW scripts to perform mesh construction for 2-dimensional geometry and user-defined mesh-size constraints. Triangle is a light weight C-based package used for exact Delaunay triangulations, constrained Delaunay triangulations, conforming Delaunay triangulations, and Voronoi diagrams (Shewchuk, 2002). Triangle mesh generation is driven by command line switches that control several aspects of the mesh generation, including minimum internal element angle, maximum triangle area constraint, and the specification of auxiliary mesh points. In OCSMesh, the merging of different meshes, which is essential for creek-to-ocean mesh generation, is entirely reliant on Triangle (see Section 3.4).

3.2. Software Design

The built-in *Geom* and *Hfun* objects are responsible for pre-mesh generation processing. *Geom* is a singleton object representing the mesh domain that can be created via input raster, shape, or pre-existing mesh files. The *Raster* class defines objects used for handling DEM files (e.g., GeoTIFF and NetCDF); it also wraps several RasterIO functionalities, including: clipping, reprojecting and CRS handling, windowing, contour extractions, etc. The Raster-based *Geom* object uses the *Raster* class to define the mesh domain by enclosing polygons created from specified elevation contour boundaries extracted from DEM files. Other geometry types supported by OCSMesh are: *Mesh*-based for defining the domain based on a pre-existing mesh, *Polygon*-based for manually defined domains, and *Collector* type for a combination of the aforementioned geometries. The *Collector* type is the most convenient *Geom* for SCHISM applications as it allows for the ingestion of a list of raster files, where the list's order corresponds to the order of priority in which raster files are considered, thus allowing for a more faithful representation of the topography and bathymetry.

The *Hfun* (size function) object defines the element size through several size specifications methods, such as: global maximum and minimum, based on DEM slopes, and based on a detected or specified region or the distance from it. These size specification methods can be combined within the *Hfun* object for highly customizable mesh resolutions and transitions. Like the *Geom* object, the *Hfun* object can be defined based on rasters, mesh, or a combination of the two (*Collector* type). Unlike the common approach used in other meshing tools, OCSMesh optimizes the size function generation by addressing it tile by tile (raster and/or mesh), where: i) a mesh is created for each tile according to the multiple size specifications, ii) the element size values of this mesh are interpolated back onto the mesh, and iii) all these tile-specific meshes are combined according to their order of priority. This unstructured *Hfun* is more memory efficient and flexible than matrix-based approaches (Mani et al., 2021) used by most mesh generation software.

Finally, the JIGSAW array representation (*msh_t*) of the *Geom* and *Hfun* objects are generated and passed to the mesh engine. Mesh post-processing in OCSMesh includes fixing common mesh generation problems such as node duplicates, isolated and pinched nodes, small elements, unconnected patches, and folded-boundaries. In addition, the interpolation of topobathy elevations into the mesh and the automatic detection of the mesh boundaries (land, open, and islands) can be carried out as part of the post-processing. Mesh boundaries can be defined either as a function of the elevation of the boundary nodes (i.e., elevation < 0 = Ocean Boundary), or based on its location in relation to predefined polygons. For more information readers are encouraged to refer to the OCSMesh GitHub repository (see Software and data availability) and official NOAA Technical Memorandum (Mani et al., 2021).

3.3. SCHISM-based Mesh Generation Criteria

OCSMesh achieves seamless land-ocean mesh generation for SCHISM-based applications by: i) enforcing customizable subdomain-specific mesh generation requirements, and ii) integrating the resulting subdomain meshes into a domain-wide mesh (pre-generated ‘ocean’ mesh). For compound flooding, the targeted subdomains are the floodplain and rivers. SCHISM-based mesh generation criteria for these subdomains are summarized as follows.

3.3.1. Floodplain

Heuristically when developing a SCHISM mesh for the floodplain, the dry watershed boundary must always remain dry. In other words, the upper limit of the floodplain mesh domain must be set at an elevation above the probable reach of extreme flood levels. This prevents the flood waves from reaching an unrealistic ‘wall’ at the boundary of the mesh, and thus being reflected back and interfering with the incoming flood. Moreover, higher resolutions should be used near the coastline and river banks for refined wetting and drying representations and smooth flow conveyance and routing.

3.3.2. River

River systems can be numerically represented during the delineation of the floodplain mesh by either passing the river network as high-resolution arcs to the *Hfun* object or as a separate system composed of basic river features, such as river bank and thalwegs. The former does not ensure channel connectivity while potentially resulting in an unnecessarily large number of elements, and therefore should be avoided in compound flooding applications. The latter demands a more detailed representation of the river arcs that captures essential riverine features, in addition to complementary inner and outer arcs (see Ye et al., 2023).

Meshes for the riverine subdomain should take advantage of SCHISM's polymorphism and treat the river system as a combination of quadrangular-like elements (for well-defined channels) and triangular elements (for river junctions and other complex areas), as well as seamlessly migrate between 3D, 2D (horizontal and vertical), and quasi-1D configurations (Ye et al., 2020). According to Zhang et al. (2024), quadrangular elements can be used to: i) more accurately represent flow due to its flow-aligned segments (Holleman et al., 2013); ii) improve the representation of 3D processes as a result of its higher and easy-to-control cross-channel resolution; and iii) reduce mesh element count. At the same time, triangular elements allow for greater flexibility, which is essential when representing channel crossings and complex riverine structures at river junctions.

3.4. Workflow

In order to automate the generation of continuous compound flooding meshes for SCHISM, we developed a workflow that separately fulfills the aforementioned criteria for each mesh subdomain. The horizontal mesh generation and merge with the pre-generated 'ocean' mesh is performed entirely within OCSMesh, while prior river delineation and model deployment are performed by other SCHISM-oriented packages (Ye et al., 2023; Calzada et al., 2023). The workflow is shown in Figure 4 and later tested in Sections 4 and 5. It should be mentioned that although the entire workflow is being considered, we will focus our discussions on the segments pertaining to OCSMesh as the evaluations of the complementary packages (Figure 4a and 4e) is beyond the scope of this study.

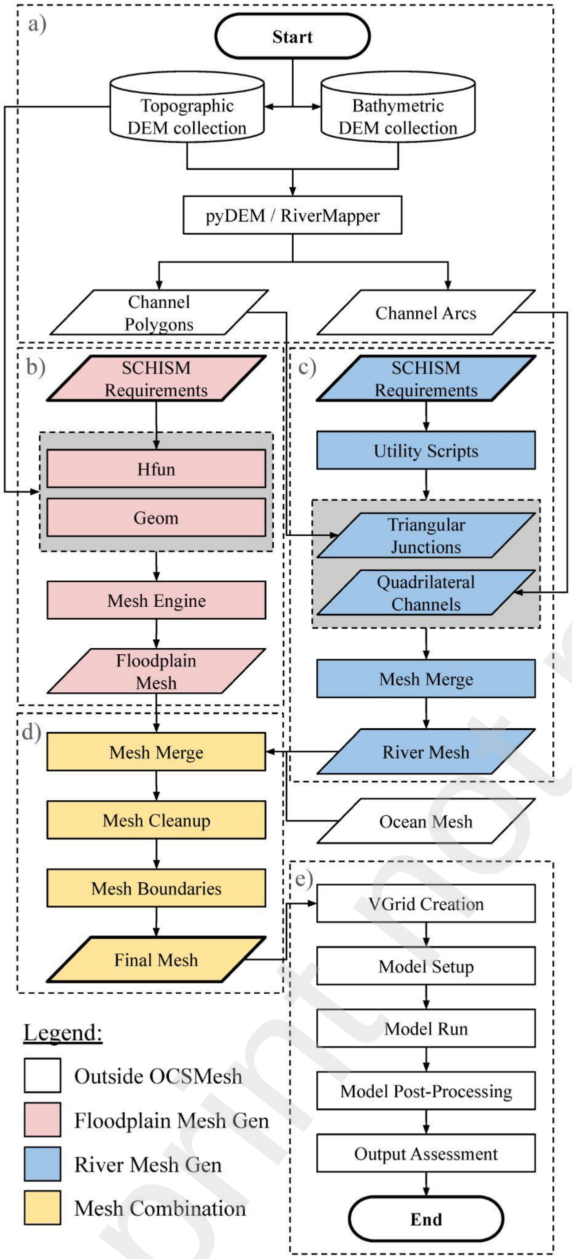

Figure 4. Workflow for deployable compound flood meshes using OCSMesh. Parts 'a' and 'e' are outside OCSMesh and are handled by other SCHISM-oriented packages. Part 'b' illustrates the floodplain mesh generation which relies on JIGSAW, and Part 'c' shows the river mesh generation that uses polygons that represent major river features and are triangulated using Triangle. Part 'd' represents the mesh merging processes which also relies on Triangle.

In addition to the topographic and bathymetric DEM datasets, the workflow also requires river channel data (Figure 4a), which can be manually delineated or automatically extracted. Here we use RiverMeshTools (Ye et al., 2023), a parallel Python-based tool that accurately detects river thalwegs from DEMs (pyDEM) to generate river data for mesh generation (RiverMapper). RiverMapper can output either arcs (line-strings with equal number of vertices along the channel, each indicating features such as channel banks, center, etc. spaced equally across each channel section) or polygons (cleaned up arcs that ensure the representation of main river features). Both RiverMapper outputs can be used as inputs to OCSMesh, with the arcs used for quadrangular elements and the polygons for triangular elements.

As mentioned before, the floodplain mesh should encompass the entire floodplain so that during storm surge events the flood does not reach the edges (dry boundary) of the mesh. In our case studies we adopted the +10 meters contour line as the standard cut-off value for the floodplain extent (*Geom* object). The wet-boundary of the floodplain mesh was extended over the ocean domain (down to the -50 meters contour line) to ensure overlap between subdomains generated by OCSMesh and the ocean side mesh (assumed an input in the scope of this paper) during the mesh merging process (Figure 4d). Although customized refinements based on the topography, contour lines, constant resolution between contour lines, feature (line) or polygon, as well as feature channels (add_subtidal_flowlimiter, add_contour, add_constant_value, add_feature, add_subtidal_flowlimiter, respectively) can be applied for generating floodplain mesh (see Mani et al., 2021), we passed a constant global minimum resolution of 500 meters to the *Hfun*, at the expense of increased final mesh size. This demonstrates that even very large meshes can be handled by the mesh generator and easily integrated by our creek-to-ocean mesh generation workflow.

Going back to the river subdomain, OCSMesh utility scripts can be used to quadrangulate RiverMapper's arcs and identify overlaps between the individual river reaches (Figure 4c). These overlap areas are treated as river junctions and then triangulated using the Triangle mesh engine. To guarantee that the junction attributes identified by RiverMapper are represented in the mesh, RiverMapper's polygon coordinates are passed to the Triangle mesh engine as auxiliary points. Connectivity is ensured by directly triangulating between the boundary nodes of the quadrangular mesh (channels) and the boundary nodes of the triangular mesh (junctions). To replicate the

STOFS3D mesh, which only has (elongated) triangular elements in the river channels, a simpler approach is adopted where only RiverMapper polygons are used (both for channels and junctions) to generate a triangular river mesh. Note that this triangle-only approach is still much more efficient than the size function approach, which results in excessively large meshes due to prioritization of mesh 'quality' while still lacking cross-channel resolution, especially in very narrow rivers.

The floodplain and river meshes are then merged to a pre-generated ocean mesh (Figure 4d) using OCSMesh utility functions for merging overlapping meshes (utils.merge_overlapping_meshes), followed by a mesh cleanup (to eliminate node duplication, isolated and pinched nodes, infinitesimally small area elements, and isolated small patches). Lastly, mesh boundaries (land, open, and islands) are defined using the OCSMesh boundary extraction functions.

Figure 5 illustrates the OCSMesh merging process, where the floodplain mesh (Figure 5a) is generated based on DEMs using the *Geom* and *Hfun* objects that are passed to the JIGSAW engine. The river mesh (Figure 5b) is triangulated from the RiverMapper polygons using the Triangle engine. These two meshes are passed to the merge_overlapping_meshes utility function (Figure 5c). The function then carves out the floodplain based on the river mesh (buffered by some layers of adjacent elements - see yellow areas in Figure 5c), and remeshes the constrained gap using the Triangle mesh engine. Note that the floodplain nodes that fall within the gap are passed to the Triangle engine as auxiliary points, thus preserving the original floodplain triangulation.

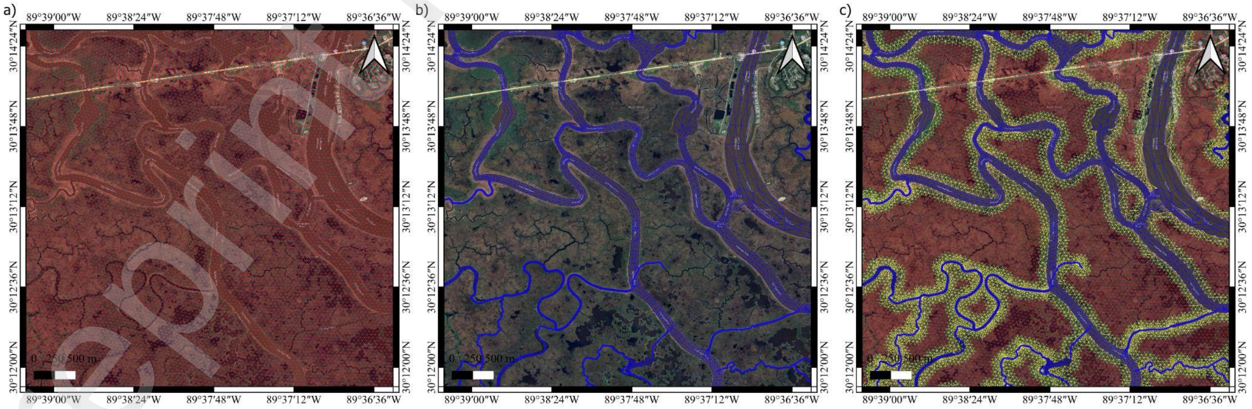

Figure 5. Generation of a continuous floodplain + river mesh, where a) depicts the *Geom* and *Hfun* - based floodplain mesh, b) depicts the RiverMapper-based river mesh, and c) depicts the final merged product (note the buffer zones around river banks).

The last section of the workflow (Figure 4e) is carried out by PySCHISM (Calzada et al., 2023), a Python-based package for preparing inputs for 3D baroclinic SCHISM runs, including open boundary forcing, riverine input, and atmospheric forcing. A description of the automated STOFS3D-based model setup and its parameters is provided in Section 2.1.

4. Example Application 1: Automating the STOFS3D-Atlantic Mesh Generation with OCSMesh

4.1. Model Pre-Processing

In order to make our results reproducible we wrote a step-by-step Jupyter Notebook tutorial[7] covering the floodplain and river mesh generation and merge process following the design criteria of Section 3.3 and the workflow of Section 3.4 for the entire STOFS3D domain. This tutorial is publicly available on the OCSMesh GitHub and can be easily adapted for specific use cases. Assuming that many OCSMesh users may not have access to High Performance Computing systems or cloud computing, we ran the tutorial on a consumer-grade laptop with a 12th Gen Intel(R) Core (TM) i9 processor and 32GB of installed RAM.

To demonstrate the versatility of OCSMesh, even for large scale and high-resolution cases such as the one proposed here, we recorded the time and memory usage during the mesh generation process. The first step (Figure 4b), which consists of the generation of the floodplain mesh, took less than 1 hour to complete, resulting in a 1.1 million-node mesh. The triangulation of the RiverMapper polygons for the entire East Coast and the Gulf rivers took about 15 minutes and resulted in a 1.4 million-node mesh. Merging the river mesh into the floodplain took close to 40 minutes and resulted in a mesh with 3.2 million nodes. Finally, this mesh was merged to the 1.4-million node ocean mesh, taking nearly 3 hours to complete (most memory-intensive step). The total meshing process is therefore close to 5 hours.

[7] Example Application 1 Tutorial: URL: https://github.com/noaa-ocs-modeling/OCSMesh/blob/main/Tutorials/ApplicationExample01.ipynb, last accessed in Apr 2025.

The resultant horizontal mesh has a significantly larger number of nodes (over 5 million) in comparison to the almost 3 million-node STOFS3D grid; as in the STOFS3D mesh, most nodes are located on the floodplain and the riverine domains (Figure 6 a, b, c).

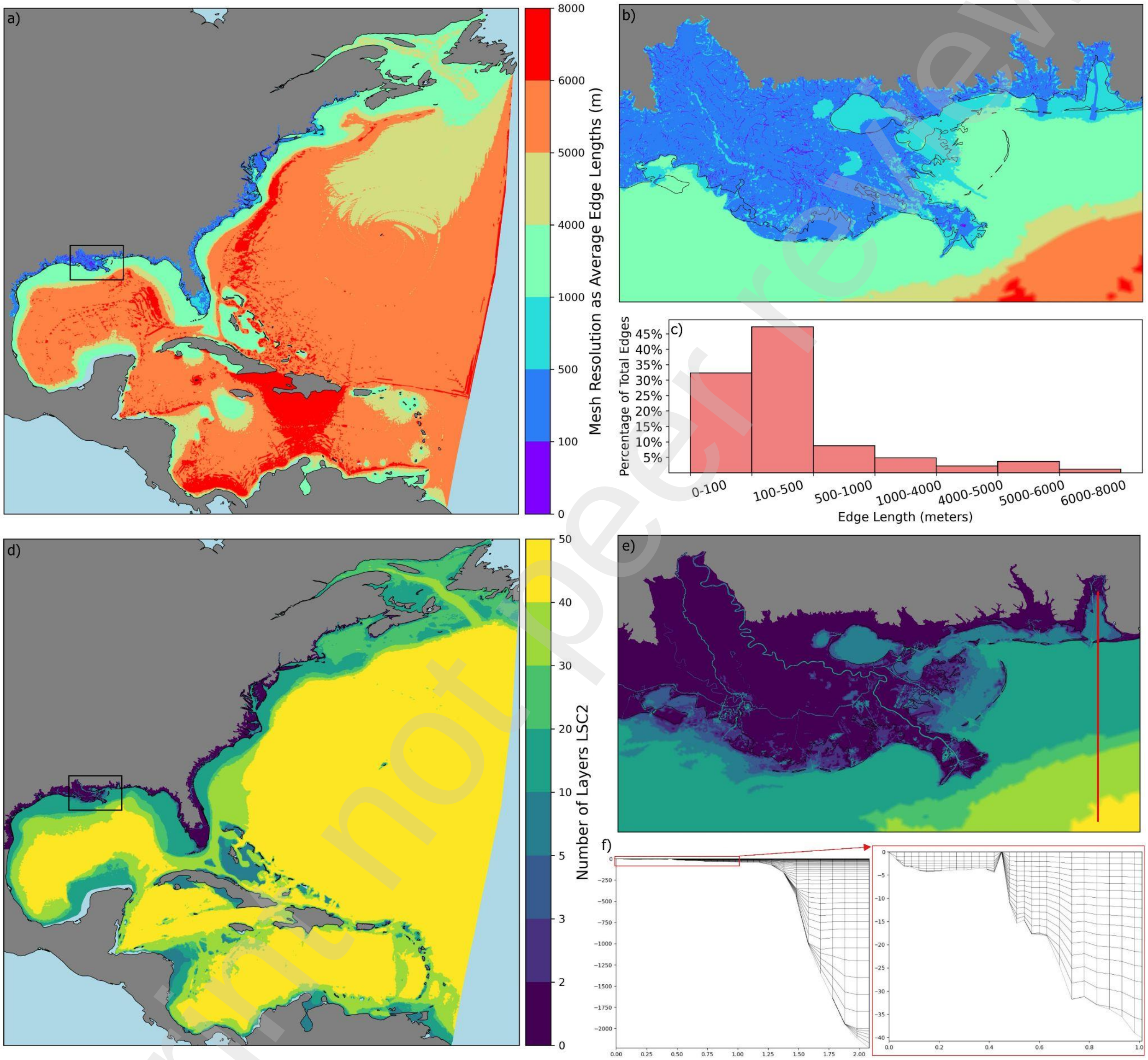

Figure 6. OCSMesh a) horizontal mesh resolution, where b) is a zoom-in view of the on coastal Louisiana and c) is the histogram of mesh resolution distribution; d) shows the number of vertical layers used; e) is a zoom-in view of the on coastal Louisiana, f) is the vertical grid configuration along a typical transect shown in (e).

Note that the automated mesh's large size is not solely due to the higher resolution of the floodplain (500 meters) but also due to the addition of a large number of elements during the floodplain and river mesh merging process (Figure 5c). For instance, the merged floodplain and river mesh has a substantially larger number of nodes (3.2 million) than the original STOFS3D's floodplain (1.1 million nodes) and river (1.4 million nodes) meshes combined. This increase in mesh size during merging is due to OCSMesh standard configuration, where all nodes from the low priority mesh (floodplain) are preserved while ensuring that the new triangles have an internal angle larger than 30 degrees. These criteria can be relaxed if computational cost is a concern. However, excessively small internal angles will create sharp transitions that may adversely affect wetting and drying over the floodplain (Figure 7).

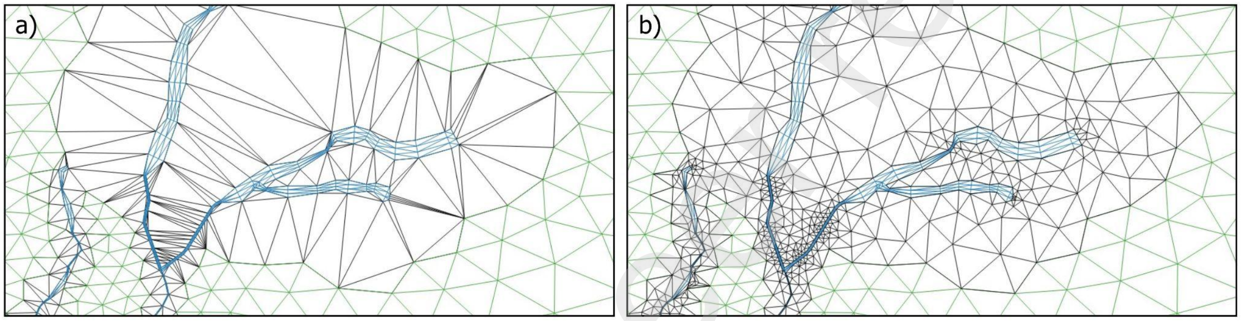

Figure 7. Mesh (black) created by merging the river mesh (blue) into the floodplain mesh (green) without restrictions (a) and with standard OCSMesh constraint for internal angles (b).

In Figure 7 we show two merging criteria; an excessively relaxed criteria (Figure 7a), where the transition (in black) does not preserve the original floodplain mesh nodes as no auxiliary points and no minimum internal angle are enforced, versus the standard criteria where a minimum 30-degree angle is enforced while also keeping the original floodplain mesh nodes (Figure 7b). Similarly, the merging of the 3.2-million node floodplain+river mesh with the pre-existing 1.4-million node ocean mesh also results in a final mesh that is larger (5 million nodes) than the combined node count of the input meshes.

The same vertical grid configuration was used for both manually-created STOFS3D mesh, and automated OCSMesh mesh. The OCSMesh mesh-based model was successfully deployed with these horizontal and vertical discretizations using the automated scripts for STOFS3D deployment (stofs3d_atl_driver.py on the SCHISM GitHub repository[8]).

[8] STOFS 3D Atlantic driver script; URL: https://github.com/schism-dev/schism/blob/master/src/Utility/Pre-Processing/STOFS-3D-Atl-shadow-VIMS/Pre_processing/stofs3d_atl_driver.py, last accessed in Mar 2025.

4.2. Model Evaluation

Simulations for both models, STOFS3D (3 million nodes) and OCSMesh-based model (5 million nodes), were carried out in a high-performance computing environment using fifty compute nodes for a total of four thousand cores. The model had to be restarted throughout the simulation as it exceeded the wall-clock time limit of 8 hours. To complete the 99-day simulation over nineteen hours for the STOFS3D simulations and forty hours for the OCSMesh-based simulations were needed.

4.2.1. Qualitative Assessment

The last of the three *in silico* oceanography guiding principles described by Zhang et al. (2024) stresses that model assessment should focus on processes. Thus, qualitative skill assessment cannot be substituted by quantitative error metrics, which are often misleading, easily manipulated, and do not necessarily assess the underlying processes. One of the most well documented issues encountered during the development of a cross-scale 3D SCHISM model is the proper delineation and meshing (horizontal and vertical) of different regimes (or zones, such as deep/open ocean or eddying regime, nearshore non-eddying regime, and near steep slopes in the transitional regime). More specifically, meshes for the eddying regime should not include abrupt changes in resolution as it may distort eddying processes (Danilov and Wang, 2015). Still, both eddying and transitional regimes must follow the hydrostatic assumption (horizontal scale >> vertical scale), as its violation may cause spurious cold-water upwelling (Zhang et al., 2016). Even though our automated mesh only covers the non-eddying and overland areas, the merging with the manually created ocean mesh may introduce errors that should be investigated. Figure 8 presents a sea-surface temperature comparison between the STOFS3D and OCSMesh-based models against level-3 satellite data from Low Earth Orbit (LS3-LEO) (Jonasson et al. 2022).

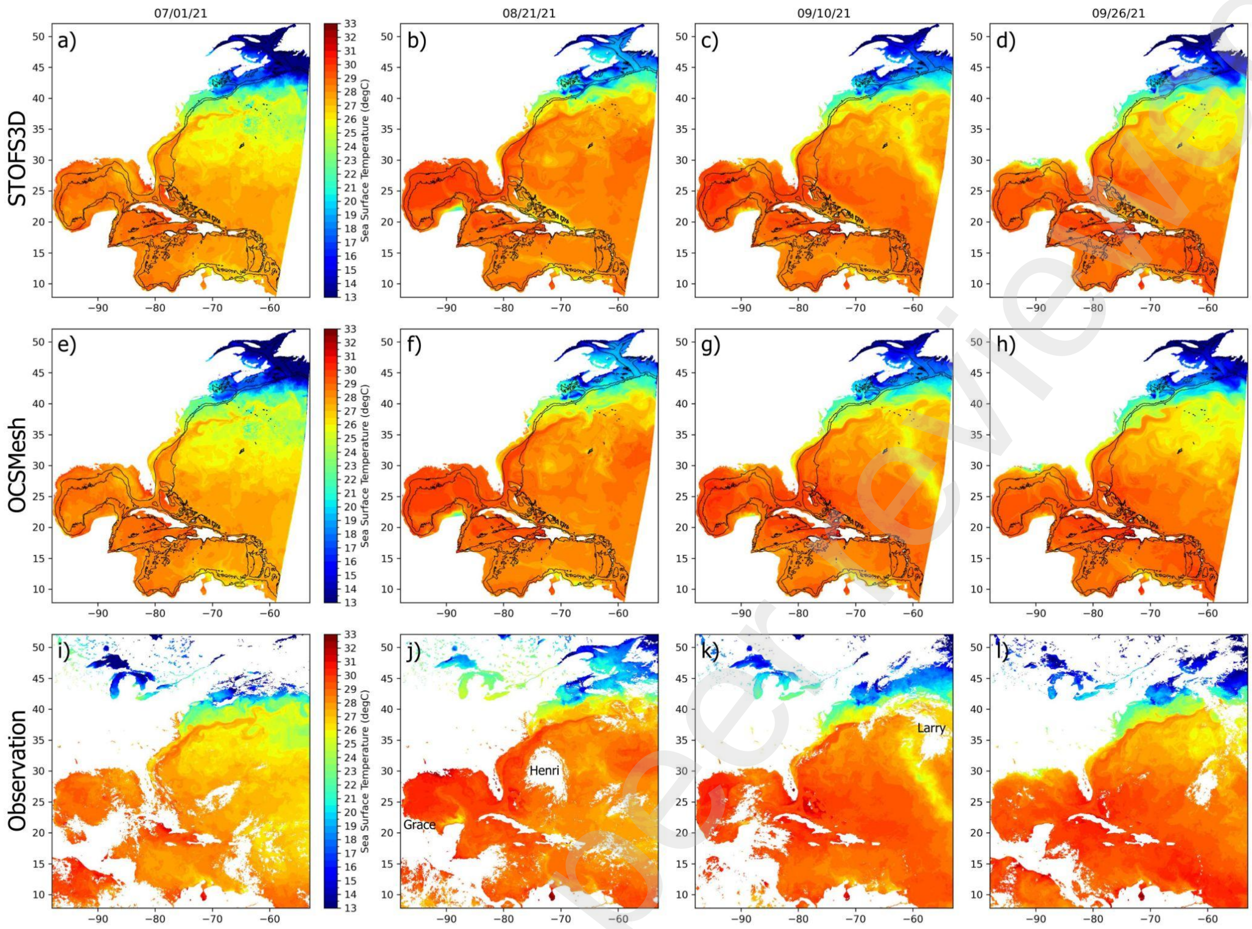

Figure 8. STOFS3D (first row), OCSMesh-based model (second row), and level-3 satellite sea-surface data (third row) comparison for the model hotstart (first column), day 52 (second column), day 72 (third column), and day 88 (fourth column).

Figures 8a, 8e, and 8i show the STOFS3D, OCSMesh, and LS3-LEO data at the time of the model hotstart using HYCOM. As expected, the STOFS3D and OCSMesh-based model hotstarts are nearly identical (as both use the same ocean mesh) and strongly resemble the observations (Figure 8i). As mentioned in Section 2.2, the first 45 days of simulations were discarded to minimize the influence of the hotstart. Figures 8b, 8f, and 8j show the sea-surface temperature at day 52, which coincides with Hurricane Grace's landfall over the Mexican Veracruz coast and the transition of Henri from a tropical storm to a Category 1 hurricane (Figure 8j). As seen in Figure 8j, Hurricane Grace caused the emergence of a significantly colder water wake along its track as it crossed the northern Yucatan Peninsula and moved westward. Both models (Figures 8b and 8f) captured this behavior, although with colder temperatures. The models also captured the upwelling from the intensification of Hurricane Henri over the US east coast (Figures 8b and 8f). On day 72, both models (Figures 8c and 8g) show a clear trail of cold water upwelling along Hurricane Larry's track, which is similar to the observations (Figures 8k). Last, we highlight day 88 (Figures 8d, 8h, and 8l), which comes after a long period of heavy rainfall over the Mississippi Basin in the aftermath of Hurricanes Ida and Nicholas. Both models' sea-surface temperatures show larger and colder river plumes (Figures 8d and 8h) than the observations (Figures 8l), attributed to either an underestimation of the river boundary temperature or an overestimation of the riverine flood. In Section 5 a high resolution, relocatable example is presented and followed by a deeper discussion on the hydrological implications of estimating compound floods without coupling to a hydrological model.

Elsewhere, the models were able to capture the general sea-temperature features, with a slight underestimation of temperatures (1 degree Celsius or less) between latitudes of 15 and 30 degrees north. We also see slightly colder temperatures along steep bathymetries, such as along the Venezuelan coast, the northern Yucatan Peninsula, northeastern Florida, and the Bahamas. It is worthwhile to mention that these slightly colder waters along the continental coast are not the spurious cold-water upwelling mentioned in the previous paragraph, as similar patterns are also observed in the satellite imagery (Figures 8j, 8k, and 8l). It can also be noted that the OCSMesh-based model produced warmer sea-surface temperatures around the Bahamas (Figures 8b, 8c, and 8d) when compared to STOFS3D (Figures 8f, 8g, and 8h). That particular behavior, although closer to the observations, seemed counterintuitive since both models use the same mesh for the ocean domain. The most likely explanation is that, as part of the mesh merging process, OCSMesh fills gaps between meshes using a user-defined maximum area threshold. The threshold selected was high enough to also fill the smaller islands within the Bahamas and Turks and Caicos, which effectively expanded the non-eddy regime meshing areas. Meshing around steep slopes requires special care as documented in the SCHISM Manual (see Meshing for cross-scale regimes chapter[9]); even small changes in the horizontal and vertical discretization can lead to major changes in cold water upwelling because SCHISM does not apply any bathymetry smoothing. This type of qualitative assessment of the simulation results is crucial for identifying features introduced

[9] SCHISM Manual - Meshing for cross-scale regimes: URL: https://schism-dev.github.io/schism/master/mesh-generation/cross_regime.html, last accessed Feb 2025.

during the automated mesh generation process that otherwise would not be captured using standard quantitative skill metrics.

The main concern over the floodplain+river domain is that the flow is properly conveyed downstream. Flow impediments, especially in the cross-channel direction, result in the underestimation of flow downstream and overestimation of lateral inundation upstream, having a direct impact on the representation of the compound nature of coastal floods (Ye et al., 2023). Figure 9 shows both models wetting and drying from the beginning of the simulation (July 01) to August 15 (when we began assessing the model outputs) for one of the longest riverine systems in the STOFS3D domain (almost 100 km in the Pascagoula River - MS).

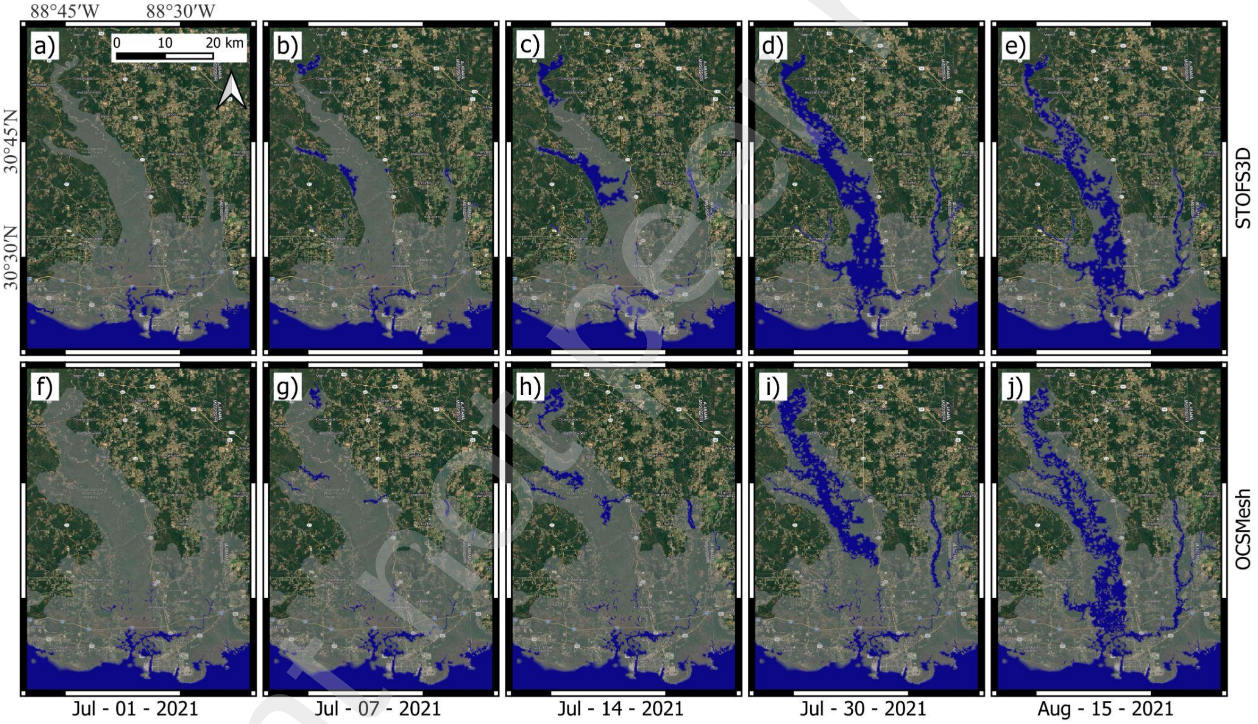

Figure 9. STOFS 3D (a, b, c, d, e) and OCSMesh (f, g, h, i, j) wetting (blue) and drying (transparent gray) for five timesteps within the 'warm up' period: Jul - 01 (a and f), Jul - 7 (b and g), Jul - 14 (c and h), Jul - 30 (d and i), and Aug - 15 (e and j).

By August 15, both models had propagated the flows from the upstream NWM all the way to the mouth of the river. It should be mentioned that the operational mesh uses an older version of RiverMapper that was triangulated using a graphical user interface software. Thus, flood conveyance is expected from the operational mesh. The STOFS3D wetting and drying shows a larger inundated area when compared to the OCSMesh-based model. That cannot necessarily be attributed to flow impediments as RiverMapper (which was used for the STOFS3D mesh) ensures flood conveyance. Instead, that may be a consequence of the coarser floodplain resolution that leads to inadequate wetting and drying. Our results also show that long ‘warm-up’ periods are necessary for compound flood studies to ensure an initial state where all rivers are properly ‘filled up’.

4.2.2. Quantitative Model-to-Model Comparison

Here we use standard metrics (i.e., correlation coefficient, standard deviation, and root mean square error) to evaluate model reproducibility. Our goal is not to evaluate STOFS3D performance, but instead to assess our ability to reproduce the operational model outputs using automated meshing. For that reason, we focused our quantitative assessment on model-to-model comparison. Outcomes from the model skill assessment (against observations) were left to a separate publication that is currently under development. This quantitative comparison was carried out with a beta version of the NOAA Shared-Cyber Infrastructure and Skill Assessment package[10]. The location of all the available NOAA CO-OPS and NDBC, and USGS stations were used and the result is summarized in Figure 10.

[10] NOAA Shared Cyber-Infrastructure Package; URL: https://github.com/NOAA-CO-OPS/Next-Gen-NOS-OFS-Skill-Assessment, last accessed in Apr 2025.

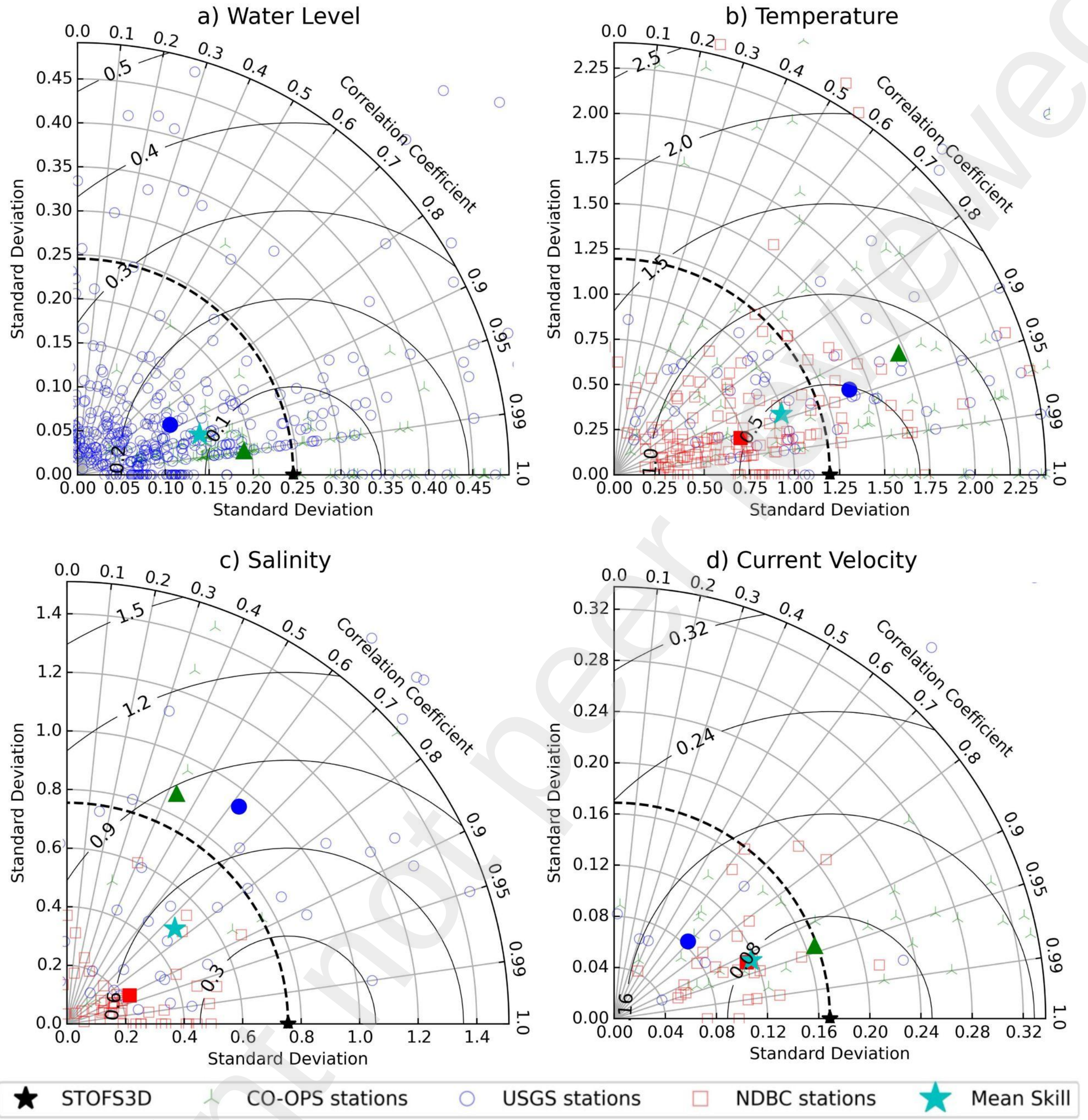

Figure 10. Taylor diagrams for a) water levels, b) temperature, c) salinity, and d) currents. The bold marks represent the median values for the location of the different data sources (CO-OPS, USGS, or NDBC). The cyan star represents the OCSMesh-based model average from all stations. The black star represents perfect skill (i.e., perfect match with STOFS3D). Contour lines represent root mean square error.

Our model produces water level estimates at downstream stations (most CO-OPS station locations) that are very close (correlation coefficient > .99, RMSE close to 5 cm) to STOFS3D (Figure 10a). The modeled water level at upstream stations (e.g., many of the USGS gauges), however, is more sensitive to the skill of the NWM river forcing. Although major improvements have been made since its release in 2016, the NWM (in its 2.1 version) still presents absolute bias > 20% and correlation coefficient < 0.8 to almost half of the 10,000 USGS gauges over the CONUS (Cosgrove et al., 2024). Even though both models (STOFS3D and OCSMesh-based) were deployed using the same model setup scripts, the automated placing of NWM river heads is mesh-dependent as different river discretization may produce different feeder channels (see Ye et al., 2023 for details). Compounding the issue is the river mesh itself. The different river discretizations between the two models, despite being able to propagate water downstream (Figure 9), may not reflect the detailed river network meanders or capture very small creeks where upstream gauges (mostly USGS) are located. It should also be mentioned that several riverine stations within large low-lying areas, such as southern Florida and the Delmarva Peninsula, were marked as ‘no-signal’ (no flood signal captured in the timeseries) and were not considered for this skill assessment.

These factors suggest that the automatic placing of NWM river heads using the STOFS3D-based model setup or the delineation of channels may not have yielded the most consistent results. On the other hand, manually setting the location of more than 1,000 USGS-based model stations over the STOFS3D and OCSMesh-based meshes would be extremely laborious and counterproductive when the goal is automation. An alternative is explored in Section 5 where we increased the level of detail of the river representation on pyDEM/RiverMapper before exporting into OCSMesh. Last, determining flow direction and flow accumulation, especially in flat areas, is a well-documented challenge in watershed analysis due to the presence of problematic parallel flow lines (Zhang et al., 2017). As described in Section 3.4, our workflow uses the pyDEM package (Ye et al., 2023) for thalweg detection, which in turn relies on the traditional D8 flow algorithm (O’Callaghan and Mark, 1984). The D8 algorithm assigns flow direction from each cell towards the single neighboring cell with the highest elevation gradient. In order to better capture the dispersive nature of flow direction in such areas, multi-flow direction, moderately or fully dispersive methods (e.g., D∞, D∞-LTD, MD∞, and MD8) might be preferred as they were designed to produce a more realistic representation of overland flow (Orlandini et al., 2012).

Our model also reproduces STOFS3D temperature with high fidelity (correlation coefficient ~ 0.95 and RMSE < 0.5 degree Celsius) (Figure 10b); thus, corroborating the similar temperature patterns shown in Figure 8. The OCSMesh-based salinity shows a similar performance to STOFS3D as well. However, salinity results are less consistent for different data sources (Figure

10c). The closest salinity model-to-model match was obtained for NDBC buoys (RMSE< 0.6 PSU and correlation coefficient > 0.9), which are mostly located in the open ocean where both models use the same 'ocean mesh' and the NWM inputs exert little to no influence. Note that the same behavior is not observed for temperature. In this case both open ocean NDBC buoys and upstream USGS gauges show similar metrics (RMSE ~ 0.5 degree Celsius and correlation coefficient ~ 0.95). Water temperatures, especially in these very shallow waters (i.e., upstream stations), are highly influenced by air-sea interactions, which are based on the same inputs for both models. Therefore, higher temperature model resemblance, when compared to salinity, is expected despite the different river discretization and river head placing, as temperature estimates are more rapidly 'rectified' by the ambient temperature.

Current velocity evaluation shows greater model similarity at tidal stations (most of the CO-OPS and NDBC), with RMSE < 0.08 meters per second and correlation coefficient > 0.9, while the models tend to diverge at riverine gauges (RMSE ~ 0.12 meters per second and correlation coefficient < 0.7). It should be noted current estimations at these riverine locations (i.e., small channels) are more sensitive to small differences in the cross-section area than those downstream. Therefore, lower model resemblance at USGS station locations is expected as the two models have different discretizations for the river network.

The similarity in results of our reproducible workflow compared to the operational mesh reinforces the claim that defensible modeling, and more specifically automatic mesh generation, is indeed within reach (Zhang et al., 2024), even for large scale cases where local tuning and calibration may still be necessary.

5. Example Application 2: Relocatable High-Resolution Mesh with OCSMesh

5.1. Relocatable Mesh Generation

For this example, suppose we would like to locally boost the resolution of the river network within the same computational domain of the previous example (see Section 4.1). To achieve this, we generate a very detailed river delineation product using RiverMapper and then create a high-resolution river mesh for the Pearl River Basin between Louisiana and Mississippi (223,349 nodes). Here we took advantage of the JIGSAW-based mesh refinements to create a high-resolution floodplain mesh that has 50-meter resolution up to the 1-meter isobath, 100-meter resolution between the 1- and 5-meters isobaths, and 200-meter resolution between the 5 to 10 meters isobaths, with a coarsest resolution of 500 meters. The final floodplain mesh has a total of 160 thousand nodes. These local river and floodplain meshes can each be created in less than 15 minutes and combined in 10 minutes. In this mesh the resolution at river junctions reaches less than 10 meters, and most rivers have resolution of up to 25 meters (Figure 11).

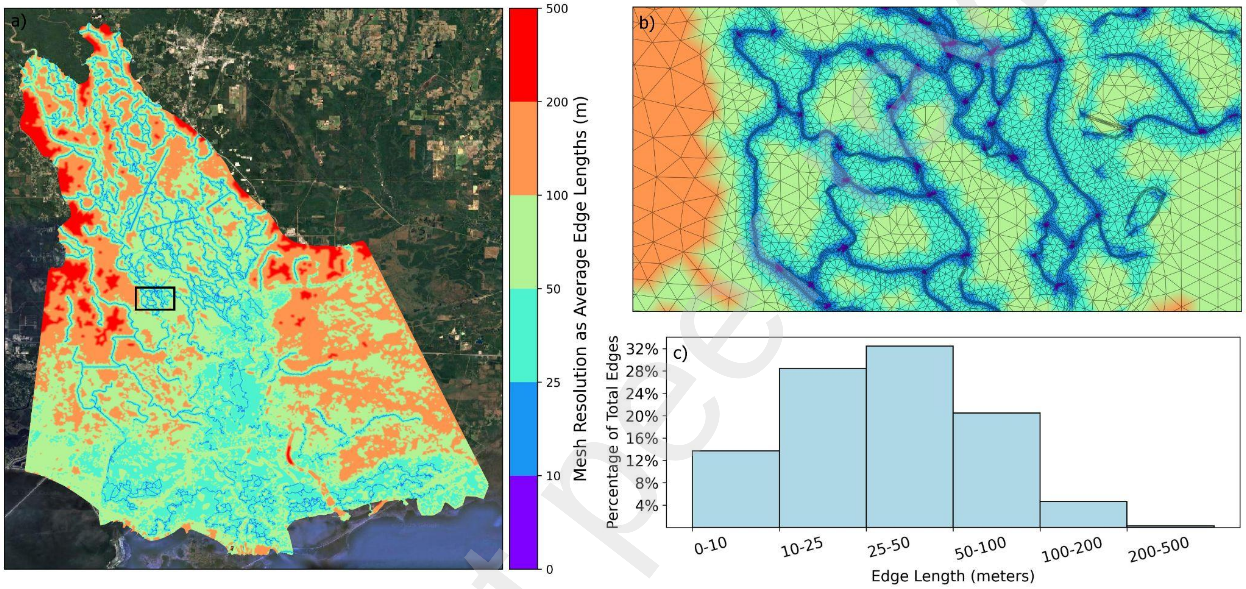

Figure 11. High resolution local mesh to be relocated inside the coarser STOFS3D mesh. Including horizontal (a) mesh resolution, where b) is a zoom-in view of an area with complex interconnected channels, c) is the histogram of mesh resolution distribution.

Timeliness is a major constraint in a relocatable scenario. More specifically, the successful merging of the high-resolution subset mesh for the area of interest into the coarse mesh must be concluded within reasonable time (e.g., in a fraction of the forecast cycle) so it can provide detailed flood predictions with enough lead time. The proposed OCSMesh workflow can achieve this in less than 107 minutes for a 425,000-node local mesh that is combined into operational STOFS3D. The final mesh has almost 3.4 million nodes and can be deployed using STOFS3D configurations and setup. A step-by-step Jupyter Notebook tutorial covering this example application, which includes the floodplain and river mesh generation and merge into the STOFS3D mesh, is available on the OCSMesh GitHub repository[11].

5.2. Model Deployment and Results

Similar to the previous application, we defined a run time of 99 days starting on July 01, 2021. Given that in a relocatable case the main target is to improve flood prediction in a timely manner, we decided to adapt the STOFS3D-based deployment to a much simpler 2D setup. In SCHISM the main difference between a 3D and a 2D case is the vertical grid configuration, where in the 2D case only two levels (top and bottom) are specified in the vertical grid. All other inputs were kept the same as the baseline model, except for the 3D boundary forcings and hotstart which are no longer needed (Section 2.1). Fifty compute nodes for a total of 4,000 cores were used to run the 2D simulations (one based on the STOFS3D horizontal grid and the other based on the relocatable mesh) on a high-performance computing environment. Both 99-day simulations were concluded in less than 4 hours wall-clock time.

In order to assess the models' performance, we selected four water level stations: two located within the Pearl River Basin (02492600 upstream and 02492700 downstream) and two on the open water (301200090072400 on Lake Pontchartrain and 300722089150100 in the Mississippi Sound). We also selected three validation times: 1) thirty-six hours before the Category 4 Hurricane Ida made landfall in Mississippi delta; 2) twelve hours after the landfall, when much of the surge had propagated overland; and 3) five days after landfall when major river flows flooded the upper basin as a result of the heavy precipitation (Figure 12).

[11] Example Application 2 Tutorial: URL:https://github.com/noaa-ocs-modeling/OCSMesh/blob/main/Tutorials/ApplicationExample02.ipynb, last accessed in Apr 2025.

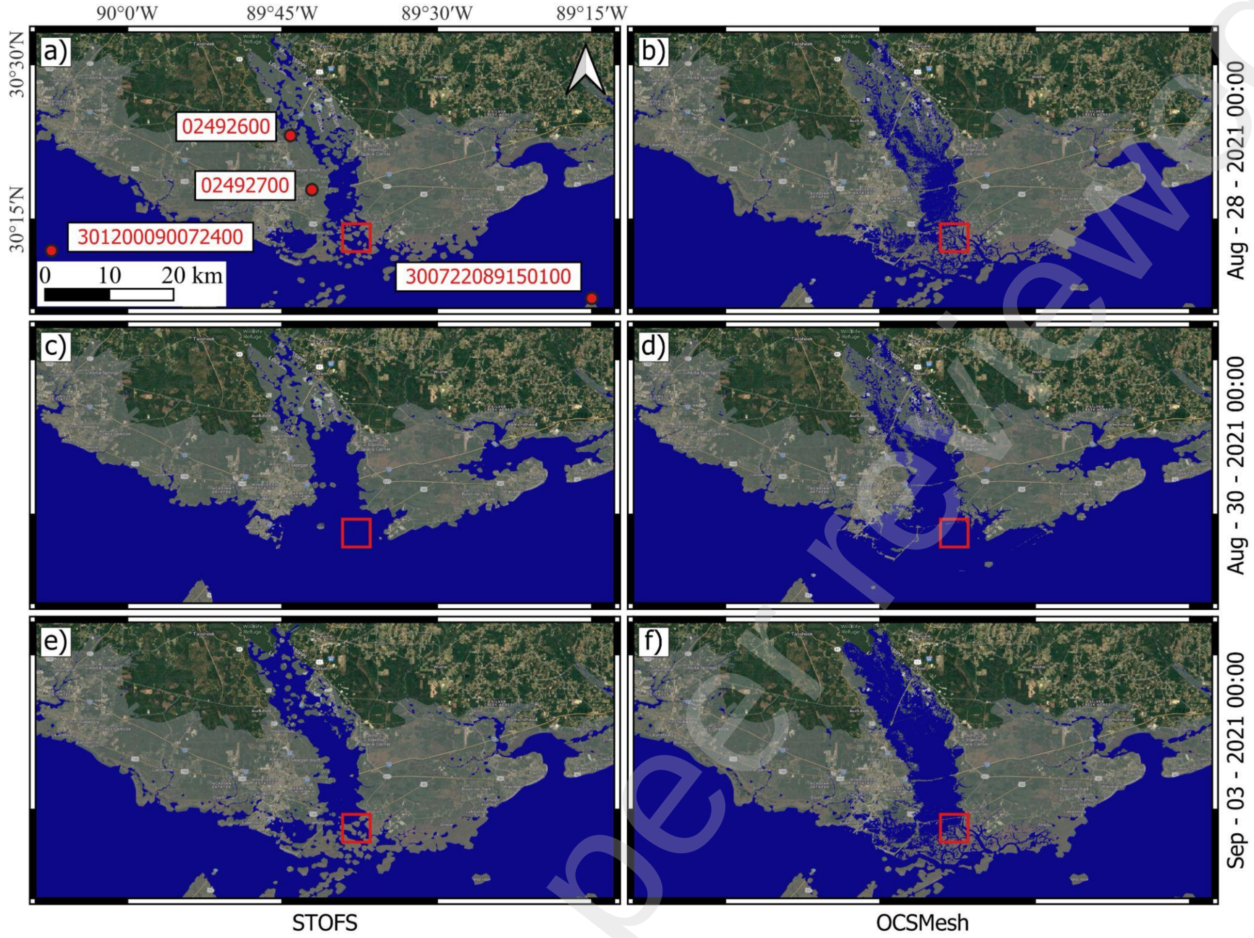

Figure 12. STOFS mesh-based (a, c, and e) and OCSMesh-based (b, d, and f) model wetting and drying on Aug-28 (a and b), Aug-30 (c and d), and Sep-03 2021 (e and f). Red dots represent the USGS stations used for model assessment. Red squares refer to the zoomed-in areas on Figure 7 and 13.

In the first snapshot, both models show the upper and lower Pearl River as mostly dry, with wet areas concentrated over the main channels (Figure 12a and 12b). In the second timestep, the lower half of the basin as well as the surrounding low-lying areas are completely wet as the storm surge propagates landward (Figure 12c and 12d). Four days later as the river flows increase the opposite pattern is observed, with most of the upper half of the basin being wet and the lower basin being dry (Figure 12e and 12f). Figure 13 depicts the wet model elements for the respective timesteps at a small area in the lower Pearl River Basin (see the area highlighted in Figure 12).

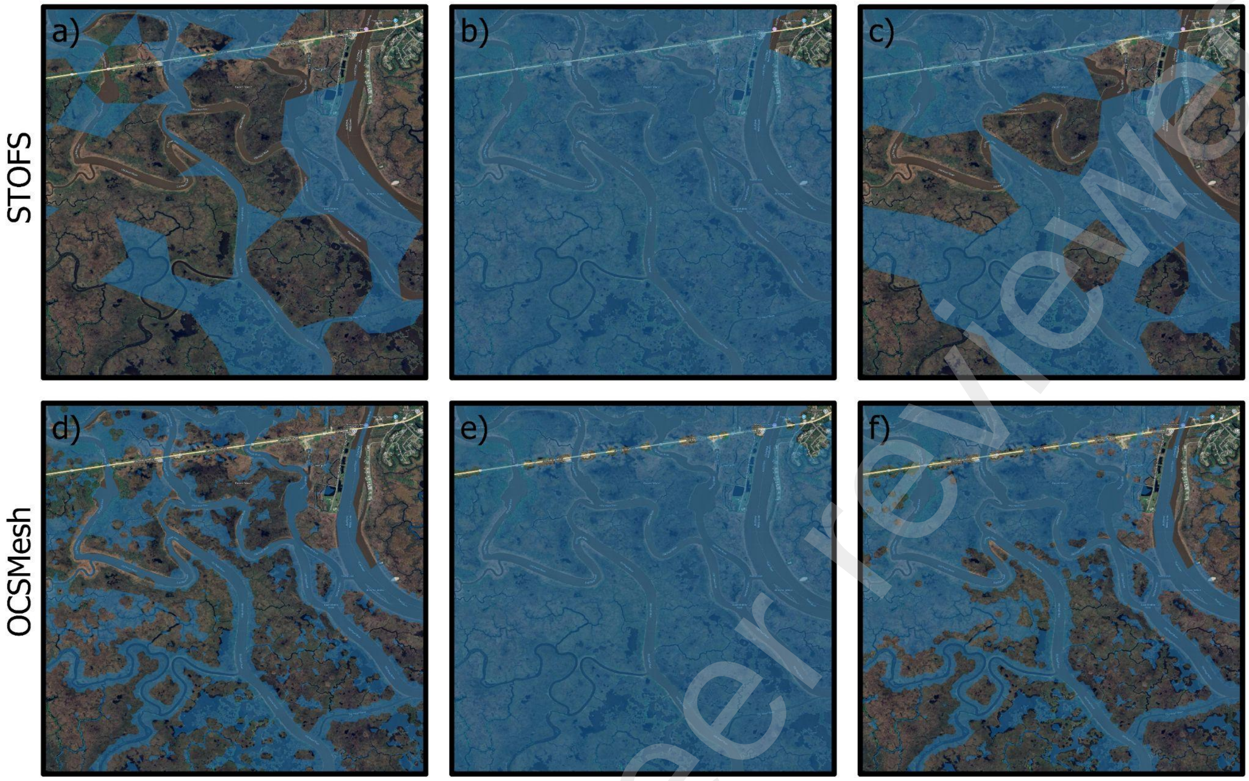

Figure 13. Zoomed in model wetting and drying on Aug-28 (a and d), Aug-30 (b and e), and Sep-03 2021 (c and f) for the STOFS mesh-based (a, b, and c) and OCSMesh-based (d, e, and f) models.

The higher resolution river mesh along with the higher resolution floodplain mesh resulted in a much more realistic wetting and drying (Figure 13d and 13f) when compared to the original STOFS3D horizontal mesh (Figures 13a and 13c). The STOFS3D mesh leads to a disjointed propagation of flow, which indicates its resolution is not ideal for such river deltas that are interconnected by narrow and sinuous channels. The OCSMesh mesh (see Figure 11), however, does not have the connectivity issues that are apparent in the coarser STOFS3D mesh, clearly resembling the complex river network seen in the basemap (Figure 13d and 13f). It can also be noted during the maximum surge snapshot (Figure 13e) that there are several breaches along US Highway 90 (Chef Menteur Highway), indicating that even the high (50-meters) resolution given to the floodplain might not be enough to capture all the hydrologically important structures, such as road embankments.

As expected, the better discretization of the river network had a direct impact on the model hydrographs by allowing more flow to propagate downstream (Figure 14a and 14b). Figures 14a and 14b show that the high-resolution relocatable model captured a higher flood peak that is closer to the upstream observations. In addition, the downscaled model also captured secondary flood peak signals from intense precipitation events following Hurricane Nicholas in late September that were not captured by the original model (Figure 14b). Yet, both models fail to faithfully represent the main features of the observed hydrograph.

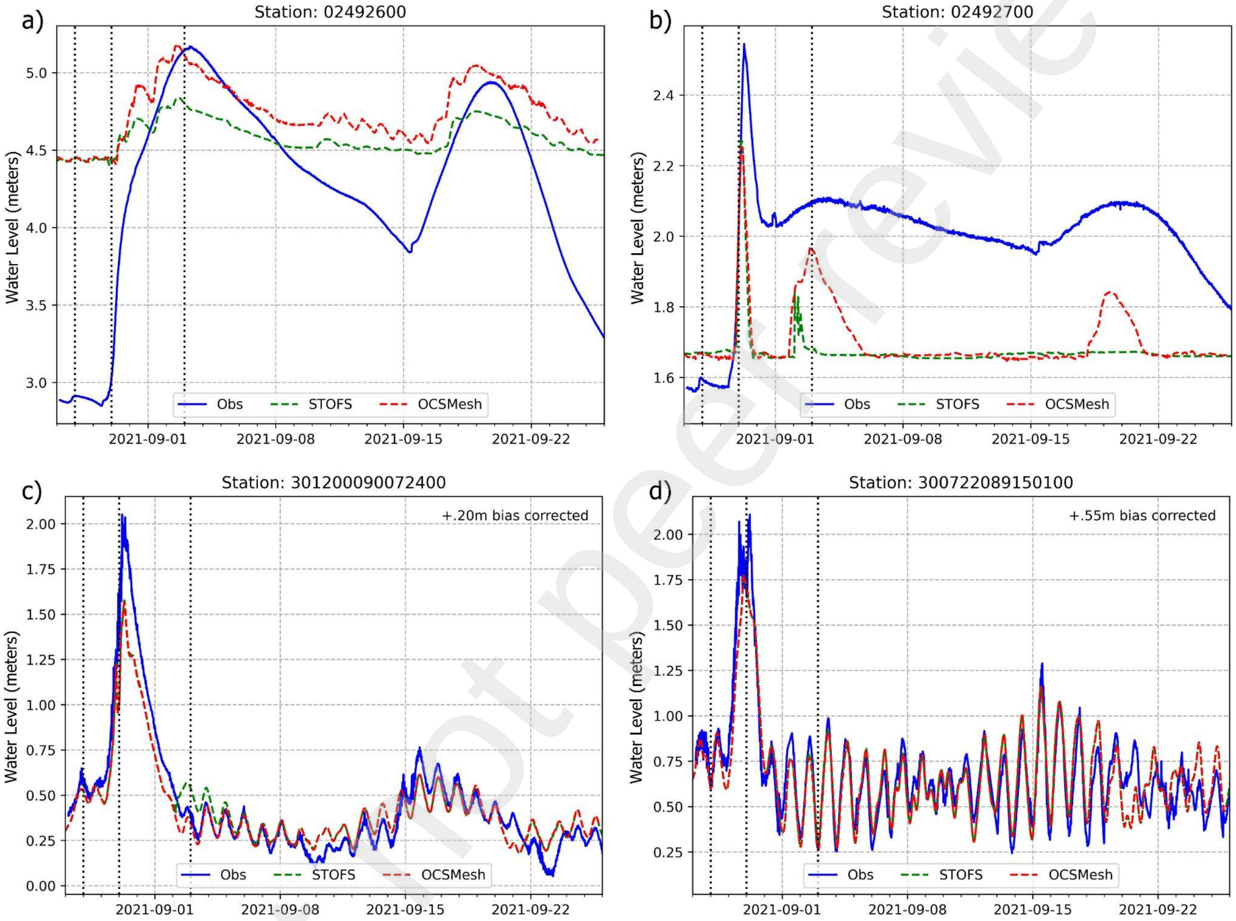

Figure 14. Water levels at the USGS stations highlighted in Figure 12. Figure a) is the most upstream station, b) is the downstream station, c) is the station in Lake Pontchartrain, and d) is the station in the Mississippi Sound. The vertical dashed lines indicate the timesteps (before, during, and after Hurricane Ida) highlighted in Figures 12 and 13. Water levels are referenced to NAVD88.

For instance, both models show a shorter basin lag-time when compared to the observations. The basin lag-time is controlled by basin characteristics such as length, slope, and drainage density, which can be captured by the mesh, and soil and land cover attributes (Wohl, 2013) that are not included in this model setup. If soil properties and vegetation are not present in the model (i.e., no interception and infiltration) a shorter lag-time and a steeper rising limb are expected as water can propagate faster downstream. Moreover, the model recession limbs also do not resemble the observed hydrograph. The slope of the recession limb is strongly controlled by the subsurface and base flows (Wohl, 2013), where the larger the below surface contributions are, the less steep the recession limb tends to be. Figure 14b shows a much steeper model recession limb when compared to station 02492700. That model behavior is also expected since only surface runoff contributions are accounted for in the models. The model recession limb at station 02492700 is less steep than the observed hydrograph. The main reason for this erroneous behavior is the initial water level overestimations. Prior to the rising limb, both models start from a much higher water level (approximately 4.5 meters) than the observations (approximately 3 meters). This erroneous starting point limits the potential recession of the modeled hydrograph to the levels observed on the data (4 meters by September 15). In addition, the larger water depths allow the upstream propagation of tides, which is not realistic when compared to the observations (Figure 14a).

The relocatable example shows similar results when compared to the original mesh-based model at locations away from the high-resolution subset (Figure 14c and 14d). This is a desirable outcome as it indicates that the effect of high-resolution mesh directly nested into the coarser mesh does not extend much further from the subsetted area, thus not affecting the model performance elsewhere.

6. Discussion and Conclusions

In this manuscript we introduced an automated unstructured mesh generation Python package targeting SCHISM-based applications, developed and maintained by the NOAA Office of Coast Survey (OCS). This data-driven, object-oriented package is highly flexible and accommodates skewed elements for faithful DEM-based mesh generation. Originally released in 2020, OCSMesh has an active user community and has been utilized for mesh generation in many SCHISM-based applications (Mani et al., 2022; Sun et al., 2022; Martins et al., 2024). Recent package updates allow for seamless end-to-end, creek-to-ocean mesh generation, which are explored in this manuscript through two example applications.

The first application presents a defensible and reproducible automated mesh generation workflow for cross-scale coastal ocean modeling, using a well-known operational forecast system (STOFS3D-Atlantic) as the basis for model setup and output comparison. The goal of this example was to show that the proposed workflow can achieve model predictions (e.g., water levels, temperature, salinity, and current velocities) that are comparable to those from the original STOFS3D model. Although reliant on outside inputs (i.e., DEMs and RiverMapper river delineation polygons) and model setup (Figure 4), the mesh generation process is completely self-contained and takes advantage of light-weight exact Delaunay triangulation libraries for otherwise computationally-heavy mesh merging steps (see Section 3.4). Our results show that the automated mesh can be easily imported within the STOFS3D workflow and closely reproduce the original model's qualitative and quantitative skill (Section 4.2). The proposed OCSMesh workflow bridges the gap raised by two recent studies that have successfully automated the river network delineation process (Ye et al., 2023) and the cross-scale three-dimensional SCHISM model deployment (Cui et al., 2024), thus completing a fully automated, end-to-end, cross-scale, model creation process.

The second example shows a variation of the aforementioned workflow (Figure 4) where a high-resolution river and floodplain mesh generated by OCSMesh for the Pearl River Basin is directly merged into the coarser STOFS3D mesh to create a relocatable model. The merge of the high-resolution local mesh into the STOFS3D mesh was performed in a little over 1.5 hours, easily fitting within a forecast cycle (usually 6 hours). It should also be mentioned that the generation of the high-resolution local mesh itself (50-meter resolution floodplain with a dense representation of the river network for the entire Pearl River Basin) did not take more than 30 minutes. Therefore, there is the potential for the deployable high resolution meshes to be generated on-the-fly (rather than pre-generated for the entire east coast). Overall, the high-resolution relocatable model performs as expected by creating detailed wetting and drying representations and stronger riverine flood signals. The efficiency of OCSMesh in rapidly creating these relocatable meshes, allied with its easy deployability within the STOFS3D automated setup, can lead to major improvements on coastal flood hazard prediction if implemented in an operational framework.

Both examples suggest that there is still room for end-to-end model generation improvements. Within what can be changed in the current model setup, we highlight: i) the proper placement of model stations, which might have to be 'adjusted' in the case of coarse river discretizations or survey errors (Zhang et al., 2024); ii) the proper placement of river heads; and

iii) the local calibration of model parameters, such as drag. To what pertains to mesh generation, improvements on the river flood signals are obtained when substantially increasing the river discretization and the floodplain resolution at the expense of higher computational costs (i.e., larger meshes). Lastly, some limitations cannot be addressed within the current framework. For instance, the steep modeled hydrographs at upstream stations cannot be reliably improved without the inclusion of a hydrological model component capable of accounting for subsurface flows, in/extrafiltration, and soil-water processes.

With this study we demonstrate that the mesh generation trilemma that attempts to balance accuracy, computational cost, and mesh ‘quality’ and often leads to ‘gray areas’ in the mesh generation process, can be overcome when mesh ‘quality’ (i.e., triangle equilateral-ness) is no longer a concern. SCHISM’s greater numerical stability due to its implicit nature, frees the automated mesh generator (OCSMesh and potentially other packages) to focus on delivering faithful representation of the DEM. The tool and reproducible workflow presented in this manuscript deliver a major contribution towards defensible coastal modeling, which has been strongly advocated for by the SCHISM modeling community (Ye et al., 2023; Cui et al., 2024; Zhang et al., 2024). Future OCSMesh development will focus on expanding the proposed workflow to the ocean domain by addressing the different mesh requirements for the eddy, non-eddy, and transitional regimes.

CRediT authorship contribution statement

Felicio Cassalho: Software, Methodology, Investigation, Formal analysis, Validation, Visualization, Data curation, Writing – original draft, Writing – review & editing. **Soroosh Mani:** Conceptualization, Formal analysis, Software, Methodology, Investigation, Writing – review & editing. **Fei Ye:** Investigation, Formal analysis, Data curation, Writing – review & editing. **Y. Joseph Zhang:** Conceptualization, Supervision, Writing – review & editing. **Saeed Moghimi:** Conceptualization, Scientific/Technical lead, Funding acquisition, Supervision, Project management, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Software and data availability

OCSMesh is an open-source package and can be accessed via GitHub (https://github.com/noaa-ocs-modeling/OCSMesh). The two application examples can be reproduced using step-by-step Jupyter Notebooks made available on the OCSMesh GitHub repo. The model deployment was based on STOFS3D-Atlantic automated workflow and can be accessed on the SCHISM GitHub repository (https://github.com/schism-dev/schism). Any remaining data will be made available on request.

Acknowledgements

This development was partially funded by NOAA COASTAL Act, NOAA Water Initiative, NOAA DRSA (Disaster Relief Supplemental Act) and BIL (Bipartisan Infrastructure Law). Simulations used in this paper were conducted using the following computational facilities: i) NOAA's Research and Development High Performance Computing Systems (RDHPCS), including HERA - NOAA Environmental Security Computing Center (NESCC) in Fairmont-WV, and HERCULES - Mississippi State University (MSU) in Starkville-MS; and ii) Texas Advanced Computing Center (TACC) at The University of Texas at Austin.

References

Beven II, J.L., Hagen, A., Berg, R., 2022. Tropical Cyclone Report: Hurricane Ida (AL092021). NOAA National Hurricane Center. https://www.nhc.noaa.gov/data/tcr/AL092021_Ida.pdf

Brown, D.P., 2021. Tropical Cyclone Report: Hurricane Larry (AL122021). NOAA National Hurricane Center. https://www.nhc.noaa.gov/data/tcr/AL122021_Larry.pdf

Cai, X., Zhang. Y.J., Shen, J., Wang, H., Wang, Z., Qin, Q., Ye, F., 2020. A Numerical Study of Hypoxia in Chesapeake Bay Using an Unstructured Grid Model: Validation and Sensitivity to Bathymetry Representation. Journal of the American Water Resources Association, JAWR-20-0023-P. https://doi.org/10.1111/1752-1688.12887

Cai, X., Shen, J., Zhang. Y.J., Qin, Q., Wang, Z., Wang, H., 2021. Impacts of Sea-Level Rise on Hypoxia and Phytoplankton Production in Chesapeake Bay: Model Prediction and Assessment. Journal of the American Water Resources Association, JAWR-20-0043-P. https://doi.org/10.1111/1752-1688.12921

Calzada, J.R., Cui L., Zhang. Y.J., 2023. schism-dev/pyschism: v0.1.5 (v0.1.5). Zenodo. https://doi.org/10.5281/zenodo.7623122

Chassignet, E.P., Hurlburt, H.E., Smedstad, O.M., Halliwell, G.R., Hogan, P.J., Wallcraft, A.J., Baraille, R., Bleck, R., 2007. The HYCOM (HYbrid Coordinate Ocean Model) data assimilative system. Journal of Marine Systems, 65(1-4), 60-83. https://doi.org/10.1016/j.jmarsys.2005.09.016

Conroy, C.J., Kubatko, E.J., West, D.W., 2012. ADMESH: An advanced, automatic unstructured mesh generator for shallow water models. Ocean Dynamics, 62(10-12), 1503-1517. https://doi.org/10.1007/s10236-012-0574-0

Cosgrove, B., Gochis, D., Flowers, T., Dugger, A., Ogden, F., Graziano, T., Clark, E., Cabell, R., Casiday, N., Cui, Z., Eicher, K., Fall, G., Feng, X., Fitzgerald, K., Frazier, N., George, C., Gibbs, R., Hernandez, L., Johnson, D. Zhang, Y., 2024. NOAA's National Water Model: Advancing operational hydrology through continental-scale modeling. JAWRA Journal of the American Water Resources Association, 60(2), 247-272. https://doi.org/10.1111/1752-1688.13184

Cui, L., Ye, F., Zhang, Y.J., Yu, H., Wang, Z., Moghimi, S., Seroka, G., Riley, J., Pe'eri, S., Mani, S., Myers, E., Park, K., Tang, L., Yang, Z., Wang, Y.-M., 2024. Total water level prediction at continental scale: Coastal ocean. Ocean Modelling, 192. https://doi.org/10.1016/j.ocemod.2024.102451

Danilov, S., Wang, Q. 2015. Resolving eddies by local mesh refinement. Ocean Modelling, 93, 75-83. https://doi.org/10.1016/j.ocemod.2015.07.006

Engwirda, D., 2014. Locally Optimal Delaunay-refinement and Optimisation-based Mesh. PhD Thesis. School of Mathematics and Statistics - Faculty of Science, The University of Sydney. http://hdl.handle.net/2123/13148

Gorman, G.J., Piggott, M.D., Wells, M.R., Pain, C.C., Allison, P.A., 2008. A systematic approach to unstructured mesh generation for ocean modelling using GMT and Terreno. Computers & Geosciences, 34(12), 1721-1731. https://doi.org/10.1016/j.cageo.2007.06.014

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., Thépaut, J.N., 2020. The ERA5 global reanalysis. Quarterly Journal of the Royal Meteorological Society, 146(730), 1999-2049. https://doi.org/10.1002/qj.3803

Holleman, R., Fringer, O., Stacey, M., 2013. Numerical diffusion for flow-aligned unstructured grids with application to estuarine modeling. International Journal for Numerical Methods in Fluids, 72(11), 1117-1145. https://doi.org/10.1002/fld.3774

Jonasson, O., Gladkova, I., Ignatov, A., 2022. Towards global daily gridded super-collated SST product from low earth orbiting satellites (L3S-LEO-Daily) at NOAA, Proc. SPIE 12118, Ocean Sensing and Monitoring XIV, 1211805. https://doi.org/10.1117/12.2620103

Latto, A.S., Berng, R., 2022. Tropical Cyclone Report: Hurricane Nicholas (AL142021). NOAA National Hurricane Center. https://www.nhc.noaa.gov/data/tcr/AL142021_Nicholas.pdf

Lyard, F.H., Allain, D.J., Cancet, M., Carrère, L., Picot, N., 2021. FES2014 global ocean tide atlas: design and performance. Ocean Science, 17(3), 615-649. https://doi.org/10.5194/os-17-615-2021

Mani, S., Calzada, J.R., Moghimi, S., Zhang, Y.J., Myers, E., Pe'eri, S., 2021. OCSMesh: a data-driven automated unstructured mesh generation software for coastal ocean modeling. NOAA Technical Memorandum NOS CS, 47. https://doi.org/10.25923/csba-m072

Mani, S., Moghimi, S., Cui, L., Wang, Z., Zhang, Y.J., Lopez, J., Myers, E., Cockerill, T., Pe'eri, S., 2022. On-demand Automated Storm Surge Modeling Including Inland Hydrology. NOAA Technical Memorandum NOS CS, 52. https://repository.library.noaa.gov/view/noaa/47926

Martins, R., Fortunato, A.B., Oliveira, A., Jesus, G., Mani, S., Myers, E., Moghimi, S., 2024. OPENMeshS: an online, open, unstructured mesh generator for OPENCoastS 8th IAHR Europe Congress, Lisbon, Portugal.

O'Callaghan, J.F., Mark, D.M., 1984. The extraction of drainage networks from digital elevation data. Comput. Vis. Graph Image Process 28 (3), 323–344.

Orlandini, S., Moretti, G., Corticelli, M.A., Santangelo, P.E., Capra, A., Rivola, R., Albertson, J.D., 2012. Evaluation of flow direction methods against field observations of overland flow dispersion. Water Resources Research, 48(10). https://doi.org/10.1029/2012wr012067

Pasch, R.J., Robbie, B., Andrew, B.H., 2022. Tropical Cyclone Report: Hurricane Henri (AL082021). NOAA National Hurricane Center. https://www.nhc.noaa.gov/data/tcr/AL082021_Henri.pdf

Pasch, R.J., Roberts, D.P., 2022. Tropical Cyclone Report: Hurricane Sam (AL182021). NOAA National Hurricane Center. https://www.nhc.noaa.gov/data/tcr/AL182021_Sam.pdf

Reinhart, B.J., Reinhart, A., Berg, R., 2022. Tropical Cyclone Report: Hurricane Grace (AL072021). NOAA National Hurricane Center. https://www.nhc.noaa.gov/data/tcr/AL072021_Grace.pdf

Remacle, J.F., Lambrechts, J., 2018. Fast and robust mesh generation on the sphere-Application to coastal domains. Computer-Aided Design, 103, 14-23. https://doi.org/10.1016/j.cad.2018.03.002

Roberts, K.J., Pringle, W.J., Westerink, J.J., 2019. OceanMesh2D 1.0: MATLAB-based software for two-dimensional unstructured mesh generation in coastal ocean modeling. Geoscientific Model Development, 12(5), 1847-1868. https://doi.org/10.5194/gmd-12-1847-2019

Shewchuk, J.R., 2002. Delaunay Refinement Algorithms for Triangular Mesh Generation. Computational Geometry, 22, 21–74. https://doi.org/10.1016/S0925-7721(01)00047-5

Sun, Y., Velissariou, P., Moghimi, S., Turuncoglu, U., Zhang, Y.J., Mani, S. and Myers, E., 2024, February. Impacts of Wave-Current Interactions on Storm Surge during the Passage of Hurricane Ian (2022): An Application of the UFS-Coastal Modeling Framework. In 104th AMS Annual Meeting. AMS.

Trotta, F., Federico, I., Pinardi, N., Coppini, G., Causio, S., Jansen, E., Iovino, D., Masina, S., 2021. A Relocatable Ocean Modeling Platform for Downscaling to Shelf-Coastal Areas to Support Disaster Risk Reduction. Frontiers in Marine Science, 8. https://doi.org/10.3389/fmars.2021.642815

Wahl, T., Jain, S., Bender, J., Meyers, S.D., Luther, M.E., 2015. Increasing risk of compound flooding from storm surge and rainfall for major US cities. Nature Climate Change, 5(12), 1093-1097. https://doi.org/10.1038/nclimate2736

Wohl, E., 2013. Streams. Reference Module in Earth Systems and Environmental Sciences, Elsevier. doi: 10.1016/B978-0-12-409548-9.03816-1.

Ye, F., Cui, L., Zhang, Y.J., Wang, Z., Moghimi, S., Myers, E., Seroka, G., Zundel, A., Mani, S., Kelley, J.G.W., 2023. A parallel Python-based tool for meshing watershed rivers at continental scale. Environmental Modelling & Software, 166. https://doi.org/10.1016/j.envsoft.2023.105731

Ye, F., Zhang, Y.J., He, R., Wang, Z., Wang, H.V., Du, J., 2019. Third-order WENO transport scheme for simulating the baroclinic eddying ocean on an unstructured grid. Ocean Modelling, 143. https://doi.org/10.1016/j.ocemod.2019.101466

Ye, F., Zhang, Y.J., Wang, H.V., Friedrichs, M.A.M., Irby, I.D., Alteljevich, E., Valle-Levinson, A., Wang, Z., Huang, H., Shen, J., Du, J., 2018. A 3D unstructured-grid model for Chesapeake Bay: Importance of bathymetry. Ocean Modelling, 127, 16-39. https://doi.org/10.1016/j.ocemod.2018.05.002

Ye, F., Zhang, Y.J., Yu, H., Sun, W., Moghimi, S., Myers, E., Nunez, K., Zhang, R., Wang, H.V., Roland, A., Martins, K., Bertin, X., Du, J., Liu, Z., 2020. Simulating storm surge and compound flooding events with a creek-to-ocean model: Importance of baroclinic effects. Ocean Modelling, 145. https://doi.org/10.1016/j.ocemod.2019.101526

Zhang, Y.J., Anderson, J., Park, K., Wu, C.H., Wipperfurth, S., Anderson, E., Pe'eri, S., Beletsky, D., Titze, D., Di Lorenzo, E., Moghimi, S., Seroka, G., Myers, E., Fujisaki-Manome, A., Kelley, J., 2024. Debunking common myths in coastal circulation modeling. Ocean Modelling, 190. https://doi.org/10.1016/j.ocemod.2024.102401

Zhang, Y.J., Ateljevich, E., Yu, H.-C., Wu, C.H., Yu, J.C.S., 2015. A new vertical coordinate system for a 3D unstructured-grid model. Ocean Modelling, 85, 16-31. https://doi.org/10.1016/j.ocemod.2014.10.003

Zhang, Y.J., Baptista, A.M., 2008. SELFE: A semi-implicit Eulerian–Lagrangian finite-element model for cross-scale ocean circulation. Ocean Modelling, 21(3-4), 71-96. https://doi.org/10.1016/j.ocemod.2007.11.005

Zhang, H., Yao, Z., Yang, Q., Li, S., Baartman, J.E.M., Gai, L., Yao, M., Yang, X., Ritsema, C.J., Geissen, V., 2017. An integrated algorithm to evaluate flow direction and flow accumulation in flat regions of hydrologically corrected DEMs. Catena, 151, 174-181. https://doi.org/10.1016/j.catena.2016.12.009

Zhang, Y.J., Ye, F., Stanev, E.V., Grashorn, S., 2016. Seamless cross-scale modeling with SCHISM. Ocean Modelling, 102, 64-81. https://doi.org/10.1016/j.ocemod.2016.05.002