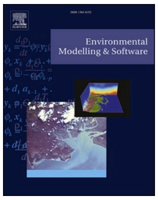
Contents lists available at ScienceDirect

Environmental Modelling and Software

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



A parallel Python-based tool for meshing watershed rivers at continental scale

Fei Ye[a,*], Linlin Cui[a], Yinglong Zhang[a], Zhengui Wang[a], Saeed Moghimi[b], Edward Myers[b], Greg Seroka[b], Alan Zundel[c], Soroosh Mani[b,d], John G.W. Kelley[b]

[a] *Virginia Institute of Marine Science, William and Mary, Gloucester Point, VA, 23062, USA*
[b] *Coast Survey Development Laboratory, National Ocean Service, NOAA, Silver Spring, MD, 20910, USA*
[c] *Aquaveo, 3210 N. Canyon Road Suite 300, Provo, UT, 84604, USA*
[d] *Spatial Front Inc., Bethesda, MD, 20817, USA*

ARTICLE INFO

Handling Editor: Daniel P Ames

Keywords:
Compound flooding
Coastal transition zone
Cross-scale modeling
Unstructured mesh generation

ABSTRACT

A major challenge in compound flooding simulation is representing small rivers in the model mesh. A parallel Python-based tool is developed to support the construction of unstructured grids (UGs) at continental scales. The tool is driven by Digital Elevation Model (DEM), ensuring accurate representation of geomorphic features in the resulting mesh. Its first component (pyDEM) extends an existing tool to detect river thalwegs from DEMs. The second component (RiverMapper) uses the thalwegs to generate river 'arcs', which can be directly ingested into meshing tools (e.g., the Surface-water Modeling System). The novelty lies in the explicit 2D representation of rivers in both along- and cross-channel directions, making it ideal for accurate, efficient, and high-resolution continental-scale research. The tool is employed to create a UG for a 3D creek-to-ocean model encompassing the US East and Gulf Coasts, and it greatly improves the flow exchange between the watershed and the coastal zone.

Software and data availability

1. Python-based tools: pyDEM (co-developed by Linlin Cui, lcui@vims.edu and Zhengui Wang, wangzg@vims.edu) and RiverMapper (developed by Fei Ye, feiye@vims.edu). Both tools along with the user manual and sample applications are freely accessible from the 'SCHISM development' GitHub repository: https://github.com/schism-dev/RiverMeshTools.

The standalone pyDEM package originally developed by Zhengui Wang is also available at: https://github.com/wzhengui/pydem.

2. Mesh generation tool (SMS v13.2 or newer): commercial software developed and maintained by Aquaveo (aquaveo.com), with a community version freely available at https://www.aquaveo.com/software/sms-community. The tool requires Windows 10 and above.
3. Compound flood model (SCHISM): open-source community model developed by Yinglong J. Zhang (VIMS), freely accessible at: schism.wiki (v5.10.0).

1. Introduction

The increasing probabilities of 'compound floods' (i.e., high water levels caused by multiple flood drivers) under the projected climate change present an urgent need for comprehending this type of high-risk events (Wahl et al., 2015; Bevacqua et al., 2019). Physical models represent a powerful means for analyzing the processes and predicting the extent of compound floods (Xu et al., 2022). Ideally, a *single* modeling framework should be used to simulate the nonlinear interactions among flood drivers (Santiago-Collazo et al., 2019), which include rainfall, runoff, and storm surge (Wahl et al., 2015; Bilskie and Hagen, 2018), as well as high tides (Jang and Chang, 2022), waves (Qiang et al., 2021), off-shore oceanic processes (Ezer, 2018), and sea level rise (Moftakhari et al., 2017; Del-Rosal-Salido et al., 2021). This modeling technique is termed 'full-coupling' if all relevant equations from hydrology, hydraulics, and hydrodynamics are solved simultaneously (Santiago-Collazo et al., 2019). The study conducted by Stephens et al. (2022) at Galveston Bay demonstrated the superior accuracy of the 'full-coupling' approach. In our previous studies (Huang et al.,

[*] Corresponding author.
E-mail address: feiye@vims.edu (F. Ye).

https://doi.org/10.1016/j.envsoft.2023.105731
Received 24 January 2023; Received in revised form 12 April 2023; Accepted 18 May 2023
Available online 24 May 2023

2021; Ye et al., 2021) and NOAA[1]'s operational forecast (STOFS-3-D-Atlantic[2]), we confirmed the feasibility of efficiently applying a full-coupling approach at the continental scale by employing a single model to simulate hydrodynamic, hydraulic, and hydrologic processes (excluding groundwater flow and evapotranspiration). The computational cost was effectively mitigated by the model's implicit solver for the shallow-water equation, which enables the use of large time steps even on very high mesh resolutions (Zhang et al., 2016, 2020; Ye et al., 2020).

Given the geomorphic complexity of coastal transition zones, the full-coupling approach strongly favors unstructured-grid (UG) models, and the remaining challenges of large-scale compound flood modeling are often associated with the complexity of generating an appropriate model mesh. The importance of accurately representing river channels in the mesh is highlighted by both uncertainty analysis (Willis et al., 2019) and model applications on specific events (Ye et al., 2021; Stephens et al., 2022), as these channels serve as the main conduit for floodwater from all flood drivers. While the use of raster-based sub-grid approach has become popular recently (Casulli and Stelling, 2011; Costabile et al., 2020), it faces difficulties in representing defense structures such as levees and dams (Zhang, 2021). Additionally, although the finite-volume based sub-grid models can utilize UGs, they require strict orthogonal UGs that are extremely challenging to generate for complex coastal topography and bathymetry (Zhang, 2021).

Translating a model domain into a UG often requires automated and manual steps. A mesh generator can discretize a given geometric entity based on certain algorithms such as sweep-line methods, Delaunay-Voronoi methods, and others explained by Thompson et al. (1998). Examples of mesh generators for free-surface environmental and geophysical flows include Surface-water Modeling System (SMS[3]), OCSMesh (Mani et al., 2022), Mike Mesh Builder,[4] OceanMesh2D (Roberts et al., 2019), JIGSAW (Engwirda, 2017), Deltares MeshKernel,[5] etc. Furthermore, optimizing a UG for coastal applications often involves manually placing polygons or polylines that follow topographic and bathymetric features such as levees, thalwegs, and characteristic contours. Overall, mesh generation for large-scale compound flood modeling is an arduous undertaking given the geometric complexity of nearshore and watershed regions, which are characterized by a multitude of fine-scale geomorphic features. Despite the capability of UGs, achieving high-resolution of these features while ensuring good resolution for other critical features essential for flow routing pushes both the mesh generator and model to the limit. As an illustration, in the case study presented in Section 3, the total number of polylines employed to guide the watershed mesh generation is approximately 350,000. Furthermore, optimizing the UG to reduce the mesh size for efficient model performance would inevitably require sharp transitions in mesh resolution. In the end, incorporating fine-scale features into the unstructured grid (UG) requires a heavily constrained triangulation, which can be challenging for many meshing tools.

Recent studies have been exploring more efficient meshing procedures that are capable of automatic local refinements (Conroy et al., 2012; Araújo et al., 2013; Roberts et al., 2019; Bacopoulos and Hagen, 2022; Mani et al., 2022). These methods enable automatic control of the spatially varying mesh resolution using size functions, which are based on the length scale of geomorphic features or physical phenomena. However, despite the popularity of size-function-based methods, our experience suggests that they often produce excessively large meshes due to a preference for smooth transitions in mesh resolution. In addition, many of these tools lack the ability to implement anisotropic elements, i.e., elements with different length scales in different directions. An exception is the strategy proposed by Legrand et al. (2007), which was used to create anisotropic meshes with skew elements across a continental slope.

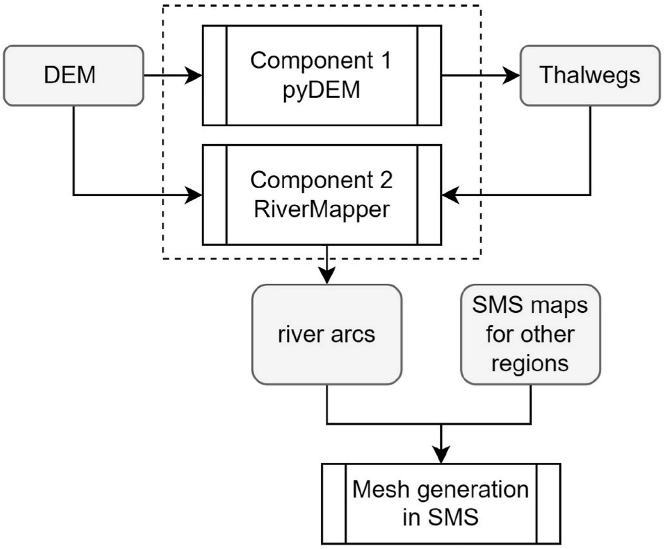


Fig. 1. Example meshing workflow for compound flooding simulations, using an automatic watershed 'map' (i.e., river arcs) generation tool with two main components: (1) pyDEM for extracting thalwegs from DEM tiles; (2) RiverMapper for generating river arcs for meshing.

In this paper, we propose a direct meshing approach for watershed rivers in which all essential features are explicitly specified using a Python tool. This approach is designed to provide greater control over the final mesh size and is entirely DEM-driven. After mesh generation, the bathy-topo DEMs are linearly interpolated onto the resultant mesh without any subjective manipulations (e.g., bathymetry smoothing) on the grid depth. Consequently, this approach directly links DEM quality to model accuracy. It also greatly reduces the turnaround time from new releases of bathy-topo datasets to their integration into operational forecasts.

To facilitate continental-scale applications, the tool is parallelized using 'mpi4py' (version 3.1.3). The first component of the tool (pyDEM) finds the 1D thalweg network of river channels using the hydrological network methodology (Section 2.1). Then, the 1D thalweg network is expanded by the second tool component (RiverMapper) to form a 2D representation of rivers. The 2D representation consists of polylines (cf. Fig. 9) to guide the delineation of river channels such that all line segments and vertices should be exactly reproduced in the final mesh. The polylines can be output as 'LineStrings' in the Esri Shapefile format for general uses with a user's preferred meshing tool; or in this study, as 'feature arcs' in the SMS map format. The meshing tool SMS is used for triangulation (Section 2.3) because it strictly preserves the user-specified polylines ('feature arcs'), which usually represent distinctive features of a terrain, such as shorelines, thalwegs, banks, and the outlines of hydraulic structures. At the same time, it also allows for resolution relaxation where the topographic gradient is small, such as in the non-river portion of a watershed where pluvial floods may occur. This helps maintain a reasonable mesh size thus improving simulation efficiency. The resultant mesh can resolve very narrow channels (often found in upstream rivers) down to the native DEM resolution to ensure channel continuity and unimpeded flow routing. In Section 3, the

[1] National Oceanic and Atmospheric Administration, U.S. Department of Commerce.

[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: Apr 6, 2023.

[3] URL: https://www.aquaveo.com/software/sms-surface-water-modeling-system-introduction, last accessed: Apr 6, 2023.

[4] URL: https://www.mikepoweredbydhi.com/products/mike-cloud/mesh-builder/, last accessed: Apr 6, 2023.

[5] URL: https://github.com/Deltares/MeshKernel, last accessed: Apr 6, 2023.

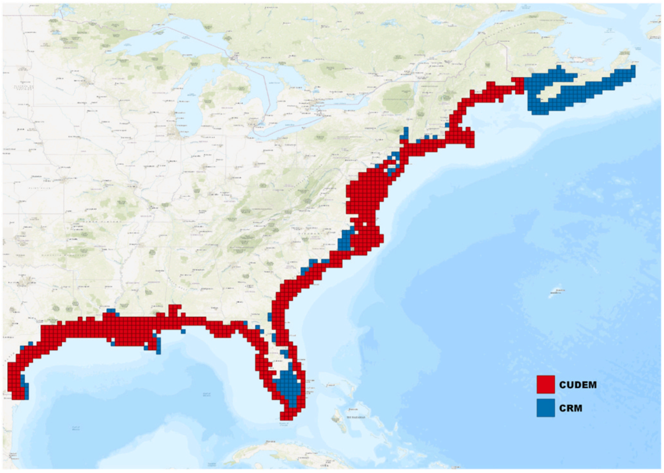


Fig. 2. Map of bathy-topo DEM tiles used along the U.S. East Coast and Gulf of Mexico (red: CUDEM bathy-topo tiles; blue: ninth arc-second resolution tiles interpolated from CRM).

meshing methodology and its products are tested with a 3D creek-to-ocean cross-scale model on a domain covering the US East and Gulf Coasts. Model results during hurricanes demonstrate that the new mesh can more accurately route the river flow than a previously generated mesh (same as the one used in Ye et al., 2021) that heavily relied on manual editing of channels. Major findings from this study are summarized in Section 4.

2. Methods

The new tool consists of two major components, both written in parallel Python. The first component generates a 1D network of river thalwegs from the DEMs. The second component utilizes existing thalweg information and DEMs to detect riverbanks, then places polylines (river arcs) based on bank positions. The river arcs are directly imported to a mesh generator (for example, SMS) to guide the meshing for watershed rivers. These steps are illustrated in Fig. 1 and expounded in detail below. Further information and sample applications can be found in the online documentation (see ‘Software and data availability’).

2.1. *Thalweg detection by pyDEM*

In this section, we use a spatial domain that covers the coastal zones along the US East and Gulf Coasts (Fig. 2) to illustrate the thalweg detection procedure. The primary DEM source used is NOAA National Centers for Environmental Information (NCEI)’s ninth arc-second resolution bathymetric-topographic DEM tiles (CUDEM[6]), augmented by the Coastal Relief Model (CRM[7]) that fills in gaps in a few regions where CUDEM tiles are missing. There are a total of 754 CUDEM tiles, with 8112 × 8112 raster cells per tile. The original resolution of the CRM dataset is three arc-seconds (~90 m), and we interpolate CRM into the

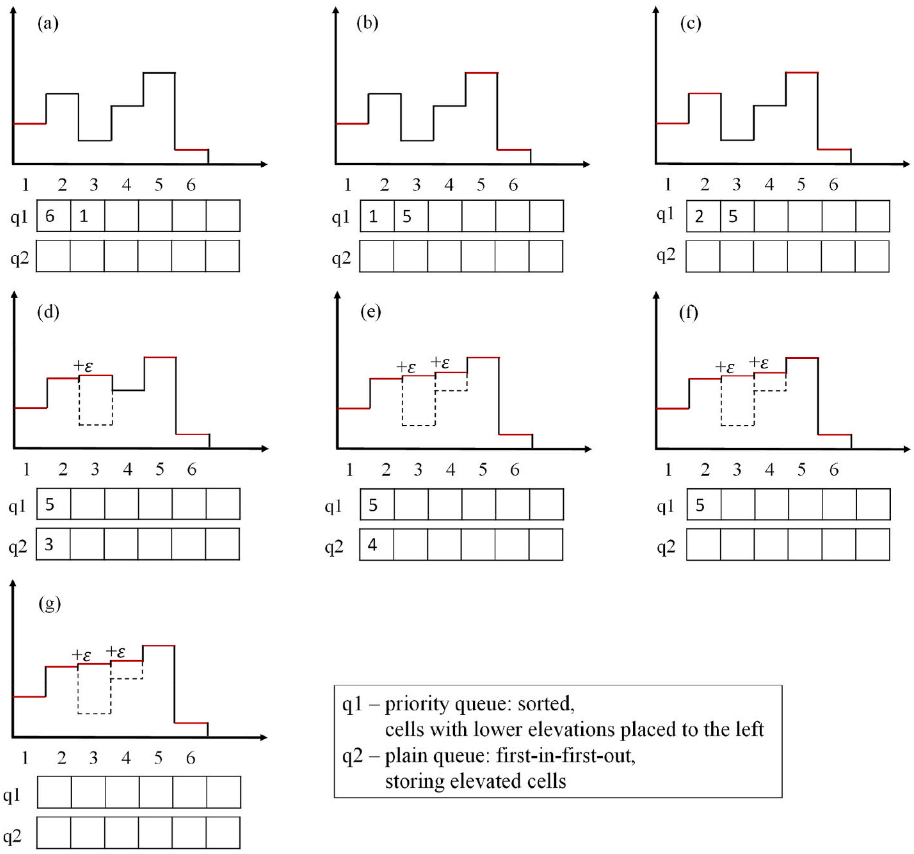


Fig. 3. The depression-filling algorithm, i.e., Alg. 3 (priority-flood + ε) from Barnes et al. (2014), illustrated with a cross-section of a DEM. Cells of a depression (in this case, cells ‘3’ and ‘4’) are raised to its lowest outlet (in this case, cell ‘2’), with a minimum difference of ε (a small positive number) between consecutive cells ensuring proper drainage.

[6] URL: https://chs.coast.noaa.gov/htdata/raster2/elevation/NCEI_ninth_Topobathy_2014_8483/, last accessed: Apr 6, 2023.

[7] URL: https://www.ncei.noaa.gov/products/coastal-relief-model, last accessed: Apr 6, 2023.

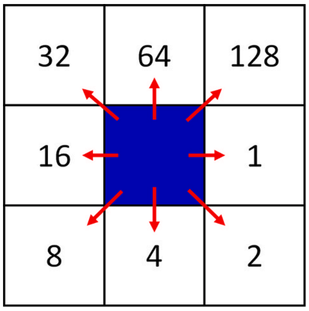


Fig. 4. D8 flow coordinate system consisted of 9 raster cells. The numbers are local indices that correspond to each neighbor of the center cell.

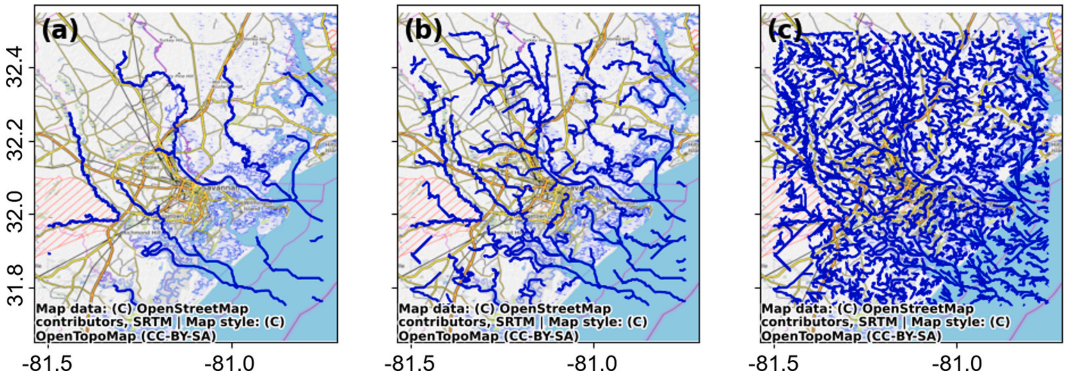

(a)
(b)
(c)
32.4
32.2
32.0
31.8
-81.5
-81.0
Map data: (C) OpenStreetMap contributors, SRTM | Map style: (C) OpenTopoMap (CC-BY-SA)

Fig. 5. Examples of thalwegs extracted with flow accumulation threshold of (a) 10^7, (b) 10^6, (c) 10^5 in Savannah, GA. Only thalwegs with flow accumulation numbers larger than the threshold will be included and the upstream thalwegs are omitted.

ninth arc-second resolution because the thalweg detection algorithm presented in this subsection requires uniform DEM tiles that dovetail with each other. The completed coverage, consisting of 1135 DEM tiles, is displayed in Fig. 2.

For the parallel algorithm to work, we first locally augment each DEM tile with its 8 neighboring tiles. This is done to avoid gaps between the calculated thalwegs and the border of each tile. After the thalwegs are calculated from the local 9-tile DEMs, only the thalwegs within the center tile are used for merging.

Extracting channel networks from a locally merged 9-tile DEM consists of three steps: (1) fill or breach depressions in the DEMs; (2) calculate flow direction based on the given flow metric; (3) calculate flow accumulation numbers, which are the total number of flow cells passing through each (raster) cell, and then apply a prescribed flow accumulation threshold to determine which cells should be included in the final digital stream network. The details of the three steps are elaborated below.

2.1.1. *Filling depression*

The algorithm used in this paper is Alg. 3 (priority-flood + ε) in Barnes et al. (2014). For completeness, this algorithm is illustrated along a DEM's cross-section (Fig. 3). Basically, it fills any local depressions (cells '3' and '4' in Fig. 3) to its lowest outlet (cell '2' in Fig. 3). Two queues are used to track the cells being processed: 'q1' is a priority queue, where cells with lower elevations are placed towards the head (left) of the sorted list and will be popped out first; and 'q2' is a first-in-first-out queue for storing elevated cells (dotted lines in Fig. 3). The procedure starts from inserting edge cells into q1 (Fig. 3a). While either q1 or q2 is not empty, the algorithm pops out a cell from either queue, prioritizing q2 as the source if it is not empty (for example, in Fig. 3a, the cell to be popped is '6' from q1 because q2 is empty; in Fig. 3d, the cell to be popped is '3' from q2), then push the un-processed neighbors (if any) of the popped cell to either q1 or q2. The destination queue is determined by comparing the neighbor's elevation to the modified elevation of the popped cell (i.e., the elevation of the popped cell plus a small positive number ε). The neighbor cell is pushed to q1 if its elevation is higher than the modified elevation (e.g., Fig. 3a and b and Fig. 3b and c); otherwise, the neighbor cell is raised to the modified elevation then pushed to q2 (e.g., Fig. 3c and d and Fig. 3d and e). The algorithm terminates when both q1 and q2 are empty.

2.1.2. *Calculating flow direction*

The D8 flow method (O'Callaghan and Mark, 1984) is used to assign a flow direction from each cell to its steepest downslope neighbor. There is only one such neighbor after the fill-depression step. The flow coordinate system is shown in Fig. 4. For example, if the direction of steepest drop is to the north of the cell of interest (blue), its flow direction value would be 64. The output of D8 flow direction calculation is an integer array associated with each cell. In addition to the indices shown in Fig. 4, there are two more special indices reserved for some special cases, with '-1' representing no-DEM data cells, and '0' representing cells on the edges of the (locally merged) DEM tile flowing outwards.

2.1.3. *Calculating flow accumulation*

This process starts from the 'outlets' cells on the boundary (of a locally merged 9-tile DEM) where the flow direction is flagged as zero, and then recursively searches upstream. For large DEMs, the regular recursion algorithm may fail because of the maximum recursion depth allowed in Python. Therefore, we implement a multi-level recursive search: when the number of search steps exceeds 100, the search will restart from the current location for another round of search, and so on, until the boundary is reached. All channel segment information is saved along the search so that we can calculate the total accumulation number for any given raster cell; the boundary cells ('0') will have the largest accumulation number. To avoid excessive details of the thalweg network, we set an accumulation threshold beyond which we stop the search and thus remove all upstream cells from the final network. Therefore, the flow accumulation threshold can be regarded as a parameter that determines the 'granularity' of the thalweg network. Using CUDEM tiles along the Georgia coast as an example, thalwegs extracted with three different flow accumulation thresholds are compared in Fig. 5. For the purpose of compound flood simulation, a threshold of 10^7 is too large because it only retains major rivers but not smaller tributaries that are also important in floodwater routing (Fig. 5a); on the other hand, a threshold of 10^5 includes too many small streams that requires an excessive amount of mesh elements to resolve (Fig. 5c). In this study, we applied a threshold of 10^6 to the whole domain of US East and Gulf Coasts to obtain a reasonably accurate representation of the river network (Fig. 5b). We do not attempt to give a recommended scope of the accumulation threshold because it is affected

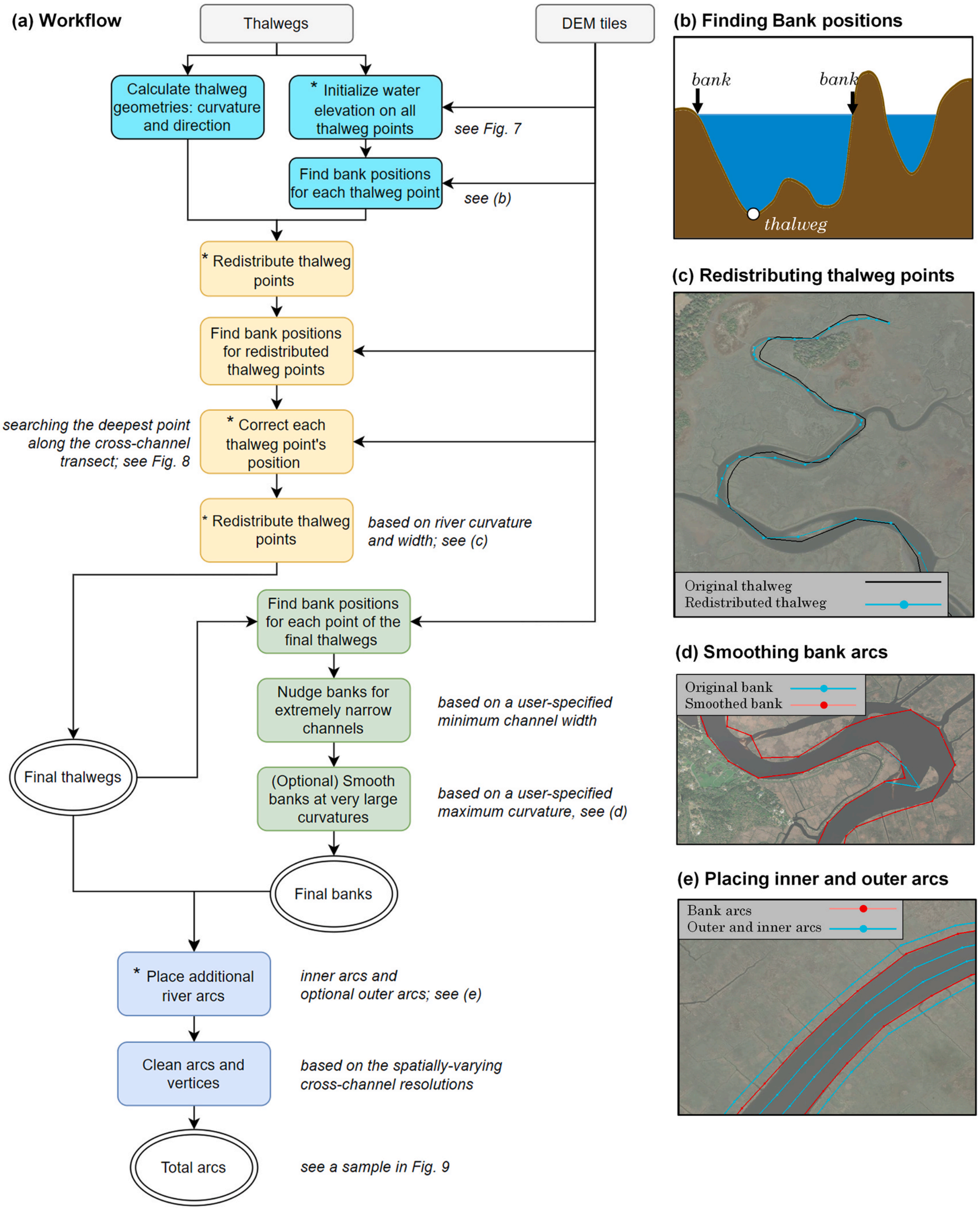

(a) Workflow
Thalwegs
DEM tiles
Calculate thalweg geometries: curvature and direction
* Initialize water elevation on all thalweg points
see Fig. 7
Find bank positions for each thalweg point
see (b)
* Redistribute thalweg points
Find bank positions for redistributed thalweg points
searching the deepest point along the cross-channel transect; see Fig. 8
* Correct each thalweg point's position
* Redistribute thalweg points
based on river curvature and width; see (c)
Find bank positions for each point of the final thalwegs
Nudge banks for extremely narrow channels
based on a user-specified minimum channel width
Final thalwegs
(Optional) Smooth banks at very large curvatures
based on a user-specified maximum curvature, see (d)
Final banks
* Place additional river arcs
inner arcs and optional outer arcs; see (e)
Clean arcs and vertices
based on the spatially-varying cross-channel resolutions
Total arcs
see a sample in Fig. 9
(b) Finding Bank positions
bank
bank
thalweg
(c) Redistributing thalweg points
Original thalweg
Redistributed thalweg
(d) Smoothing bank arcs
Original bank
Smoothed bank
(e) Placing inner and outer arcs
Bank arcs
Outer and inner arcs

Fig. 6. Workflow of RiverMapper's core routine, 'make_river_map': (a) flow chart, which can be viewed as an expanded version of 'Component 2' in Fig. 1, with '*' denoting important steps that will be explained in more details; (b) finding bank positions for each thalweg point; (c) redistributing thalweg points so that the point spacing decreases as the channel bends or narrows. (d) smoothing bank arcs to remove large bank curvatures caused by small-scale noises, for example a sandbar; (e) placing inner and outer arcs based on bank arc positions.

by (1) different configurations of DEM tiles; (2) the granularity required for a particular study or application; and (3) diverse topographical features. For instance, mountainous areas may exhibit narrower river channels compared to coastal plains, necessitating a smaller threshold if other conditions and requirements are similar. As a result, identifying the ideal threshold requires a case-by-case approach, incorporating both trial-and-error iterations and visual evaluations.

The run time to process one augmented DEM tile with 73008 × 73008 cells (~2.3 gigabytes, augmented with its 8 neighbors) is about half an hour. The processing is parallelized by distributing DEM tiles across multiple compute cores, allowing each core to handle more than one tile if necessary. With the parallelization, the run time for the whole domain of US East and Gulf Coasts is about 6 h using 20 computing nodes on W&M's Vortex cluster. Even though each node has twelve cores, we could only use two cores per node as the large tile size requires large memory. Obviously, using more cores can significantly reduce the time given the nearly perfect parallel scaling.

2.2. *Explicit 2D river representation by 'RiverMapper'*

RiverMapper takes existing 1D river segments and DEM tiles as inputs and generates polylines that constitute a 2D representation of the rivers in the watershed. It is worth noting that the input river segments are not restricted to the thalwegs generated in the previous step (Section

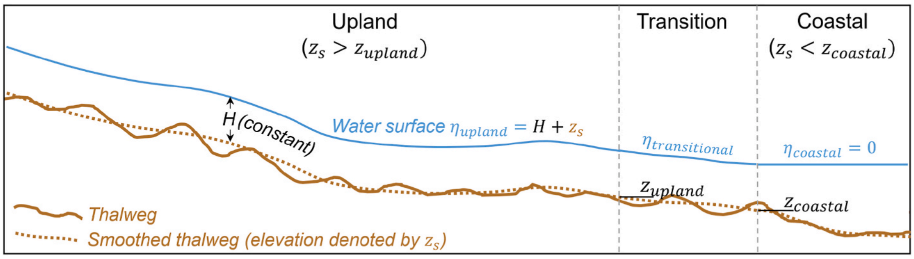


Fig. 7. Illustration of the approximated water surface elevation, η, in three zones. The ground elevation along the smoothed thalweg is denoted by z_s. A constant water depth H is assumed on top of the smoothed thalweg in the upland zone, and a constant water elevation is assumed in the coastal zone. The water level in the transitional zone is a weighted average between the upland value and the coastal value based on the relative position of z_s in the range $[z_{coastal}, z_{upland}]$ according to Eq. (1).

2.1). Instead, any reasonable approximation of the river thalwegs can be used as input, such as the 1D channel network from an existing hydrological model or manually drawn polylines for local mesh improvements. RiverMapper can run either in serial mode or in parallel mode with an optional parallel driver.

2.2.1. *Workflow*

The procedure of RiverMapper's core routine ('make_river_map') is illustrated in Fig. 6a, with each phase colored differently. In the preparation phase, the geometries (curvature and direction) of the input thalweg segments and the initial water elevation on all thalweg points are calculated. In addition, a pair of bank points are located for each thalweg point (Fig. 6b), which provides river width information for the next step. In the second phase, the information of river curvature and width obtained in the first phase is used to redistribute the thalweg points (Fig. 6c), and the bank positions are updated based on the new thalweg points. Additionally, this phase involves correcting thalweg positions and recording the cross-channel resolution at each thalweg point. Upon completion of this phase, the final corrected thalwegs are obtained, which are useful as a standalone product and are also utilized in subsequent steps. In the third phase, the bank positions are updated again, which is necessary because the thalwegs have been modified. Depending on user specifications, this phase may also apply final touch-ups on bank positions (e.g., Fig. 6d). These optional edits can help generate a more visually appealing mesh but may not have a direct impact on model accuracy. The fourth phase involves placing additional river polylines based on the final bank positions (Fig. 6e) and cleaning up the geometry to avoid crowded polyline vertices. The spatially varying cross-channel resolution is utilized to determine the acceptable level of crowding among vertices. Finally, the output that contains all polylines for guiding the river meshing is produced.

The remaining part of this sub-section provides further details on the crucial steps required to achieve high accuracy and efficiency. For additional information, readers can refer to the online documentation available in the GitHub repository (see Software and data availability).

2.2.1.1. Efficient query of ground elevation. During the execution of the tool, the locations of thalweg points and bank points are frequently calculated and adjusted. On a continental scale, this can involve millions of points and thousands of DEM tiles. Moreover, the tiles can be from one or more products, thus having different spatial coverages and resolutions. Consequently, an efficient method for querying river points' z values from DEM tiles is essential for the overall computational efficiency. In this tool, the nearest neighbor interpolation is implemented by directly computing the corresponding 2D indices of the points in multiple DEM tiles. A more precise interpolation is not required based on our experience, because the primary DEM sources typically have a resolution of a few meters, which is fine enough for delineating rivers. Secondly, since a specific thalweg typically resides within a limited number of DEM tiles, grouping thalwegs by their parent tiles is preferable for large applications. A parallel driver (see Section 2.2.2) is provided to automatically generate an optimized grouping, where each group consists of a small subset of thalwegs and DEM tiles. These groups are then distributed across multiple compute cores, which accelerates the search process and reduces memory consumption. Furthermore, the code utilizes NumPy (Harris et al., 2020) vectorized operations when dealing with large vectors and arrays. These operations are significantly faster than Python's for-loops and offer performance comparable to that of C or Fortran programs.

2.2.1.2. Detecting bank positions. Bank positions are calculated multiple times in the workflow (Fig. 6a) because multiple steps depend on the estimated channel width, which must be updated following any changes in the thalwegs. Given a thalweg point, two bank edges (where the mean water depth is 0; Fig. 6b) on both sides of the thalweg are searched in the cross-channel direction. The search process is straightforward but determining the water depth requires knowing the time-varying free water surface, which is not trivial. Although the free surface can be assumed to be the same as the local mean sea level for nearshore regions, this approximation is no longer valid for inland rivers where the riverbed is higher than the local sea level. A number of options were considered to improve the approximation. One potential solution is to interpolate the surface elevations from the river stage observations. Here we propose a simpler alternative that is more straightforward to implement. As illustrated in Fig. 7, the domain is divided into three parts (upland region, transitional region, and coastal region) by two representative ground elevation values (z_{upland} and $z_{coastal}$). In the coastal zone, a constant water level (e.g., the mean sea level) is assumed. In the upland zone, a constant water depth is assumed. Since the free water surface should be smoother than the underlying riverbed, the ground elevation along each thalweg is smoothed with a 100-m moving window filter. Then, the water surface is found by adding the assumed constant water depth to the *smoothed* thalweg elevation. In the transitional zone, the water level varies linearly from the coastal value to the upland value as:

$$\eta_{transitional} = \frac{z_s - z_{coastal}}{z_{upland} - z_{coastal}}\eta_{upland} + \frac{z_{upland} - z_s}{z_{upland} - z_{coastal}}\eta_{coastal} \quad (1)$$

with the symbols defined in Fig. 7. In this study, we choose $z_{upland} = 3$ m (NAVD 88) and $z_{coastal} = 0$ m (NAVD 88).

The choice of the constant water depth (H) in the upland region (Fig. 7) is somewhat arbitrary. From our experience, a small water depth (e.g., 1.0 m) is preferred as it usually results in a clean delineation of river channels. The resultant bank positions from this algorithm are generally satisfactory; an example is shown in Fig. 9. Additionally, the margins of errors in the bank delineation can be partially accounted for by inner and outer arcs as described in Section 2.2.1.5.

2.2.1.3. Optimizing thalweg points distribution. The spatial resolution of the extracted thalwegs depends on the DEM sources, and the typical spacing between two adjacent thalweg points is 1–5 m. This resolution is too fine for resolving rivers in the along-channel direction. In watershed rivers, we recommend using quasi-1D elements, which are anisotropic elements longer in the along-channel direction and shorter in the cross-channel direction. Compared to the size-function based approach, this quasi-1D river representation significantly reduces the final mesh size while maintaining high resolution especially in the cross-channel direction regardless of the river width. The shape of the elements is

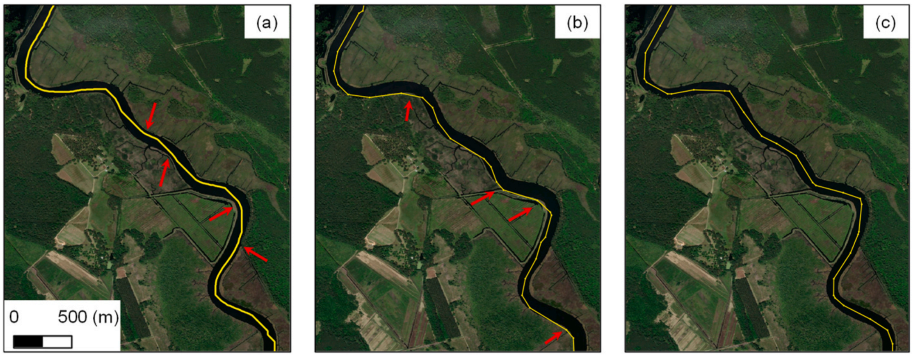


Fig. 8. Examples of thalweg positions from different products: (a) extracted thalwegs from DEMs, with red arrows indicating locations where the thalweg is too close to either bank; (b) 1D river segments from the National Water Model v2.1, with red arrows indicating locations where the river segment is slightly outside the channel; (c) corrected thalwegs. The base map is provided by Esri.

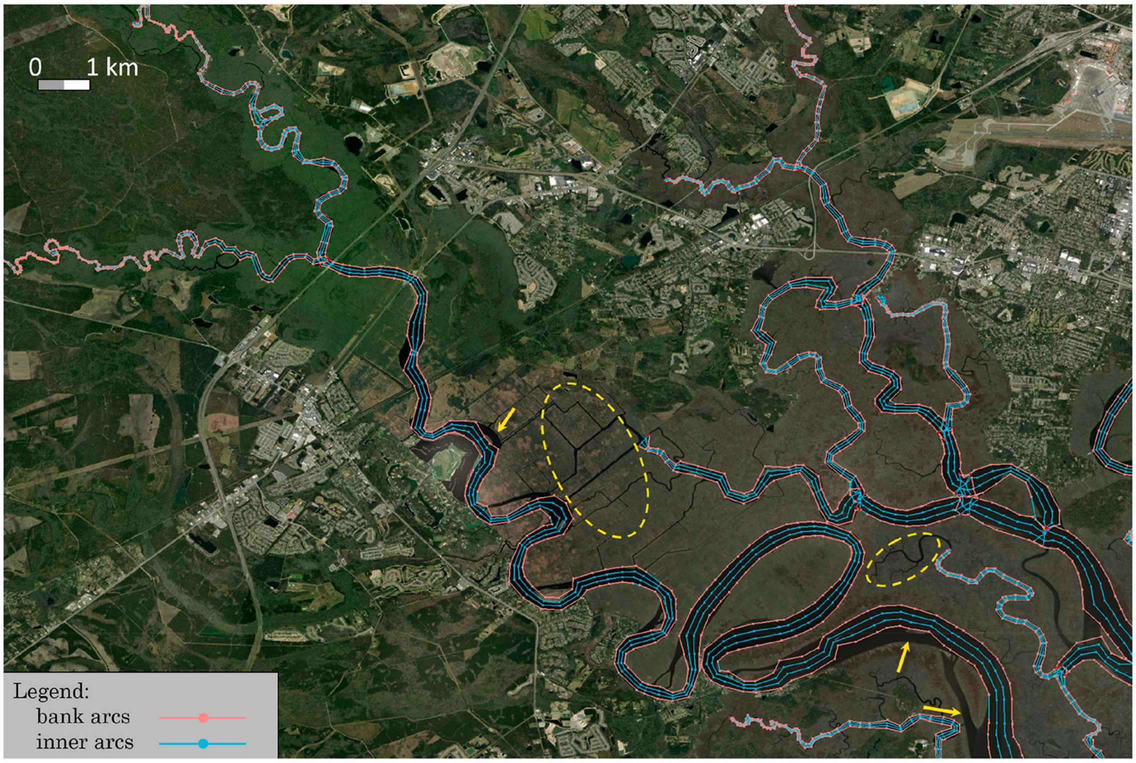


Fig. 9. Sample output of RiverMapper containing bank arcs and inner arcs used for guiding river meshing in Georgia, US. Some bifurcations formed by river islands (e.g., those highlighted by yellow arrows) and minor rivers (e.g., those highlighted by yellow circles) are not resolved because the level of details in the extracted thalweg network is restricted by the parameterization of Step 1 (cf. the effect of flow accumulation threshold in Fig. 5). The base map is provided by Esri.

controlled by a user-specified ratio between the along-thalweg point spacing to the cross-channel resolution. In addition, the along-channel resolution should also account for river curvatures, because more points are needed at sharp river bends than along straight channels to fit the river geometry. With these considerations, the script redistributes the thalweg points based on the along-channel resolution *dl* calculated at each thalweg-points as:

$$dl = min\left(\frac{wr_1}{n}, \frac{r_2}{\kappa}\right), \tag{2}$$

where w is river width, n is the user-specified number of cross-channel segments (which can be constant or dependent on river width), so w/n is the cross-channel resolution; r_1 is the user-prescribed ratio between the along-thalweg point spacing and the cross-channel resolution; r_2 is the user-prescribed ratio between the along-thalweg point spacing and the radius of the thalweg's curve, which is the reciprocal of the curvature κ. In practice, we typically use a value of 4.0–5.0 for r_1, and 0.4 for r_2. Before calculating the curvature κ at each thalweg point, the script also automatically smoothes the thalweg in the horizontal dimension with a 30-point moving average. This smoothing process eliminates the impact of minor zigzags (e.g., on a scale of a few meters) on the computed river curvature. Note that the horizontally smoothed thalweg is only an intermediate product for calculating curvatures, the point redistribution is still processed on the original thalweg.

2.2.1.4. Correcting thalweg positions. The thalwegs, whether extracted from DEMs or imported from other sources, may exhibit deviations from

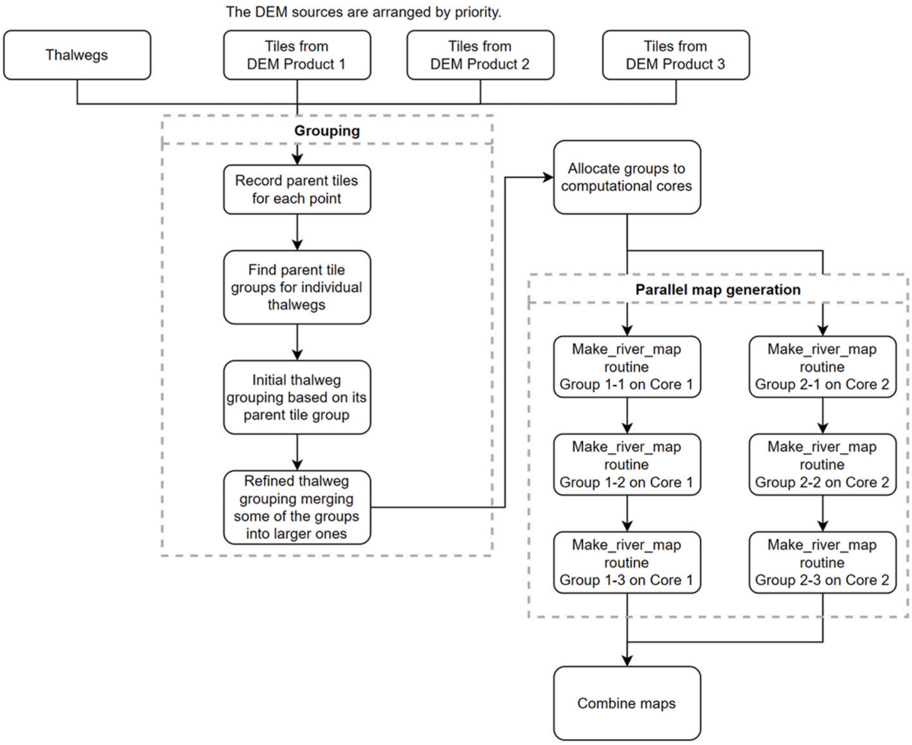

The DEM sources are arranged by priority.
Thalwegs
Tiles from DEM Product 1
Tiles from DEM Product 2
Tiles from DEM Product 3
Grouping
Record parent tiles for each point
Find parent tile groups for individual thalwegs
Initial thalweg grouping based on its parent tile group
Refined thalweg grouping merging some of the groups into larger ones
Allocate groups to computational cores
Parallel map generation
Make_river_map routine Group 1-1 on Core 1
Make_river_map routine Group 1-2 on Core 1
Make_river_map routine Group 1-3 on Core 1
Make_river_map routine Group 2-1 on Core 2
Make_river_map routine Group 2-2 on Core 2
Make_river_map routine Group 2-3 on Core 2
Combine maps

Fig. 10. Workflow of the parallel driver for large applications.

their true positions. For instance, some thalwegs produced in Step 1 have a tendency to cling to one side of the channel along a sinuous river stretch (Fig. 8a); and in the case of the National Water Model[8] (NWM), some segments can even be slightly outside the channel shown in the satellite imagery (Fig. 8b). As the quality of the generated river map relies on the thalwegs' position (which should ideally be situated between the riverbanks), a thalweg correction is applied prior to determining the final bank positions. This is done by relocating each thalweg point to the deepest location along the cross-channel transect (Fig. 8c). In practice, we typically expand the transect (i.e., the search range) to three times the estimated channel width to accommodate potential inaccuracies in the initial thalweg position.

2.2.1.5. Specifying inner and outer arc locations. After determining the positions of both riverbanks, users have the option to add one or more arcs inside and/or outside the channel (as illustrated in Fig. 6e). Typically, the number of inner arcs should be set to increase with river widths, with a minimum of one inner arc placed in very narrow channels. Since the bank points always come in pairs, specifying the relative position of inner arcs between banks is straightforward; Fig. 9 provides an example where inner arcs evenly divide the cross-channel transect. On the other hand, implementing a pair of outer arcs is advisable due to two advantages: (i) it facilitates a smoother transition between the quasi-1D river elements (which are elongated in the along-channel direction) and the uniform watershed elements (which closely resemble equilateral triangles); and (ii) it accommodates the uncertainty in the estimated water levels and allows for a broader channel during high-flow events.

2.2.2. Parallel driver

An optional parallel driver is provided for large applications. The driver first divides thalwegs into groups based on their parent DEM tiles (note that tiles from heterogeneous DEM products of different resolutions are allowed in RiverMapper), and then automatically allocates an optimized workload to the worker routine 'make_river_map' on each compute core. The workflow is illustrated in Fig. 10.

2.2.3. Execution

For a small- or medium-size application, e.g., with a spatial coverage of one or two states, the serial function 'make_river_map' is efficient enough; the runtime is on the order of a few minutes. The function call looks like:

```
make_river_map(tif_fnames, thalweg_shp_fname, output_dir)
```

The first input argument 'tif_fnames' is a list of names of DEM files in *.tif format; the second input argument is the name of the shapefile that contains all thalwegs as 'LineStrings'.

For a large application on the continental scale (cf. the case study in Section 3), it is preferable to call the serial function by the parallel driver. A sample script is provided in the GitHub repository (see Software and data availability), which can be executed as:

```
mpirun -n 20 ./sample_parallel.py
```

The function call to the parallel driver looks like:

```
river_map_mpi_driver(dems_json_file, thalweg_shp_fname, output_dir)
```

The input argument 'dems_json_file' simplifies the process of specifying numerous tiles from multiple DEM products by using an input file instead of a list. The runtime for the US East and Gulf Coasts domain is about 1.5 h with one computation node (20 cores) on W&M's Bora cluster; in comparison, the serial mode takes roughly 16 h.

The default output format is Esri Shapefile and the main file is named 'total_arcs.shp'. It assumes a "LineString" geometry type and contains all polylines for guiding river meshing. The shapefile should be readable by common meshing tools such as SMS. It is worth noting that the mesh generator is not limited to the one used in this study. The only requirement is that the mesh generation process must reproduce the original polylines without adding new vertices or removing existing

[8] URL: https://water.noaa.gov/about/nwm, last accessed: Apr 6, 2023.

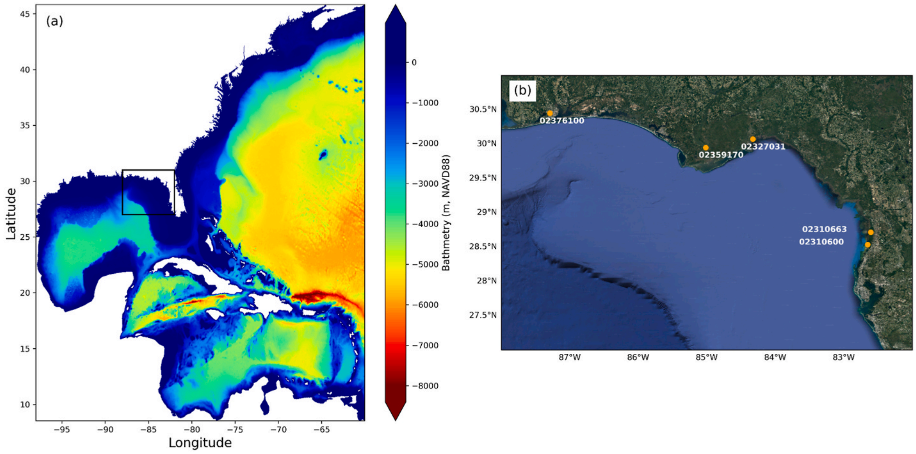


Fig. 11. (a) Domain extent, with (b) zoom-in near northwestern Florida showing the USGS gauge locations used in this section.

vertices; in other words, the final mesh must preserve the original line segments as element sides.

2.3. Mesh generation of the unstructured grids

The high-resolution polylines generated in Section 2.2 are directly fed into SMS to generate the final mesh. Note that the tool used here only creates polylines (arcs) for the watershed rivers, and the users are responsible for generating the SMS 'maps' for other parts of the domain, which have simpler geometry/constraints and are thus easier to create. Also, defense structures available in shapefile format (e.g., from the National Levee Database[9]) are imported into SMS. Various maps are then merged inside SMS. Due to the sheer number of arcs (~350,000) and vertices (~1.14 million) involved, the meshing is so heavily constrained that in our experience, few mesh generators can handle it.

The current tool is capable of generating UGs that capture rivers in extremely high resolution, with the only limitation being the resolution of the underlying DEM. With the availability of high-quality DEMs in most US coastal areas, we are able to resolve small rivers of ~10 m wide with 2–3 cross-channel elements (cf. Fig. 15c). In general, the accuracy of the river channels depicted in the mesh is quite satisfactory when visually compared to satellite imagery (cf. Fig. 9). Obviously, the quality of the DEM is the primary factor influencing this level of fidelity.

Due to the highly constrained triangulation, the generated mesh inevitably includes some skewed elements that necessitate a robust model, such as SCHISM, to manage effectively. Courtesy of the quasi-1D river representation, the mesh size is moderate (with ~2.9 million nodes) for the domain of US East and Gulf Coasts. In contrast, methods that use the size function approach often result in an excessive amount of resolution in small channels, leading to mesh sizes that are 5–10 times larger. However, even with larger mesh sizes, these methods still have difficulty in providing adequate cross-channel resolution for narrow channels that are a few meters wide. Therefore, the current method should lead to significant improvements in both the efficiency and accuracy of the model simulation.

3. Application to a continent-scale study of compound flooding

The mesh generated by the new tool is ideal for compound flooding studies because it explicitly resolves river channels and other important features in the watersheds. Previously, we attempted the same goal by extensive manual editing (cf. Fig. 16), which is time-consuming and labor-intensive. Moreover, it is prone to errors and can easily misrepresent narrow channels. Severe consequences of such errors include: 1) impeded river flow and tidal movements; and 2) inaccurately elevated surface levels due to flow obstruction. The results of this case study demonstrate that the newly developed tool represents a major step forward in simulating watershed flows in the context of compound flooding.

3.1. Creek-to-ocean model

The simulation is carried out using an open-source, cross-scale, 'creek-to-ocean' 3D model, SCHISM (Zhang et al., 2016; schism.wiki), which applies a hybrid finite-element/finite-volume method to solve the Reynolds averaged Navier-Stokes equation together with tracer transport. A major feature of SCHISM that is crucial for the current study is its implicit time stepping method that guarantees numerical stability (free of CFL restriction) even with skew elements and very fine resolution. Its robustness is also beneficial to the automatic generation of on-demand forecast systems, for example by OPENCoastS (Oliveira et al., 2020, 2021). Another major feature that has great implications for the compound flood study is that SCHISM does not smooth bathymetry but instead interpolates directly from the original DEMs using linear interpolation. This makes the model much more sensitive and responsive to DEM quality than other models that rely on bathymetry smoothing/manipulation. The detrimental effects of bathymetry smoothing have been demonstrated in Ye et al. (2018) and Cai et al. (2020) for physical and biological processes respectively. We remark that the commonly used nearest-point interpolation method with channel 'greedy' approach (i.e., using the maximum depth from the surrounding DEM raster cells) would lead to discontinuous behavior as the mesh is revised. The linear interpolation method used in our model in all pre- and post-processing is C^1 continuous, or in other words, the function approximated by the elements has a smooth gradient across element boundaries. However, care should be taken when extracting the model results at gauge locations via linear interpolation, because SCHISM does not allow partial wetting and drying in a cell. To avoid interpolating from a dry node, the mesh design needs to ensure that all nodes of the gauge's parent element are within the channel.

The model domain and setup are similar to Huang et al. (2021) and to STOFS-3D-Atlantic. The domain (Fig. 11) includes the entire US East and Gulf Coasts in high resolution, with the land boundary located at 10 m above MSL, where the river flows calculated from NWM v2.1 are injected into our domain. The offshore boundary is located at 60°W. The simulation lasts 61 days starting from Aug 17, 2018, and thus covers both Hurricane Florence (2018) and Hurricane Michael (2018). The vertical datum used is NAVD88. The model uses a non-split time step of 150 s with an implicitness factor of 1.0 (i.e., fully implicit), and modified

[9] URL: https://levees.sec.usace.army.mil/#/, last accessed: Apr 6, 2023.

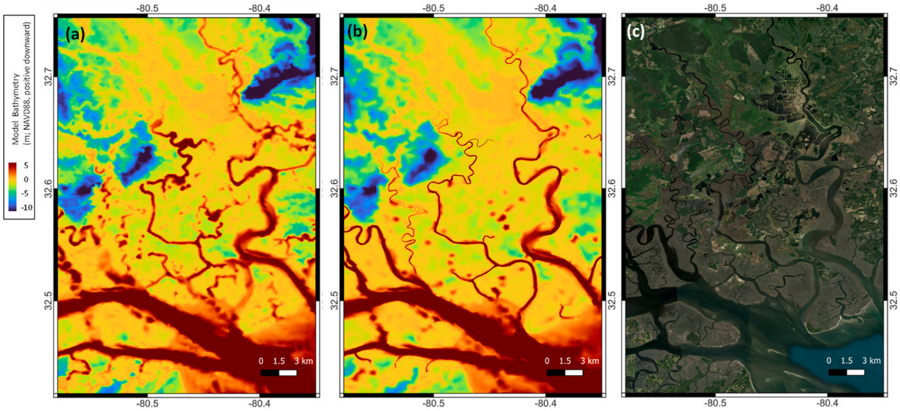


Fig. 12. Channel representation of the model meshes in a coastal region of South Carolina, US: (a) model bathymetry of the old mesh, note the broken channels in small rivers; (b) model bathymetry of the new mesh; (c) satellite imagery (provided by Esri).

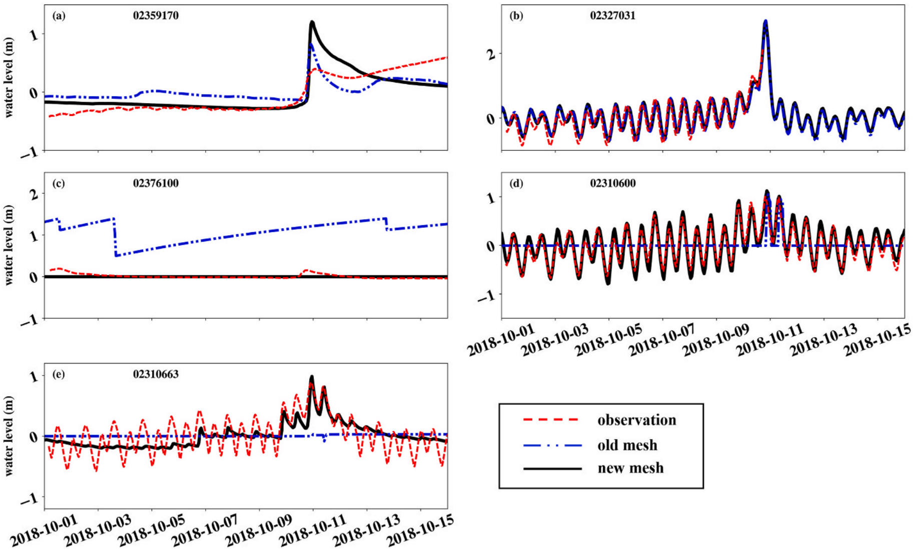


Fig. 13. Comparison of water level (cf. Fig. 10 for locations) during Hurricane Michael (2018) at the following USGS gauges in western Florida: (a) 02359170, Apalachicola River; (b) 02327031, Spring Creek; (c) 02376100, Bayou Marcus Creek; (d) 02310600, Gulf of Mexico near Bayport, FL; (e) 02310663, Chassahowitzka River. The two model curves are nearly identical in (b).

Mellor-Yamada scheme (*k-kl*; Umlauf and Burchard 2003) as the turbulence closure. For tracer transport, the 3rd-order WENO scheme (Ye et al., 2019) is used for eddying regime and upwind scheme is used for the watersheds. The biharmonic viscosity is used for the horizontal mixing, augmented by a slope-dependent Shapiro filter (Huang et al., 2021). The 3D model is initialized and boundary-forced by the Hybrid Coordinate Ocean Model (HYCOM[10]) as the non-tidal component. For the tidal component of the elevation and velocity, the global tidal database of FES2014 (Lyard et al., 2021) is used. The atmospheric forcing source consists of 1) NOAA's High-Resolution Rapid Refresh (HRRR; Benjamin et al., 2016) numerical weather prediction modeling system, with a 3 km and hourly resolution, for coastal ocean and watersheds; and 2) ERA5 (Hersbach et al., 2020), with a 30 km and hourly resolution, for areas not covered by HRRR. The high-resolution precipitation from HRRR is used in the model to simulate the overland runoff. Since we start from a fully dynamic state (from HYCOM), the spin-up period is only several days. Our experience suggests that the watershed flows require ~10 days to equilibrate. Therefore, the results from the first 20 days are excluded in the analysis. Using 3360 cores of Texas Advanced Computing Center's Frontera cluster, the model runs ~100 times faster than real time.

In addition to the new mesh generated in Section 2.3, for comparison purposes we also include the results from an older mesh, which used the 1D river network from NWM to guide the watershed river meshing. Higher resolution was specified at the thalwegs to enhance the detail near the channels. However, the representation of the lateral direction of the channel was insufficient, as thalwegs are only 1D features. In addition, considerable manual editing was performed to improve channel connectivity. As shown in Fig. 12a, the resultant mesh still has many broken rivers due to imprecise channel specification. The old mesh has 2.7 million nodes; in contrast, the new mesh is slightly larger with 2.9 million nodes, but it offers a significantly improved channel representation (Fig. 12b) as compared to the satellite imagery (Fig. 12c).

3.2. Hurricane study

In 2018, the STOFS-3D-Atlantic domain (US East and Gulf Coasts) experienced two major hurricanes (Florence and Michael) in quick succession. Hurricane Florence was initially a Category 4 storm, which downgraded to Category 1 when making landfall at Wrightsville Beach, North Carolina on September 14, 2018. The hurricane was characterized

[10] URL: https://www.hycom.org/, last accessed: Apr 6, 2023.

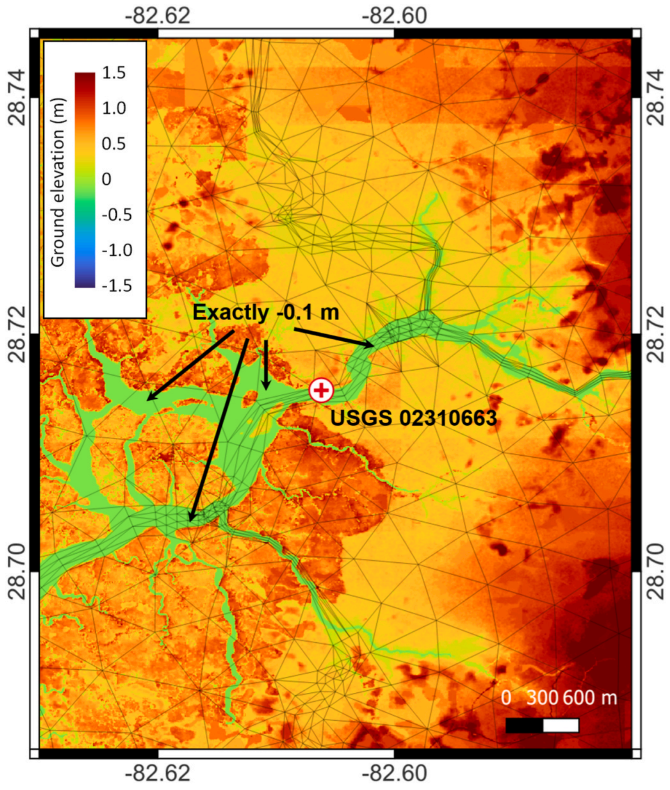


Fig. 14. CUDEM around USGS gauge 02310663 (Chassahowitzka River Near Chassahowitzka, FL), overlaid by the computational mesh. A dubious value of exactly "-0.1 m" is present in the river channels.

Huang et al. (2021) and Ye et al. (2021). Multiple US agencies, including NOAA, US Geological Survey (USGS), and US Army Corps of Engineers (USACE), maintain many water level gauges along the US East and Gulf Coasts, from coastal regions into the watershed, which can be used to assess the model skill. Since the coastal part of the new mesh is identical to that used in Huang et al. (2021), we skip the comparisons at NOAA stations and focus on USGS gauges and 'High Water Marks' (HWMs) here.

Accurate simulation at USGS gauges is very challenging especially in small rivers, as it requires high-quality DEMs and precise knowledge of the gauge locations. While LiDAR-based DEMs are highly accurate for the watershed topography, the bathymetry in some rivers can be questionable. This is because LiDAR captures only the water surface elevation, and algorithms used to convert surface elevation to bottom elevation have substantial uncertainties without actual bathymetric survey data. Only recently did the ground penetrating LiDAR become available, which will greatly reduce the DEM uncertainties in the future. Meanwhile, the DEMs used in this paper were derived from older surveys and are thus subject to this issue. In addition, the mesh may also need to have higher resolution locally in small rivers via selection of the meshing parameters mentioned in Section 2. With the new technique developed in this paper, we can arbitrarily increase the mesh resolution locally to capture features, subject to the quality of the DEMs used.

Given the caveats mentioned above, we have achieved some success in capturing the total water elevation in the watershed at several USGS gauges in the Florida Panhandle (Fig. 13a–c) and western Florida (Fig. 13d and e). The locally very high resolution employed to resolve small rivers proved to be quite effective in accurately capturing the tidal propagation into those rivers. Remaining issues are found in the areas near some gauges where the DEM quality is questionable. An example is illustrated in Fig. 13e with the associated DEM shown in Fig. 14. The gauge is located in the upstream of a narrow river and the DEM suggests

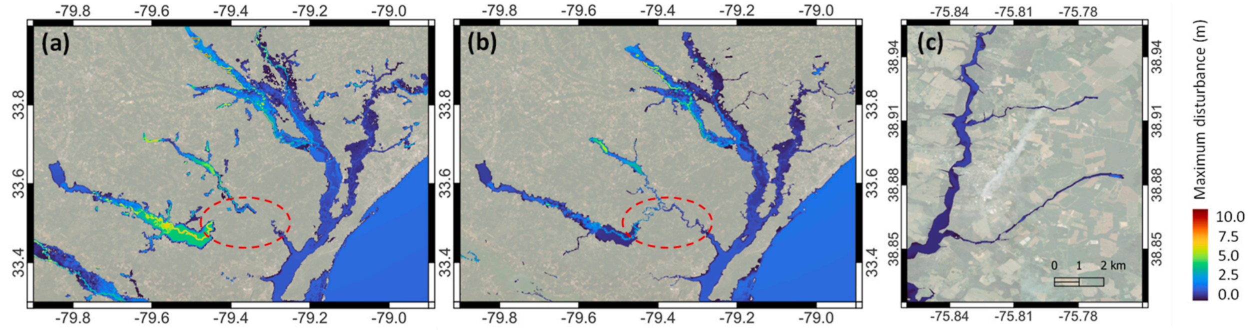


Fig. 15. Simulated channel connectivity during Hurricane Michael (2018): (a) simulated maximum disturbance on the old mesh in South Carolina, US, with transparent color indicating dry areas; (b) same as (a) but on the new mesh, showing the improved connectivity in the highlighted region; (c): same as (b) but showing a highly constricted channel in the Chesapeake Bay with unimpeded floodwater routing even at its narrowest point which is about 10 m wide. Precipitation is excluded to clearly show the fluvial flood. The base maps are provided by Esri.

by its slow movement and excessive precipitation, leading to devastating compound flooding. The estimated total damage was $24 billion according to NOAA[11]. Hurricane Michael, the first Category 5 hurricane to strike the contiguous United States since Hurricane Andrew in 1992, made landfall on October 10, 2018, at Mexico Beach, Florida. With peak winds reaching 261 km/h, this powerful storm devastated homes in Mexico Beach and Panama City, Florida. Its total damage was estimated to be $25 billion according to the same NOAA source.

The focus of this study is the total water level, while assessment of 3D variables (salinity, water temperature, velocity) have been presented in

very shallow depths along the channel. A constant bed elevation of -0.1 m was found in the river and the inlet that connects the river to the shelf (Fig. 14), which is highly dubious. There may be additional uncertainty in the exact gauge location with respect to the channel. Therefore, even though the mesh well resolves the 'channel' as depicted in the DEM, the simulated tides are more attenuated than the observation (Fig. 13e).

The most striking differences between the old and new mesh are seen in the simulated flow routing in the watershed. For compound flooding studies, it is preferable to visualize the flood using the concept of 'disturbance', which is defined as the water surface elevation in the 'wet' coastal portion of the domain and total water depth in the 'dry' (hydrologic) regime (Huang et al., 2021):

$$D = \begin{cases} \eta, \text{if } h \geq 0, \text{i.e., wet regime} \\ \eta + h, \text{if } h < 0, \text{i.e., dry regime} \end{cases} \tag{3}$$

[11] URL: https://www.ncei.noaa.gov/access/billions/events.pdf, last accessed Apr 6, 2023.

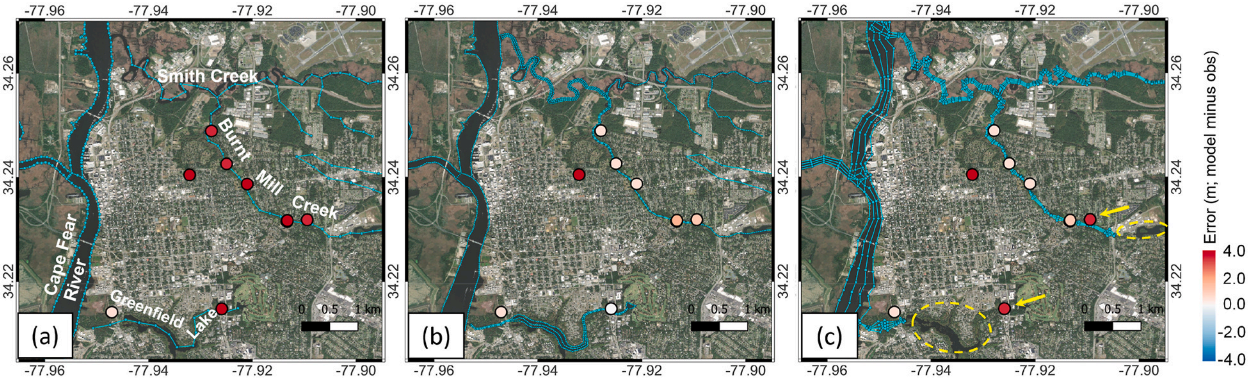


Fig. 16. Model error of water surface elevation on High Water Marks (denoted by colored dots, with red indicating over-prediction) near Wilmington, North Carolina, US, during Hurricane Florence 2018. The 'feature arcs' (i.e., polylines used to guide meshing) are shown as cyan lines. The results are derived from three different meshes in which small rivers are represented by (a) one row of polylines (1D channel representation); (b) manually edited polylines, forming 2D channel representation; (c) automatically generated polylines, forming a 2D representation for designated river networks. In (c), large errors (as compared to (b)) outside the designated river network are highlighted.

where η is the water surface elevation and h is the positive downward bathymetry. 'Disturbance' essentially measures the deviation of water level from the initial (undisturbed non-storm) condition. Despite a great deal of manual editing in the old mesh to align the mesh to the river channels, there are many places where the flow is interrupted due to inadequate channel resolution especially in the cross-channel direction (Fig. 15a). As a result, the blocked flow leads to over-estimation of lateral inundation upriver (Fig. 15a vs Fig. 15b). This issue has been largely fixed with the new mesh, as illustrated in Fig. 15b. In addition, Fig. 15c reveals that the new mesh is able to well resolve narrow stretches (~10 m wide) of channels, thus allowing unimpeded flow. The accurate conveyance of river flow from the watershed to the coastal region, which essentially establishes the boundary condition for the coastal regime, is a crucial prerequisite for the precise simulation of compound flooding.

A better flood conveyance in the mesh also helps reduce the error of the simulated water surface elevation at HWMs. As illustrated in a previous hindcast study of Hurricane Florence (Ye et al., 2021), using thalwegs alone as guiding polylines at the meshing stage did not guarantee channel connectivity of the resultant mesh. Consequently, floodwater tended to accumulate in the upstream region due to the lack of draining conduits, resulting in significant overestimations of water surface elevation in the watershed (Fig. 16a). As a temporary remedy, we manually edited some of the problematic river channels to apply a 2D channel representation, which effectively reduced the model error (Fig. 16b). However, this was only done for limited locations because the amount of manual labor involved can be formidable for large-scale applications like STOFS-3D-Atlantic. In contrast, the newly introduced tools allow achieving mesh quality similar to the manually edited one (compare Fig. 16b and c) but in a much more efficient manner for the whole domain. Note that the model errors in Fig. 16c are comparable to those in Fig. 16b on most data points except for the two points highlighted in Fig. 16c. This discrepancy is related to the level of detail in the river network, which is prescribed by the user during the thalweg extraction process (Section 2.1). Some small rivers or lakes (circled in Fig. 16c) may be neglected if they are outside the coverage of the extracted river network, hence the large errors near them.

Similar to the issue with HWMs, the use of the old mesh also caused unrealistic high waters near the land boundary where the NWM flows were injected (Fig. 17a). In addition to the channel blockage problem mentioned earlier, the issue is partially attributed to the momentum-less injection of freshwater. Depending on the magnitude of the injected flow, this approach can temporarily lead to a high water elevation at the injection point before the pressure gradient drives the water away. The issue has been largely resolved with better channel representations

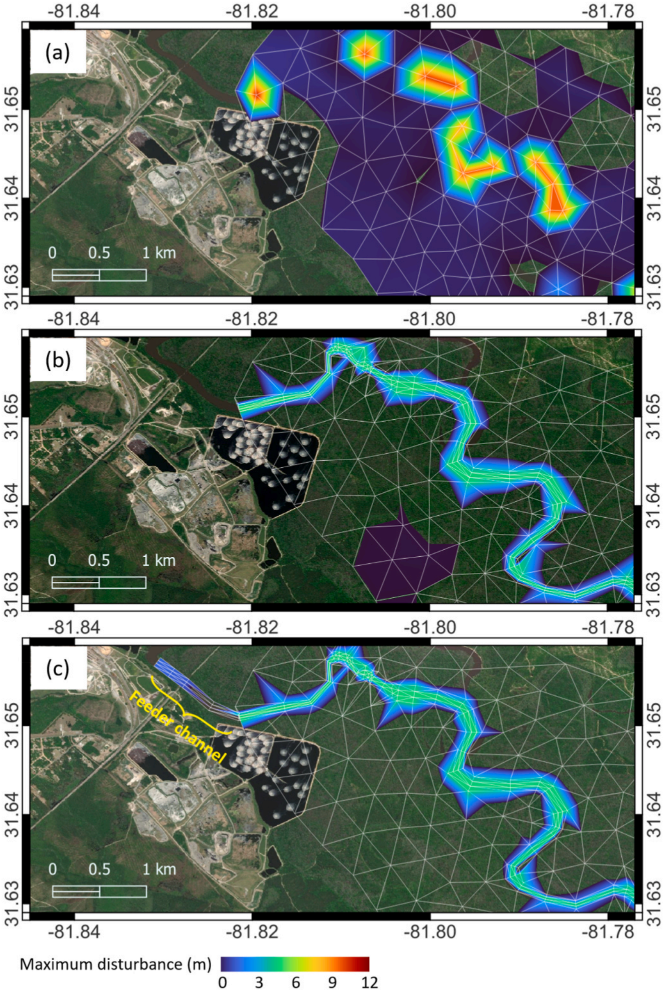


Fig. 17. Simulated flow around a typical freshwater injection point near the land boundary: (a) maximum disturbance during Hurricane Michael (2018) on a mesh with a 1D channel representation; (b) same as (a) but with the channel resolved by three rows of elements; (c) same as (b) but with an additional pseudo (feeder) channel extending beyond the land boundary. The base map is provided by Esri.

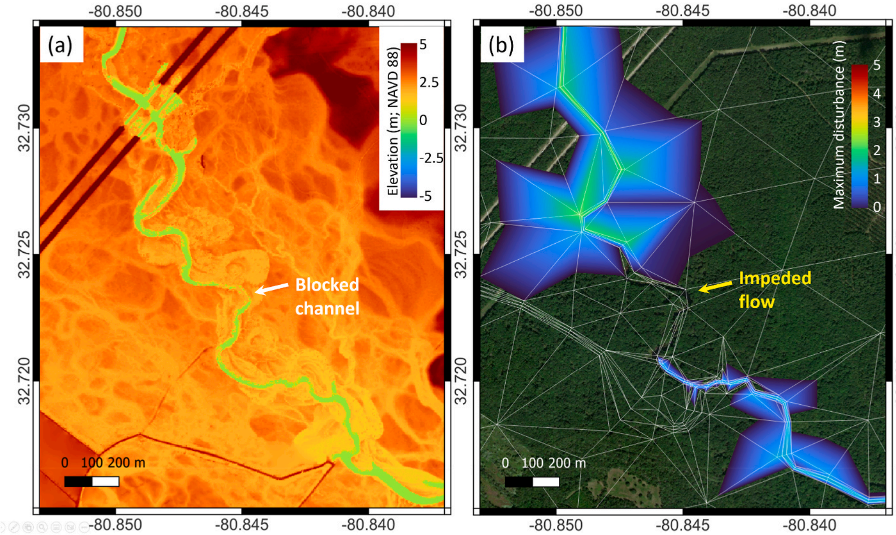


Fig. 18. Blocked channel related to DEM quality: (a) channel blockage as shown in NOAA's CUDEM; the version of the DEM tile is '2019v1'; (b) impeded channel flow illustrated by maximum disturbance overlaid by model mesh, with the transparent color indicating dry area. The base map is provided by Esri.

(Fig. 17b), which facilitates the downstream transfer of floodwater. However, minor unphysical accumulation may still occur under sudden high flow events, which can be problematic if the hydrodynamic model needs to send information back to the hydrological model. To ensure sufficient time and space for the injected flow to reach a dynamic equilibrium, feeder channels can be incorporated during the meshing stage. This is achieved by extending the thalwegs slightly upstream beyond the original land boundary, as highlighted in Fig. 17c. Under this configuration, incoming flows are imposed at the *upstream end* of the feeder channels and the elevations at the *downstream end* are transmitted back to the hydrological model, thereby preventing any undesirable boundary problems.

Most of the remaining issues are related to the DEM quality. To determine the appropriate DEM quality for applying the automated tools, a simple rule-of-thumb is that channels should be discernible by visualization. As an example, the river channel shown in Fig. 18a is clearly visible except for a small, highlighted portion. Although in this case the river arcs are still clean (Fig. 18b), the simulated flow inevitably shows channel blockage because the channel depth is ultimately dependent on the DEM. On the other hand, if most parts of a river are poorly defined in the DEM, the resultant river arcs may be chaotic. As temporary measures, users have the options from RiverMapper to 1) discard the messy river arcs locally, or 2) generate a pseudo channel of a prescribed channel width to allow artificial dredging in the mesh. Of course, these issues can be more effectively addressed with an updated DEM of better quality. After all, the automated tool is specifically designed to reduce the time required for the development cycle from a new bathymetric survey to a new operational forecast.

4. Conclusions

We have successfully developed a parallel Python-based tool to effectively support mesh generation for watersheds. This tool addresses a key challenge in watershed and compound flood modeling by providing an explicit 'guide' in the form of polylines (or equivalently, arcs and vertices) to the mesh generator. To accomplish this, the tool first identifies 1D thalwegs from Digital Elevation Models (DEMs) and subsequently expands the 1D thalweg networks into 2D river representations. A mesh generator (SMS) is then used to triangulate the mesh, with explicit constraints from the generated river polylines (or 'feature arcs'). Other mesh generators can also be utilized, as long as they preserve the river representation. Furthermore, it is possible to combine the current method with size-function-based methods; an example is the ongoing work with OCSMesh,[12] which will be reported in a future publication.

We used this tool to generate a continental-scale mesh for US East and Gulf Coasts that includes watersheds, then used the mesh and an unstructured-grid model (SCHISM) to simulate compound flooding during hurricane events. The model skill in the watershed was greatly improved with the new mesh and the water delivery from the watershed to the coastal region showed uninterrupted flow in most places. Comparisons at USGS gauges in some small rivers confirmed the superior model skill. The resultant accuracy from the meshing tool is critically dependent on the DEM quality especially for river bathymetry, for which the new ground penetrating LiDAR should help. The tool presented in this paper represents a major step forward in watershed modeling and compound flood studies in general.

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.

Data availability

I have shared the link to my code/data in the manuscript.

Acknowledgements

This study was funded by NOAA (grant #NA21NOS0080197). Simulations used in this paper were conducted using the following computational facilities: (1) William & Mary Research Computing (URL: https://www.wm.edu/it/rc); (2) Texas Advanced Computing Center (TACC) at the University of Texas at Austin.

References

Araújo, M.A.V., Mazzolari, A., Trigo-Teixeira, A., 2013. An object oriented mesh generator: application to flooding in the Douro estuary. J. Coast Res. 65 (10065), 642–647.

Bacopoulos, P., Hagen, S.C., 2022. Physics-based sizing functions for ocean-to-estuary shallow-water models. Ocean Model. 176, 102061.

[12] URL: https://github.com/noaa-ocs-modeling/OCSMesh, last accessed: Apr 6, 2023.

Barnes, R., Lehman, C., Mulla, D., 2014. Priority-flood: an optimal depression-filling and watershed-labeling algorithm for digital elevation models. Comput. Geosci. 62, 117–127.

Benjamin, S.G., Weygandt, S.S., Brown, J.M., Hu, M., Alexander, C.R., Smirnova, T.G., Olson, J.B., James, E.P., Dowell, D.C., Grell, G.A., Lin, H., 2016. A North American hourly assimilation and model forecast cycle: the Rapid Refresh. Mon. Weather Rev. 144 (4), 1669–1694.

Bevacqua, E., Maraun, D., Vousdoukas, M.I., Voukouvalas, E., Vrac, M., Mentaschi, L., Widmann, M., 2019. Higher probability of compound flooding from precipitation and storm surge in Europe under anthropogenic climate change. Sci. Adv. 5 (9), eaaw5531.

Bilskie, M.V., Hagen, S.C., 2018. Defining flood zone transitions in low-gradient coastal regions. Geophys. Res. Lett. 45 (6), 2761–2770.

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. JAWRA J. Am. Water Resour. Assoc. 1–24. https://doi.org/10.1111/1752-1688.12887.

Casulli, V., Stelling, G.S., 2011. Semi-implicit subgrid modelling of three-dimensional free-surface flows. Int. J. Numer. Methods Fluid. 67 (4), 441–449.

Conroy, C.J., Kubatko, E.J., West, D.W., 2012. ADMESH: an advanced, automatic unstructured mesh generator for shallow water models. Ocean Dynam. 62, 1503–1517.

Costabile, P., Costanzo, C., Ferraro, D., Macchione, F., Petaccia, G., 2020. Performances of the new HEC-RAS version 5 for 2-D hydrodynamic-based rainfall-runoff simulations at basin scale: comparison with a state-of-the art model. Water 12 (9), 2326.

Del-Rosal-Salido, J., Folgueras, P., Bermudez, M., Ortega-Sanchez, M., Losada, M.A., 2021. Flood management challenges in transitional environments: assessing the effects of sea-level rise on compound flooding in the 21st century. Coast Eng. 167, 103872.

Engwirda, D., 2017. JIGSAW-GEO (1.0): locally orthogonal staggered unstructured grid generation for general circulation modelling on the sphere. Geosci. Model Dev. (GMD) 10 (6), 2117–2140.

Ezer, T., 2018. The increased risk of flooding in Hampton roads: on the roles of sea level rise, storm surges, hurricanes, and the Gulf Stream. Mar. Technol. Soc. J. 52 (2).

Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., 2020. Array programming with NumPy. Nature 585 (7825), 357–362.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., 2020. The ERA5 global reanalysis. Q. J. R. Meteorol. Soc. 146 (730), 1999–2049.

Huang, W., Ye, F., Zhang, Y.J., Park, K., Du, J., Moghimi, S., Myers, E., Pe'eri, S., Calzada, J.R., Yu, H.C., Nunez, K., 2021. Compounding factors for extreme flooding around Galveston Bay during hurricane harvey. Ocean Model. 158, 101735.

Jang, J.H., Chang, T.H., 2022. Flood risk estimation under the compound influence of rainfall and tide. J. Hydrol. 606, 127446.

Legrand, S., Deleersnijder, E., Delhez, E., Legat, V., 2007. Unstructured, anisotropic mesh generation for the Northwestern European continental shelf, the continental slope and the neighbouring ocean. Continent. Shelf Res. 27 (9), 1344–1356.

Lyard, F.H., Allain, D.J., Cancet, M., Carrère, L., Picot, N., 2021. FES2014 global ocean tide atlas: design and performance. Ocean Sci. 17 (3), 615–649.

Mani, S., Moghimi, S., Cui, L., Wang, Z., Zhang, J.Y., Lopez, J., Myers, E., Cockerill, T., Pe'eri, S., 2022. On-demand Automated Storm Surge Modeling Including Inland Hydrology Effects, vol. 52. NOAA technical memorandum NOS CS, Series.

Moftakhari, H.R., Salvadori, G., AghaKouchak, A., Sanders, B.F., Matthew, R.A., 2017. Compounding effects of sea level rise and fluvial flooding. Proc. Natl. Acad. Sci. USA 114 (37), 9785–9790.

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.

Oliveira, A., Fortunato, A.B., Rogeiro, J., Teixeira, J., Azevedo, A., Lavaud, L., Bertin, X., Gomes, J., David, M., Pina, J., Rodrigues, M., 2020. OPENCoastS: an Open-Access Service for the Automatic Generation of Coastal Forecast Systems, vol. 124. Environmental Modelling & Software, 104585.

Oliveira, A., Fortunato, A.B., Rodrigues, M., Azevedo, A., Rogeiro, J., Bernardo, S., Lavaud, L., Bertin, X., Nahon, A., de Jesus, G., Rocha, M., 2021. Forecasting Contrasting Coastal and Estuarine Hydrodynamics with OPENCoastS, vol. 143. Environmental Modelling & Software, 105132.

Qiang, Y., He, J., Xiao, T., Lu, W., Li, J., Zhang, L., 2021. Coastal town flooding upon compound rainfall-wave overtopping-storm surge during extreme tropical cyclones in Hong Kong. J. Hydrol.: Reg. Stud. 37, 100890.

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. Geosci. Model Dev. (GMD) 12 (5), 1847–1868.

Santiago-Collazo, F.L., Bilskie, M.V., Hagen, S.C., 2019. A Comprehensive Review of Compound Inundation Models in Low-Gradient Coastal Watersheds, vol. 119. Environmental Modelling & Software, pp. 166–181.

Stephens, T.A., Savant, G., Sanborn, S.C., Wallen, C.M., Roy, S., 2022. Monolithic multiphysics simulation of compound flooding. J. Hydraul. Eng. 148 (9), 05022003.

Thompson, J.F., Soni, B.K., Weatherill, N.P. (Eds.), 1998. Handbook of Grid Generation. CRC press.

Umlauf, L., Burchard, H., 2003. A generic length-scale equation for geophysical turbulence models. J. Mar. Res. 61 (2), 235–265.

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. Nat. Clim. Change 5 (12), 1093–1097.

Willis, T., Wright, N., Sleigh, A., 2019. Systematic Analysis of Uncertainty in 2D Flood Inundation Models, vol. 122. Environmental Modelling & Software, 104520.

Xu, K., Wang, C., Bin, L., 2022. Compound flood models in coastal areas: a review of methods and uncertainty analysis. Nat. Hazards. 116, 469–496.

Ye, F., Zhang, Y.J., Wang, H.V., Friedrichs, M.A., 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 Model. 127, 16–39.

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 Model. 143, 101466.

Ye, F., Zhang, Y.J., Yu, H., Sun, W., Moghimi, S., Myers, E., Nunez, K., Zhang, R., Wang, H.V., Roland, A., Martins, K., 2020. Simulating storm surge and compound flooding events with a creek-to-ocean model: importance of baroclinic effects. Ocean Model. 145, 101526.

Ye, F., Huang, W., Zhang, Y.J., Moghimi, S., Myers, E., Pe'eri, S., Yu, H.C., 2021. A cross-scale study for compound flooding processes during Hurricane Florence. Nat. Hazards Earth Syst. Sci. 21 (6), 1703–1719.

Zhang, Y.J., Ye, F., Stanev, E.V., Grashorn, S., 2016. Seamless cross-scale modeling with SCHISM. Ocean Model. 102, 64–81.

Zhang, Y.J., Ye, F., Yu, H., Sun, W., Moghimi, S., Myers, E., Nunez, K., Zhang, R., Wang, H., Roland, A., Du, J., 2020. Simulating compound flooding events in a hurricane. Ocean Dynam. 70 (5), 621–640.

Zhang, Y.J., 2021. Assessment of subgrid method in a finite-volume model. Comput. Math. Appl. 81, 220–236.