3-23-2022

Automation and Coupling of Models for Coastal Flood Forecasting in South Texas

Cesar Davila Hernandez
The University of Texas Rio Grande Valley, cesar.davilahernandez01@utrgv.edu

Sara E. Davila
The University of Texas Rio Grande Valley

Martin Flores
The University of Texas Rio Grande Valley

Jungseok Ho
The University of Texas Rio Grande Valley, jungseok.ho@utrgv.edu

Dong-Chul Kim
The University of Texas Rio Grande Valley, dongchul.kim@utrgv.edu

Recommended Citation

Hernandez, Cesar Davila, et al. "Automation and Coupling of Models for Coastal Flood Forecasting in South Texas." Journal of Extreme Events (2022): 2250001. https://doi.org/10.1142/S2345737622500014

Automation and coupling of models for coastal flood forecasting in South Texas

Cesar Davila Hernandez
Department of Civil Engineering
cesar.davilahernandez01@utrgv.edu

Sara Davila
Department of Civil Engineering
sara.davila01@utrgv.edu

Martin Flores
Department of Civil Engineering
martin.flores05@utrgv.edu

Jungseok Ho
Department of Civil Engineering
jungseok.ho@utrgv.edu

Dong-Chul Kim
Department of Computer Science
dongchul.kim@utrgv.edu

The University of Texas Rio Grande Valley,
1201 W. University Dr.
Edinburg, Texas 78539, United States

Forecasting of natural disasters such as inundations can be of great help for emergency bodies and first responders. In coastal communities, this risk is often associated with storm surge. To produce flood forecasts for coastal communities a system must incorporate models capable of simulating such events based on forecasted weather conditions. In this work, a system for forecasting inundations based predominantly on storm surge is explored. An automation and coupling strategy were implemented to produce forecasted flood maps automatically. The system leverages an ocean circulation model and a channel water flow model to estimate flood events in South Texas specially alongside the Lower Laguna Madre. The system around the models is implemented using Python and the meteorological forcing input is obtained from weather forecasting models maintained by the National Oceanic and Atmospheric Administration. The forecasted weather data retrieval, data processing and automation of the models is successful, and the complete stack of software can be deployed locally or in cloud solutions to accelerate computations. The resulting system performs as expected and successfully produces flood maps automatically providing vital information for flood emergency management in coastal communities.

Keywords: Forecasting; coastal flood; automation; storm surge.

1. Introduction

Coastal communities along the Gulf of Mexico are prone to suffering damage from storms and flood events. The Gulf of Mexico has seen hurricanes of varying categories which impact has been detrimental to the communities alongside its shore. The geological features and location of South Texas coastal communities render them especially vulnerable to the harmful effects of tropical cyclone. Since the 1850s nearly 300 hurricanes have struck the United States mainland with almost a 100 of them being of category 3 or greater. Every year, hurricanes and storms are the cause of substantial damage to property and infrastructure in coastal and inland cities, major hurricanes such as Katrina of 2005 or Ike 2008 have caused damages worth $20 billion dollars (Morss & Hayden, 2010). The apparent trend since the 1970's is that tropical cyclones continue to grow in strength and destructive potential (Emanuel, 2005). The increase in ocean temperatures combined with the expected increase in population can be a potentially harmful combination for coastal communities. The 2000s saw a surge in population on coastal areas in the United States. The population growth ever since has driven the demand and construction of infrastructure to sustain the influx, from highways and bridges to new water and electrical power. These new changes have exposed billions of dollars worth of infrastructure to potential damages due to hurricane events. The potential for widespread long-term disaster is heightened in the case of coastal areas where chemical complexes exist. In the Houston Ship channel, for example, many petrochemical complexes are located in potentially vulnerable areas that could be easily damaged during a storm surge event. These locations heighten the destructive power of storm surge and provide a good incentive for creating contingency plans and prevention systems (Burleson et al., 2015; Christian et al., 2015; Cutter et al., 2007). In recent years, South Texas communities have experienced the negative impact that floods can have on their economic activities and daily life. Floods are especially frequent in the area because of its geological features that allow water to accumulate easily even in the presence of moderate precipitation and storm surge events. Storm surge refers to the abnormal rise of the ocean above predicted astronomical tides and is one of the main dangers that hurricanes pose to coastal communities(Zachry et al., 2015). Storm surges are a common contributor to floods in South Texas coastal communities and while it is not common for hurricane storms to hit land in South Texas the remnant winds and precipitation of passing by storms can still increase storm surge and precipitation in the area, causing floods. South Texas coastal communities are aware of the potential havoc that hurricanes can bring to property and infrastructure in the region as well as the impact it can have on private individuals, further, the latent worry of an environmental disaster is heightened when considering the importance of the Lower Laguna Madre, a hypersaline lagoon, that sustains the tourism, fishing and shrimping industries of the region (Bartlett, 2002).

The goal of this study is to develop a flood forecasting system using a couple numerical model to predict flood events in coastal areas due to contributions of coastal storm surge and precipitation surface runoff. A system will be developed to couple the models together and execute their computations automatically to estimate floods based on precipitation and storm surge contributions. The automation and coupling work will allow the system to operate unsupervised and reliably. The flood forecasts will then be distributed through a publicly accessible delivery system. In extreme emergency management, providing the right set of tools could be the difference that prevents the deployment of sub-optimal responses to disasters. As explored by many studies such as (Gunes & Kovel, 2000; Nóbrega et al., 2008) the use of interactive systems that provide a better picture of what a potential disaster can look like is vital. The system can also deliver time-series maps of flood coverage to visualize the evolution of the disaster event. This provides emergency bodies with the capacity to see how the inundation will spread over time into the affected area and prioritize their efforts to areas that will be immediately affected. The granularity and the detail that can be extracted from the provided maps can be of great help for emergency response and help focus the resources available more efficiently. The preliminary concepts of the flood forecasting system, modeling progress, and test-run results were presented in this paper.

1.1 Forecasting System Development

This coastal flood forecasting system is composed of three major phases to maximize practical benefits of the flood forecasting: external data retrieval pre-processing; hydrodynamic computational model; and forecasting results post-processing as depicted in Figure 1. This diagram shows the succession of events and the communication steps that the system takes to produce and publish the final forecasted flood maps. In the pipeline of the computational events, this pre-processing section is concerned with the acquisition of such data automatically. The forecasting system requires externally forecasted data to initialize the models and provide the data as input for their computations.

The hydrodynamic computation model is a main compartment of the forecasting system and is responsible for the computation of the flood forecast. This phase is composed of two numerical models of the ocean flow circulation model and the watershed flood model to compute the coastal storm surge and its impact on precipitation surface runoff. ADvanced CIRCulation (ADCIRC) program was adopted for the ocean flow circulation computation (Rick A. Luettich & Joannes J Westerink, 2018). This model produces an estimation of water surface elevation due to forecasted atmospheric conditions retrieved during the data retrieval and processing step. In this project, the vertically-integrated modality or 2D ADCIRC is leveraged (Luettich & Westerink, 2004). The ADCIRC model utilized was conceived through an improvement and calibration process detailed in (Davila et al., 2020). The main improvements to the model lie on the usage of specific tidal constituents and spatial nodal attributes derived from TxBLEND references (Matsumoto, 1993; Mayo et al., 2014; Schoenbaechler et al., 2011; Szpilka et al., 2016). The estimation from the ocean model will be forwarded to Hydrologic Engineering Center River Analysis System (HEC-RAS) a computer program, developed by the U.S. Army Corps of Engineers, used in this system to predict coastal watershed floods events. This watershed flood model also produces final watershed flooding maps. To ensure an efficiency of the comprehensive forecasting of the coastal flood event due to the inland surface runoff and coastal storm surge, the model runs of the system were automized. The watershed flood model runs under a Windows operating system, while the ocean circulation model uses a Linux operating system as shown in Figure 1.

In this study, a communication framework that handle data transfer between the two different operation systems and the models. The framework is responsible of coupling the models needs to create a pathway of communication between the operating systems and a way to signal the succession of events to both models. This flow of events starts from the acquisition of data and can be followed all the way to the distribution of the map in a delivery system. The post-processing section was implemented for the computation output data processing and distributions as a final phase of the forecasting system. A linkage between the hydrodynamic models and a web application running GIS tools was also developed to complete the model automations. Python was adopted for the creation of the forecasting flood system automations due to its flexibility and applicability of modules and libraries such as NumPy and SciPy (Oliphant, 2007). The final step in the flood forecasting system is the implementation of a delivery system. This delivery system is aimed at distributing the results produced by the numerical models. The flood forecasts produced will be posted on an interactive web map which will display the locations where floods will occur in South Texas coastal areas. The web map is expected to work in low end devices and some optimizations were made to the flood maps in order to reduce the strain of loading and running the website on low-speed connections and devices.

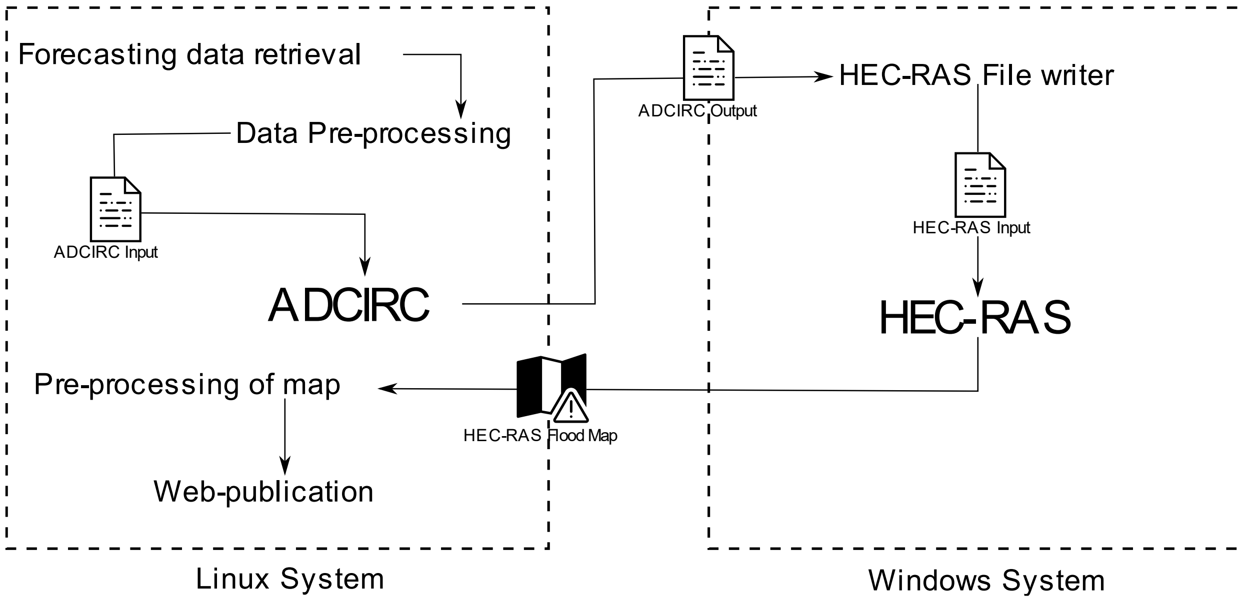

Figure 1. High-level representation of the proposed forecasting system. Succession of events begins at the top-left corner.

2. Data Retrieval and Pre-processing

The models involved in the system require forecasted data to be initialized and fed, such forecasted data needs to provide a specific set of atmospheric variables. The retrieved forecasting data contains meteorological forcing variables such as winds and pressure in a gridded format. This information serves as numerical boundary conditions for the coupled models. The National Oceanic and Atmospheric Administration (NOAA) provides various types of climate forecasting data, such as precipitation, wind direction and speed as well as atmospheric pressure which are required by the models. Surface level and time series data were retrieved and then were spanned to the modeled area.

The North American Mesoscale Forecasting System (NAM) (National Oceanic Atmospheric Administration, 2006) was adoped for the forecasting data in this study. This model is one of the major climate models distributed and executed by the National Centers for Environmental Prediction (NCEP). This model provided an 84-hour forecasting range meteorological information, which will be adopted as input data for the ADCIRC in a gridded form and more importantly it does so while covering the entirety of the ADCIRC model domain used in this forecasting system.

The data retrieval and processing step consists of mainly four major stages to produce the final file that will serve as input for the models. The delineation of these stages is defined by the responsibilities of subroutines existing in the forecasting system code that makes the data retrieval step possible. A Python script was developed to perform each aforementioned step following the logic in the subroutine that can be seen in Figure 2 below.

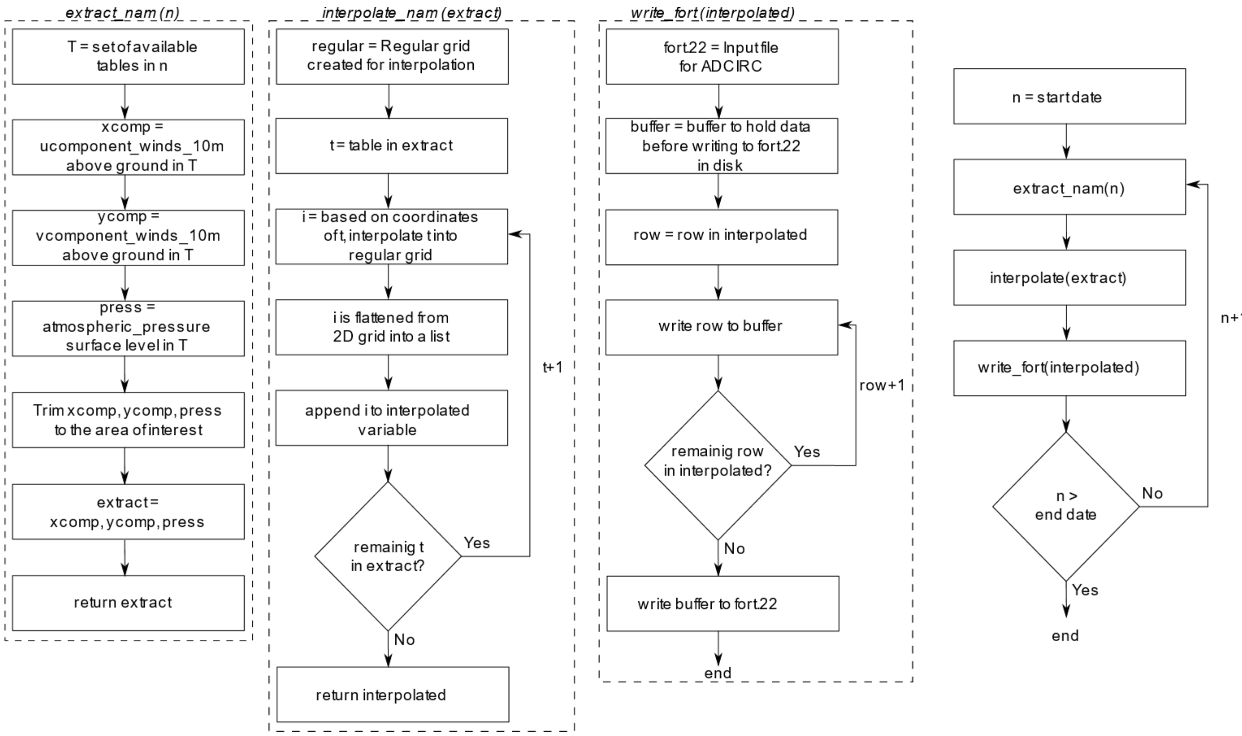

Figure 2. Flowchart representing the Python script developed for the retrieval of NAM datasets.

2.1 Model Input Parameters

The ADCIRC model provides the Model Parameter and Periodic Boundary Condition File (fort.15) required in every model, which enable to change the model inputs according to a single variable. The flexibility that this model provides for its input allows a certain freedom to investigate and compare all the options for providing data to the ocean circulation model and discern the most optimal method for the needs of the forecasting system in this study. Among the several input options that ADCIRC offers two of them were strongly considered. These options are defined by a parameter called Node Wind Stress (NWS) that must be specified in the fort.15 file. Depending on the value given to the NWS parameter ADCIRC expects a different type of input file. Only two input options were strongly considered for adoption as the preferred input format for the forecasting system developed in this study. The NWS parameter setting is specified by an integer and in this case, it can be set to two chosen data formats which are: Wind velocity and atmospheric pressure defined at every node and every time interval (NWS = 5), and Wind velocity and atmospheric pressure defined as a grid at every time interval (NWS = 6). Regardless of which NWS parameter is chosen ADCIRC expects the Meteorological Forcing Input File (fort.22). The fort.22 file contents change according to which NWS setting was written into the fort.15 input file. The ADCIRC program expects different fort.22 contents for different NWS parameters and will not run a simulation unless the fort.22 file contents match what is expected. A script was developed to automate the creation of a fort.22 file for NWS six parameter.

2.2 Data Retrieval and Extraction

NAM data is distributed by the National Centers for Environmental Prediction (NCEP) FTP server in a data format called GRIB2. This format is a binary file containing grids for different atmospheric levels over a specific domain space. Location information is also embedded on the file. GRIB and GRIB2 are data formats standardized by the World Meteorological Organization and used in numerical weather prediction models such as NAM. To proceed with the next phases of data processing it is necessary to interact with the GRIB files retrieved. The Pygrib library (Jeffrey Whitaker, 2010) was adopted to serve

es as an interface with the ECWMF GRIB_API C library and the NCEP grib2 C library. These are the most popular libraries for interacting with GRIB files and with Pygrib easily available.

For the data extraction process, a script was developed to extract the correct data that ADCIRC needs for its input fort.22 file of the two meteorological input variables: components of winds and atmospheric pressure at ten meter above ground and surface level respectively. The Pygrib was used to select the variables that we need and extract them from the records retrieved during data retrieval as shown in the extract_nam(n) method in Figure 2. Once the data retrieval step is completed the subsequent routines will be performed for each time step in the forecasting dataset retrieved. A time step of the NAM dataset represents a single forecasted hour, and it contains all the meteorological variables forecasted for that time step. Once a time step is handed over to the subroutines the first step of data extraction will happen. The script will select the tables for Wind Components at 10 meters above surface and Atmospheric Pressure at the surface level. These tables selected will be stored in memory for later processing. The tables selected in the data extraction step contain gridded information spanning the complete domain of the NAM model.

2.3 Data Clean up and Trimming

A script was created to optimize the computing disk space and to reduce the complexity of the data used to create the input fort.22 file. The script leverages the same libraries used in the data extraction step and is also included in the method extract_nam(n) in Figure 2. These libraries offer a couple of data trimming methods for limiting the size of the grid to a quadrangular shape. These methods are used on the data loaded in memory to reduce the size of it into a rectangle containing information only pertaining to the Gulf of Mexico. The data that is outside the boundaries specified in the method is discarded. The resulting dataset is considerably smaller and flexible to process and handle. This table replaces the original table created during data extraction and will be handed over to the next subroutine for further processing.

2.4 Data Interpolation

The NWS six parameter data should be assigned to ADCIRC model as a regular grid spanning the complete modeled area as well as provide information for each computational time step from the beginning to the end of the simulation. The requirement that this step focuses on is the fulfillment of the “regular grid” requisite. The regular grid requirement by NWS six specifies that the gridded information that was provided by the fort.22 file should be regularly spaced, that is, the size of the grid cells in the latitude and longitude coordinates should be constant in the complete grid. The NAM forecasting data grid is not regularly spaced and tilts in some places as it can be seen on Figure 3. Data interpolation process was needed to adjust the irregular grid.

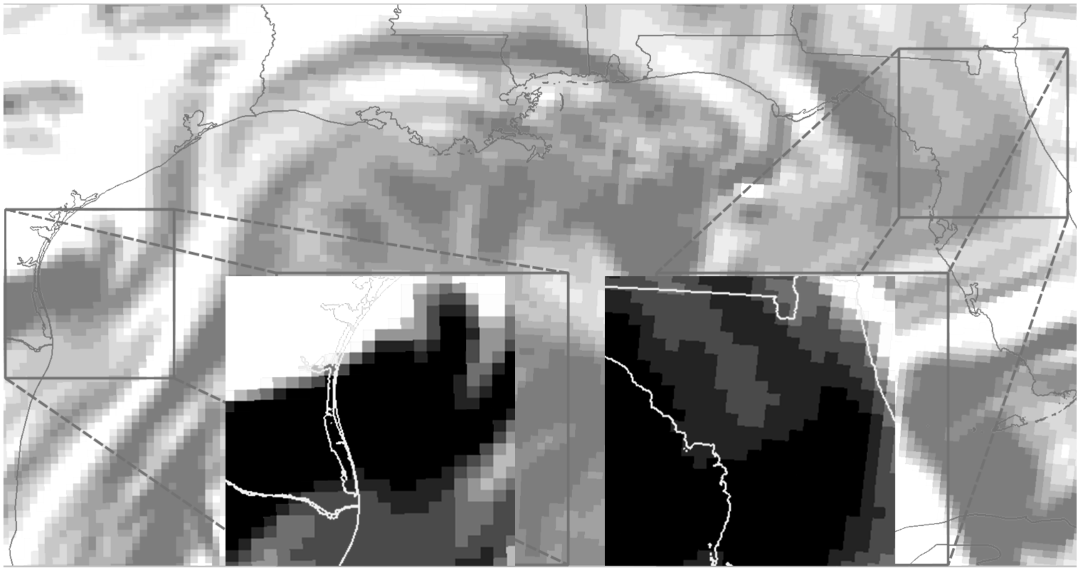

Figure 3. Snapshot of raw NAM data over the Gulf of Mexico divided in two sections: Texas Lower Laguna Madre coast and Florida. The irregularity (incline) of the squares in the grid can be seen better on the zoomed in square over Florida.

When selecting the NWS six parameter for simulation, ADCIRC expects the user to specify target information of the shape of the grid provided alongside the fort.22 file. The fort.22 file doesn't contain information on the location of each cell, while it only contains the meteorological variables for each cell in the grid. Since ADCIRC expects information on the shape of the grid provided, the fort.22 file is read by using those parameters that explain the shape of the grid and the number of cells it contains. When utilizing NWS six parameter the program expects the user to provide information on the size, shape, location and number of grid cells that the gridded fort.22 dataset will map into. This information was specified beforehand into the fort.15 input file for ADCIRC. The script developed to handle the data interpolation generates these numbers that define the regular grid.

For data interpolation, NumPy (Harris et al., 2020) subroutine was added to support for multidimensional arrays and matrices to the system. This library was used by the routine interpolate_nam(extract) seen on Figure 2 to create a regular empty grid with the parameters that a user specifies. The empty regular grid that was created with the help of NumPy was used for interpolating the NAM irregular gridded data. The fabricated grid contains location information that was used to interpolate data from the NAM forecasting irregular grid. The size of the grid, location, and the size of each grid cell were also defined using the library. A regular grid with no wind and pressure information was also created to be used for the data interpolation.

The SciPy (Virtanen et al., 2020) library was adopted for the data interpolation due to its flexibility on the interpolation of data into a grid. The grid fabricated using NumPy served as input for the SciPy functions which allows for the interpolation to be accomplished. The functions provided by SciPy require an empty grid and a grid containing the information from which the interpolation will take place. The SciPy routine tessellates the domain into N-dimensional simplices. Linear interpolation is applied to each simplex with SciPy's LinearNDInterpolator function which triangulates the input data and performs linear barycentric interpolation. The interpolation is performed using the three delimiting points of each triangle which are weighted based on their barycentric coordinates. The routine handles the interpolation of each parameter separately, meaning, the interpolation of the x-component of winds is performed first followed by the y-component and then the atmospheric pressure grid. Once the interpolation step is accomplished the previous NAM data are discarded, only data that pertain to the regular interpolated grid for all the meteorological variables required by ADCIRC are kept.

After the interpolation step is completed for one forecasted time step the data in memory will be handed to the fort.22 writing routine explained at the beginning of the section which its logic can be seen on the method write_fort(interpolated) on Figure 2 . This method dumps the tables into a buffer while following the ADCIRC's documentation format for fort.22 files. The buffer is then written to disk as a fort.22 file. The consequent steps simply append to this file in the same manner until every time step in the NAM data is processed. An example of the resulting fort.22 file output can be seen on Figure 4. The arrows and circle indicate two-dimensional wind velocity vectors and the eye of the hurricane, respectively. The wind vectors clearly define the symmetrical flow field of the hurricane, which tends to push water in the circular motion of the symmetrical cyclone. The wind stress is a significant contributor to storm surge propagation.

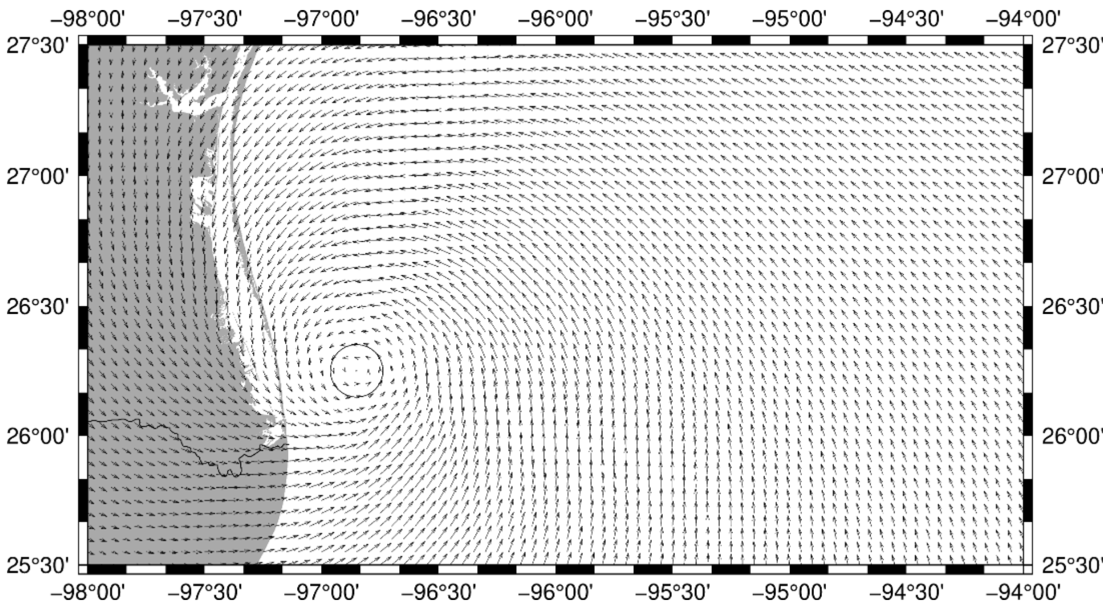

Figure 3. Result of processing and interpolation of NAM data for ADCIRC input creation. This specific time step shows Hurricane Dolly (2008) approaching the Lower Laguna Madre.

3. Automation and Model Coupling

The automation of this process is aimed at increasing efficiency and allow for the scheduling of tasks around forecasting data availability. The creation of a set of scripts that control and encompass three main processes for each model was focused on the automation process: Model Input, Model Execution and Model Output Handling. For the two models each of these processes are similar in function but different in implementation and must be addressed separately. The Model Input and Model Output Handling processes are tightly related with previous and following steps and therefore require special attention. This system with the capability of forecasting flood maps in a range of up to 3.5 days should increase the planning capacity of emergency management personnel in the area as well as private individuals. The forecast can be updated at least every 6 hours once the most updated forecasting weather dataset is made available.

3.1 Ocean Circulation Model

The estimation of storm surge through the use of ADCIRC is the first model simulation to be executed in the forecasting system pipeline of events as shown in Figure 1. The preliminary considerations for creating an automation script concerns the platform that the script will interact with. In order to leverage the computing resources available at The University of Texas Rio Grande Valley (UTRGV) campus the model was installed in a Linux environment during development and deployment. The development of

ADCIRC automation scripts in a Linux platform ensures compatibility with most High-Performance Computing solutions and hosting services. Figure 5 provides the logic of the routines implemented for the automation of ADCIRC model.

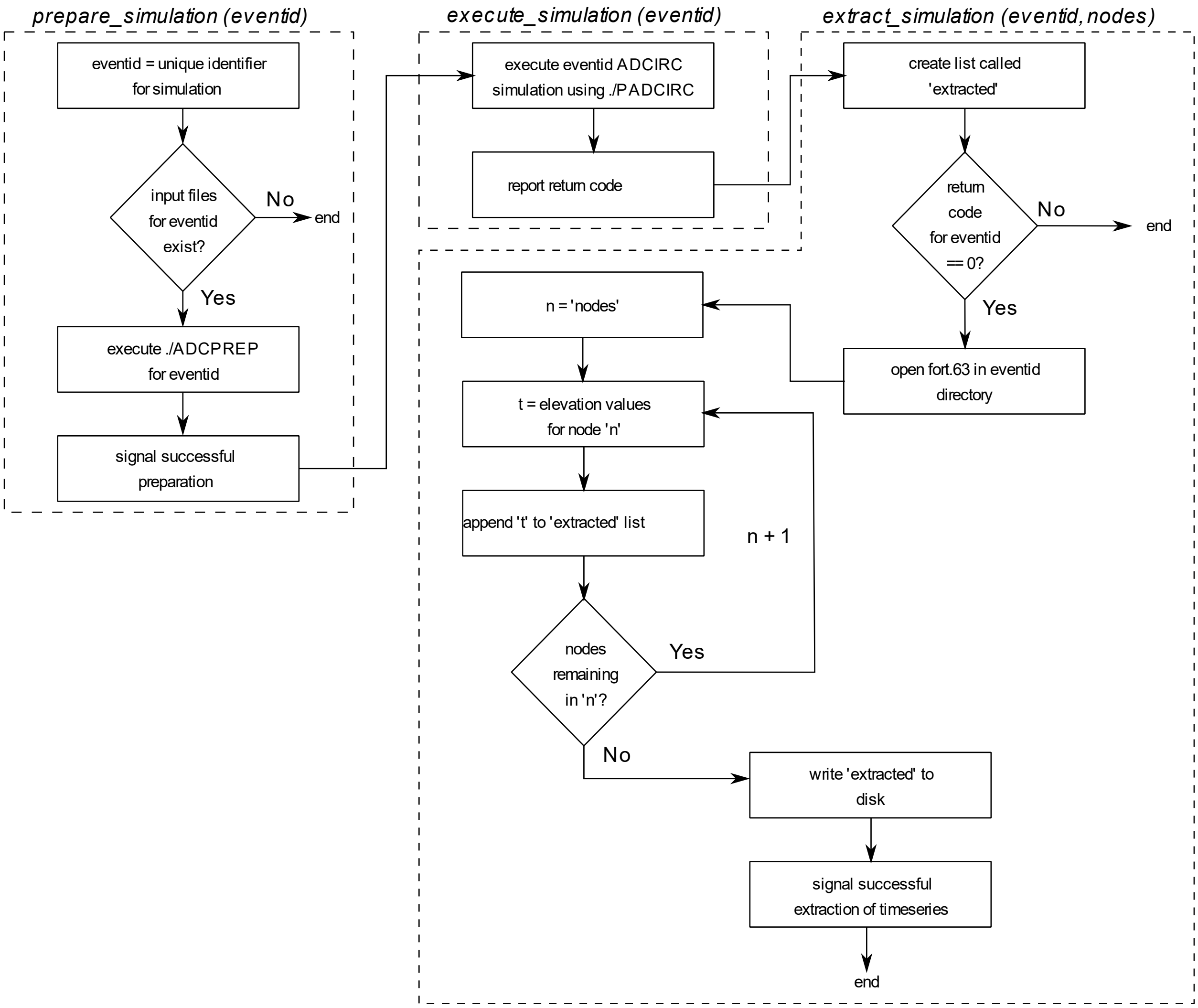

Figure 5. Flowchart representing the Python script developed for the automation of ADCIRC.

3.1.1 Ocean Circulation Model Input

The input for ADCIRC is considered as the set of files given to the program that includes the meteorological forcing data in the form of a fort.22 file and the model parameters and tidal forcing information contained in the fort.15 file. Other files containing information, e.g., grid data and nodal attributes were not included of the Model Input automation process for the model, because that information remain constant for all forecasting runs and no automation strategy is required.

A script was created to automate the creation of a fort.15 file, since the tidal constituent boundary conditions change in relationship with time. The script controlling Model Input for ADCIRC considers the time the forecast will be made in order to create an appropriate fort.15 file containing synchronized tidal constituents for the initialization of the model. The fort.22 file was provided as well as the fort.15

file and they were merged into a directory with the rest of the files necessary to run the model. Through the automation, the fort.22 file was created following the routines explained in the data retrieval and processing section.

3.1.2 Ocean Circulation Model Execution

ADCIRC behaves as a Command Line Application (CLA) in Linux operation system and provides the user with a set of commands to interact with its executables. A script was developed to control both aspects of the Model Execution. Pexpect (Noah Spurrier, 2003) library offers a set of tools to interact with CLA's programmatically and was used for the automation of CLA. The preparation of the model requires the presence of the input in a directory that is predefined and known to the script that will prepare the simulation. The script then, uses Pexpect to spawn the executable responsible for prepping a simulation called "adcprep", which is shown in Figure 5 as the method prepare_simulation(eventid). The script issues commands to this executable based on a predefined set of steps programmed, and also uses Pexpect to read the prompts given by the CLA after every step executed. This is done to find errors and, if needed, restart the prepping process.

After this script has successfully prepped the simulation for execution, the second routine called execute_simulation(eventid) in Figure 5 starts the process for executing ADCIRC. The routine uses Pexpect as well to spawn an instance of "padcric", the executable for parallel simulations. This second script works as a trigger when the prepping step is completed and utilizes the MPI library, the "mpiexec" command to start an instance of "padcirc". The routine responsible for running the simulation is aware of what output files are to be expected. This is useful because when the MPI call returns 0 the script checks for the presence of the output files in order to confirm the size of the data written to them is the expected. This is a naive check that lets the routine know the simulation has written output files until completion. Although this doesn't ensure that the simulation results are correct but is a preliminary check that will catch some errors before they are spread down the line.

3.1.3 Ocean Circulation Model Output Handling

The Automation routines responsible of extracting information out of the output files generated by a successful run of an ADCIRC simulation are concerned with mainly location and type of data extracted and storage in disk of the data extracted. ADCIRC is the first model to be executed in the forecasting system, its output data will be used as input for the second model. The generated output files were dictated by the configuration file fort.15, which is composed of the settings of the output that ADCIRC writes to disk in each simulation. In this study, the Elevation Time Series at All Nodes in the Model Grid (fort.63), which contains information of water surface elevation for every node in the grid, was used to account for the contribution of storm surge to floods in coastal areas.

A script was developed to, based on the contents and structure information, parse the fort.63 file and extract the output of specific nodes that are of interest to the forecasting system, and which coincide with locations already set up in the HEC-RAS coastal flood model. The script also extracts the information and construct a time series table for the specified nodes. In addition, the script extracted location information of the nodes from the Grid and Boundary Information File (fort.14), which is not included in the fort.63 file. The nearest nodes to the specified locations were selected. Once this information is attained the script should be able to simply extract all the water surface elevation time series information for a specific set of nodes for the length specified within the simulation time period. When the extracted information is on memory it should be formatted and written into a file for later use by the coastal flood model.

3.2 Coastal Flood Model

HEC-RAS coastal flood model will execute automatically after a successful run of ADCIRC as shown in Figure 1. The model utilizes the water surface elevation output from ADCIRC as a stage hydrograph to initialize the model and to read input at every time step. This forecasting system initially designed to

adopt a Windows operating system HEC-RAS 5.0.4. This added a layer of complexity to the communication between the models. Having two different operating systems implies the creation of a communication framework for the computers and models. Such a framework should supply a pathway for sharing information between operating systems, and effectively, between models. Figure 6 illustrates the logic behind the Python scripts developed to control HEC-RAS used in the forecasting system.

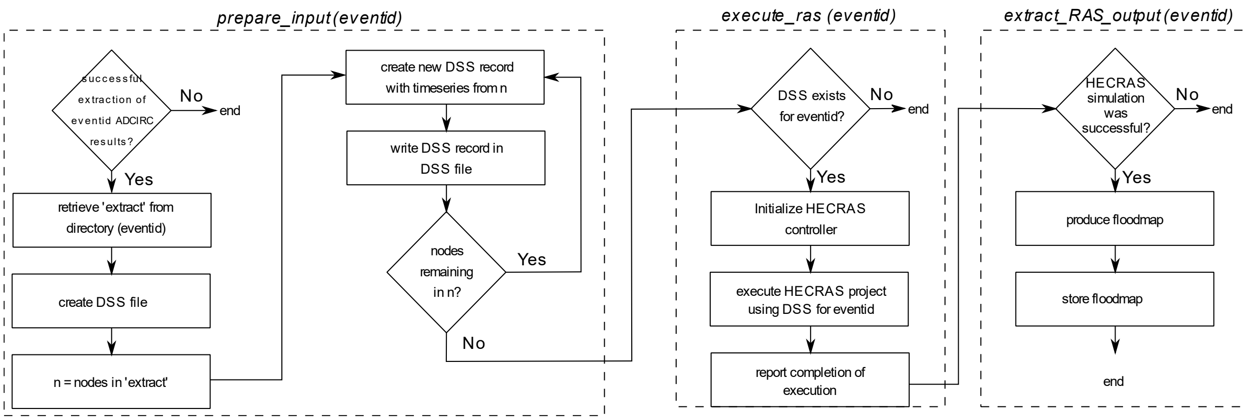

Figure 6. Flowchart representing the Python script developed for the automation of HEC-RAS.

3.2.1 Coastal Flood Model Input

The extract_simulation(eventid) in Figure 5 stores relevant information from specified nodes from the ADCIRC model output in a file. The file was shared with HEC-RAS to start the Model Input process. A stage hydrograph, hourly water surface elevations in a unit of feet were recorded for every different location. Since the file created by the Model Output Handling process of ADCIRC cannot be directly used by HEC-RAS, the file was sent to the machine running HEC-RAS and a data conversion process began. The logic behind this process is described in the method called prepare_input(eventid) in Figure 6.

Database Storage System (DSS) developed by U.S. Army Corps of Engineers (Hydrologic Engineering Center, 2005) was adopted as the input data assigning. DSS files are a type of database for data that is mostly sequential in nature. A DSS file contains records that can be read and written by HEC-RAS and other HEC applications. DSS files can be constructed and modified by Python with the help of a library called Pydsstools (Gyan Basyal, 2018). Utilizing DSS files and the Pydsstools library helped the readability of HEC-RAS input and reduce bugs and problems related with other input alternatives for HEC-RAS. A script was developed to handle the automatic creation of a DSS file record to serve as input for HEC-RAS. The script is aimed at creating a set of DSS records that come from ADCIRC's output. The DSS file will be shared with HEC-RAS to initialize and perform its simulation. The library Pydsstools allowed to perform CRUD (Create, Read, Update and Delete) operations in a DSS database file. The script responsibility was to read the output produced by ADCIRC for water surface elevations and stored the information into various records in a DSS file.

3.2.2 Coastal Flood Model Execution

In this study, HECRASController, a Component Object Model (COM) was used to control HEC-RAS. This COM interface provides a set of predefined methods to control certain characteristics of the model and was accessed through Visual Basic for Applications following a suggestion from (Goodell 2014). A Python script was developed to access HECRASController by using Pywin32 library (Hammond & Robinson, 2000), which enable to approach to the WIN32 Application Programming Interface (API), which is a Microsoft Windows API that allows for the use of COM objects. The script loaded a HEC-

RAS project and perform computations, the method execute_RAS(eventid) in Figure 6 showcases the Python routine for executing the model.

3.2.3 Coastal Flood Model Output Handling

HEC-RAS provides computation output as the DSS formation files, and RASMapper function visualizes the combination of geometric data and computation results, which fit for one of objectives of the forecasting system. Accessing RASMapper was not necessary when performing a forecasting run. RASMapper function was set with HEC-RAS to produce flooding maps before running a simulation. The model that is already set up could be reused with new input data to perform another simulation and get another set of flood map. This output handling step is responsible of saving the output flood map into a labeled location to be later pre-processed and uploaded into the delivery website for the public.

3.3 Output Delivery System

An online delivery system was created to distribute the maps of coastal flood forecast generated by the system to the end users. The characteristics of the delivery system are intended to widen the accessibility of any prospective user trying to reference the flood forecasts. The forecasted flooding maps produced need to be handled and processed before being posted in the delivery system. This pre-processing step is aimed at extracting shape information out of the raster flooding maps produced by the coastal flood model. The resulting shape information needs to be cleaned and simplified to reduce its size, thereby reducing loading times on slow connections, which supports the objective of emergency decision support system.

In this study, two libraries were adopted for the pre-processing task of manipulating geospatial data. The Geospatial Data Abstraction Library (GDAL) (Warmerdam, 2008) is used in conjunction with the JavaScript library called Mapshaper (Harrower & Bloch, 2006). These libraries allow for the conversion of the original raster into a different geospatial file format. The libraries are also capable of reducing the size of the shape file by eliminating redundant shape information. A script was created to use both libraries and simplify all flooding maps before being posted in the delivery system. The raster files are converted to a GeoJSON format for its display in the interactive web map. GeoJSON is a data format used to represent geographic elements that uses a JavaScript Object Notation (JSON) style formatting. It allowed compatibility with future features that require shape information such as the implementation of a routing engine to help users navigate safely around the flood zone.

4. Results

A complete run of the forecasting system illustrated in Figure 1 was performed to test the automation/coupling of the models. The scope of this test-run was to confirm the functionality of the system only rather than assessing the accuracy of the results. The coastal area to be simulated in this test-run belongs to the Willacy County watershed in south Texas which was modeled in the HEC-RAS coastal flood model. The Gulf of Mexico with the Lower Laguna Madre area was selected for the ADCIRC storm surge model. The system was set to run two experiments. The first run was set to be a decoupled run where only the Willacy County flood model would be executed. The second run included the Lower Laguna Madre storm surge model into the Willacy County flood model.

4.1 Test-Run Model Input

Two types of input were used in this test-run model. For the storm surge model, a historical NAM forecast for Hurricane Dolly (2008) was used to highlight differences in floods when compared to the non-coupled run. Hurricane Dolly is classified the category 2 hurricane determined by the Saffir Simpson Scale (National Aeronautics and Space Administration, 2018) had a 95-mph maximum wind speed with 100 miles of radius. The hurricane's landfall makes a direct impact on the Laguna Madre. Figure 3 provides a snapshot of the fort.22 file to be used as input in this example model. The data was retrieved and processed before being fed into the automated forecast system following the process described. The

computed water surface elevation from the ADCIRC will be forwarded in a form of DSS to HEC-RAS to be used as the downstream boundary condition. While the water surface elevation was used as the downstream flow boundary condition, the watershed sub-basin outlet flowrates were used as the upstream and tributary boundary condition. The major drainage canal cross section survey data and Digital Elevation Model (DEM) terrain data covering the modeling watershed were used as the model geometric data. A total of 5.4 days of the Hurricane Dolly event was simulated in the unsteady state condition.

4.2 Test-Run Model Output

The coupled costal storm surge flood model was successfully tested for the Willacy County coastal watershed. The coupled model executes automatically after a successful run of ADCIRC model. HEC-RAS utilizes the water surface elevation output from ADCIRC as a stage hydrograph to initialize the model and to read input at every time step. The Willacy County HEC-RAS model is composed of two major drain channels, which has downstream stations near the Laguna Madre shoreline. Figure 7 depicts the downstream boundary conditions assigned to the HEC-RAS model for the category 2 hurricane. The coupled indicates downstream boundary condition with hurricane storm surge, while the uncoupled shows the normal tidal level variation without consideration of storm surge.

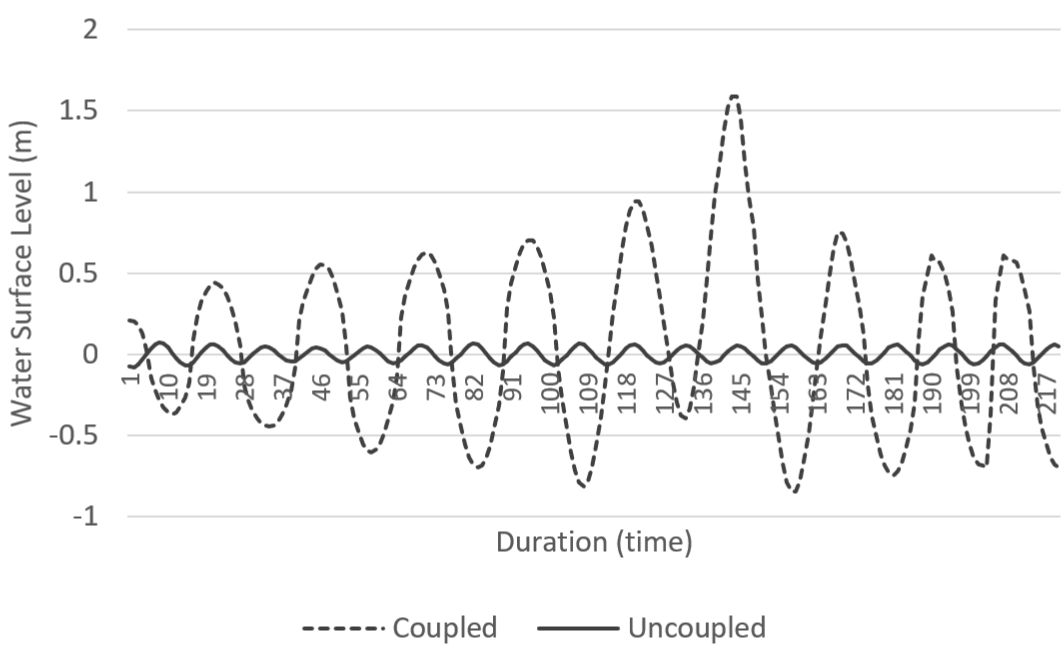

Figure 7. Willacy County coastal flood model downstream boundary conditions for category 2 hurricanes.

The flood map estimation for both runs can be seen on Figure 8. The first automated run only included the execution of the HEC-RAS model. The second run included the coupled ADCIRC and HEC-RAS models. In the output of the coupled system, it is noticeable that the coastal line has advanced significantly when compared with the non-coupled run. This indicates the system is configured correctly since the file exchange and input procedures between ADCIRC and HEC-RAS perform as expected and produces a noticeable change in the output.

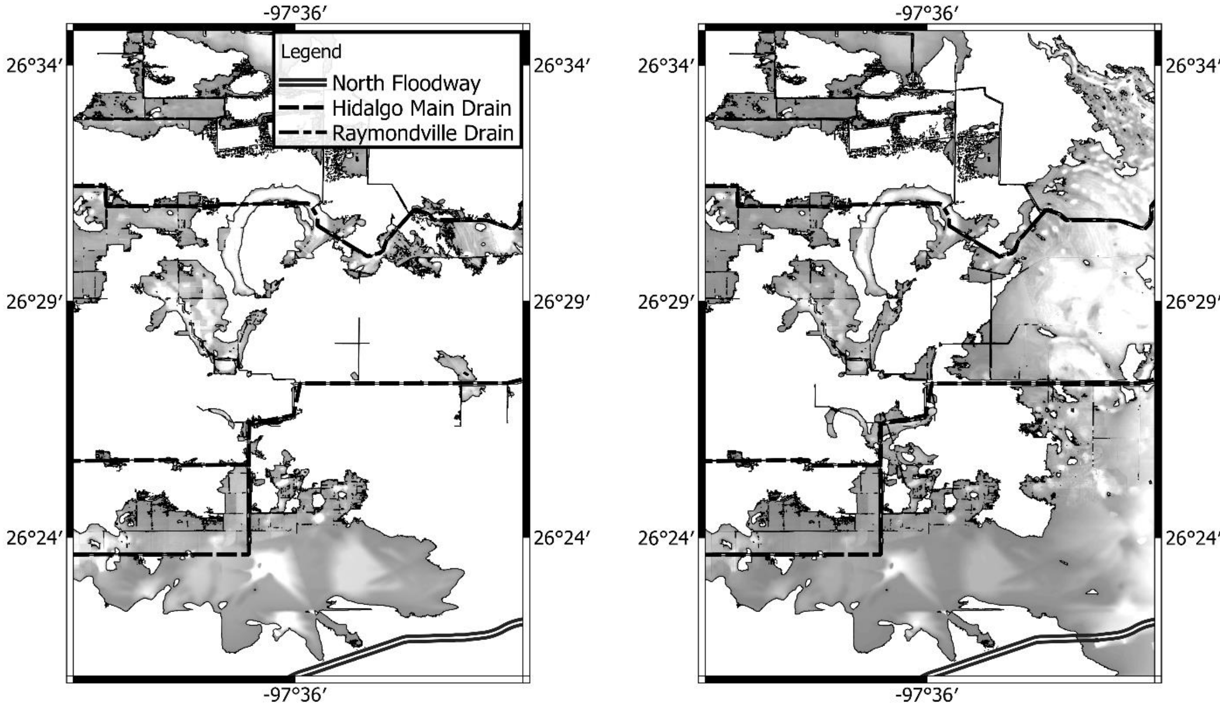

Figure 8. Forecasted flood maps over Willacy County, Texas. Flood is represented in gray. Left map shows floods without contribution from storm surge model (no coupling); right map shows floods with contribution from storm surge model (coupled).

5. Conclusion

The efforts detailed in this study describe the automation of hydrologic models for their integration in a coastal flood forecasting system, which can be useful on emergency planning and disaster management. The system developed takes advantage of two hydrodynamic models for the creation of flood forecasting maps with the help of forecasted climate data. Expanding the functionality of Python with scientific and data processing libraries allowed the development process to focus completely on the automation and coupling strategy and less on the development of tools. In the same manner a set of scripts were developed to handle all the data required, in the form of gridded forecasted weather variables, and the forecasted flood maps produced by the system. The automation strategy and implementation proved successful when conducting testing. The system is capable of sharing data between the models while conducting the necessary format changes. The process can be replicated to provide modelers the capability of automating their workflow when necessary or reproducing the results described. The system described in this work provides a certain amount of flexibility of being deployed on cloud or local systems, leaving such decision to the developers and maintainers. This will be helpful directly to cloud computing solutions are widely available and tend to have reasonable costs. The information provided by this forecasting system can aid in the development of strategies and plans for flood resiliency. Emergency bodies can create plans, establish evacuation routes and focus resources not only in space, but in time. The public can also draw important conclusions from the forecasted flood maps provided. It is imperative to keep improving the model and validate its predictions since currently such aspects have not been considered.

Acknowledgements

The authors acknowledge the support of the Storm Surge Flood Maps Development for the Lower Laguna Madre Coastal Emergency Management grant project sponsored by the Texas General Land Office

Coastal Management Program. The study was also supported in part by the Dwight David Eisenhower Transportation Fellowship program by the Federal Highway Administration.

References

Bartlett, R. C. (2002). *The Laguna Madre of Texas and Tamaulipas*. Texas A&M University Press.

Burleson, D. W., Rifai, H. S., Proft, J. K., Dawson, C. N., & Bedient, P. B. (2015). Vulnerability of an industrial corridor in Texas to storm surge. *Natural Hazards*, *77*(2), 1183–1203.

Christian, J., Fang, Z., Torres, J., Deitz, R., & Bedient, P. (2015). Modeling the Hydraulic Effectiveness of a Proposed Storm Surge Barrier System for the Houston Ship Channel during Hurricane Events. *Natural Hazards Review*, *16*(1), 04014015. https://doi.org/10.1061/(ASCE)NH.1527-6996.0000150

Cutter, S. L., Johnson, L. A., Finch, C., & Berry, M. (2007). The US hurricane coasts: Increasingly vulnerable? *Environment: Science and Policy for Sustainable Development*, *49*(7), 8–21.

Davila, S. E., Davila Hernandez, C., Flores, M., & Ho, J. (2020). South Texas coastal area storm surge model development and improvement. *AIM Geosciences*, *6*(3), 271.

Emanuel, K. (2005). Increasing destructiveness of tropical cyclones over the past 30 years. *Nature*, *436*(7051), 686–688.

Goodell, C. (2014). *Breaking the HEC-RAS Code: A User's Guide to Automating HEC-RAS*. H21s.

Gunes, A. E., & Kovel, J. P. (2000). Using GIS in emergency management operations. *Journal of Urban Planning and Development*, *126*(3), 136–149.

Gyan Basyal. (2018). gyanz/pydsstools: Python library for simple HEC-DSS functions. In *GitHub*. https://github.com/gyanz/pydsstools

Hammond, M., & Robinson, A. (2000). *Python programming on win32: Help for windows programmers*. O'Reilly Media, Inc.

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., & Others. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362.

Harrower, M., & Bloch, M. (2006). MapShaper.org: A map generalization web service. *IEEE Computer Graphics and Applications*, *26*(4), 22–27. https://doi.org/10.1109/MCG.2006.85

Hydrologic Engineering Center. (2005). *HEC-DSS Vue HEC data storage system visual utility engine: User's manual*. US Army Corps of Engineers Davis, Calif.

Jeffrey Whitaker. (2010). jswhit/pygrib: Python interface for reading and writing GRIB data. In *GitHub*. https://github.com/jswhit/pygrib

Luettich, R., & Westerink, J. (2004). *Formulation and Numerical Implementation of the 2D/3D ADCIRC Finite Element Model Version 44.XX*. 74.

Matsumoto, J. (1993). User's manual for the Texas Water Development Board's hydrodynamic and salinity model: TxBLEND. *Texas Water Development Board, Austin, Texas*.

Mayo, T., Butler, T., Dawson, C., & Hoteit, I. (2014). Data assimilation within the Advanced Circulation (ADCIRC) modeling framework for the estimation of Manning's friction coefficient. *Ocean Modelling*, *76*, 43–58.

Morss, R. E., & Hayden, M. H. (2010). Storm surge and "certain death": Interviews with Texas coastal residents following Hurricane Ike. *Weather, Climate, and Society*, *2*(3), 174–189.

National Aeronautics and Space Administration. (2018). *NASA - Hurricane Season 2008: Dolly (Atlantic Ocean)* [Feature]. Brian Dunbar. https://www.nasa.gov/mission_pages/hurricanes/archives/2008/h2008_dolly.html

National Oceanic Atmospheric Administration. (2006). *North American Mesoscale Forecast System (NAM)*. https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/north-american-mesoscale-forecast-system-nam

Noah Spurrier. (2003). pexpect/pexpect: A Python module for controlling interactive programs in a pseudo-terminal. In *GitHub*. https://github.com/pexpect/pexpect

Nóbrega, R., Sabino, A., Rodrigues, A., & Correia, N. (2008). Flood emergency interaction and visualization system. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5188 LNCS*, 68–79. https://doi.org/10.1007/978-3-540-85891-1_9

Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science and Engineering*, *9*(3), 10–20. https://doi.org/10.1109/MCSE.2007.58

Rick A. Luettich & Joannes J Westerink. (2018). *User's Manual—V53—ADCIRC*. https://adcirc.org/home/documentation/users-manual-v53/

Schoenbaechler, C., Guthrie, C. G., Matsumoto, J., & Lu, Q. (2011). TxBlend Model Calibration and Validation for the Laguna Madre Estuary. *Tex Water Dev Board*, *60*.

Szpilka, C., Dresback, K., Kolar, R., Feyen, J., & Wang, J. (2016). Improvements for the western north atlantic, caribbean and gulf of mexico adcirc tidal database (EC2015). *Journal of Marine Science and Engineering*, *4*(4), 72.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., & Others. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272.

Warmerdam, F. (2008). The Geospatial Data Abstraction Library. In *Open Source Approaches in Spatial Data Handling* (pp. 87–104). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74831-1_5

Zachry, B. C., Booth, W. J., Rhome, J. R., & Sharon, T. M. (2015). A National View of Storm Surge Risk and Inundation. *Weather, Climate, and Society*, *7*(2), 109–117. https://doi.org/10.1175/WCAS-D-14-00049.1