Flood and Hurricane Analysis of Hurricane Katrina

GISC 3200K

December 4th, 2020

Authors:

Merritt Gazaway

Prepared For:

Dr. Huidae Cho, Assistant Professor of Geospatial Science and Computing

UNG
UNIVERSITY of NORTH GEORGIA™
LEWIS F. ROGERS INSTITUTE
FOR ENVIRONMENTAL AND
SPATIAL ANALYSIS

Abstract

Although Hurricane Katrina is now 15 years old, it is still interesting to look at a few characteristics that could to be focused on to help create new levee system or processes so catastrophic damage and flooding can be handled and limited if a Hurricane of this power were to occur again. The purpose of this paper was to create a Flood Analysis and Hurricane Analysis by using a ArcGIS Python Tool Box to create two tool scripts to create both analysis's. The Flood Analysis projection was used to find the stream features in certain areas of the state of Louisiana. The Hurricane projection was used to create a model that shows the hotspots and wind speeds of Hurricane Katrina hitting and passing through Louisiana. My ultimate goal was to create a model that gives an indication as to what areas flooded in Louisiana during Hurricane Katrina.

Introduction

Hurricane Katrina will go down as one the most destructive Hurricanes in American History. Billions of Dollars in Damage, an estimated 1800 fatalities, and over 90,000 square miles were affected. Some of the damage caused was due to over 20 levee's that were destroyed by the flood waters destroying the soil that the levees were built on. Most of these areas were washed out due to the types of soil that were in the area. Due to different compaction rates, the water washed away the soil and the levee's allowed the water to spill in. By going back in time, we can look at features such as water bodies that rose during the Hurricane and the Hurricane itself. That way to look at these waterbodies better, creating a flood analysis model that shows the stream features in a certain area allow us to better understand why these waterbodies rose compared to others. It is also necessary to create Hurricane model. For this model, it is best to use a hotspot analysis and a kriging tool to know exactly where the most flooding would be caused based off the hurricane.

Data & Materials

1. HAZUS GPS DATA
2. USGS DEM Louisiana Raster file
3. Python Toolbox with 2 Tools

Study Area

The Primary study area for this project was the state of Louisiana due to the areas of the state below sea level which could have areas of high flood zones.

Methods

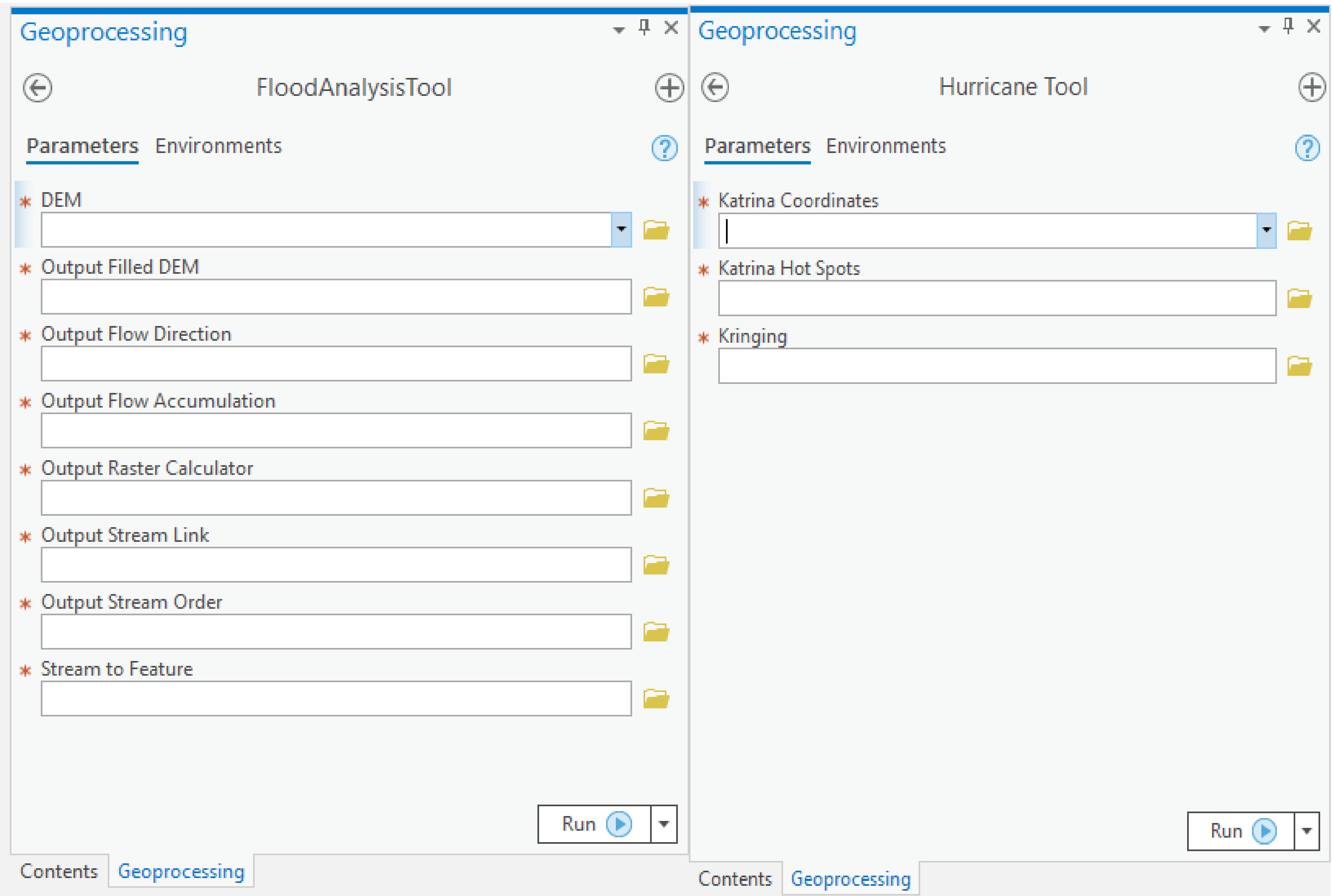

Figure 1. Python Tool Boxes for Hurricane and Flood Analysis

Data for this presentation was downloaded from HAZUS and USGS. The first tool uses the DEM to start the process needed to create this model. From there, the Fill Raster was created using the DEM to create an output raster layer. From there the Flow Direction and Flow Accumulation were created using the same process. From there, the Raster calculator was used. I used the equation >= to create the streams in the raster calc. Using >=50, I was able to create the raster needed to create a layer light enough to see the streams. The next thing needed was the stream link and stream order layer. For both, you need an input flow direction raster. When you create the links, you are connecting the junctions and sections of streams together. Once the stream link and stream order are created you then need to create the stream to feature tool. This will be the first shape file you create in this toolbox and it is also the last output needed for the first tool. After the first tool is created and ran, start the creation of the second tool. This tool is known as the hurricane tool and is created using GPS data. To start you need to add the GPS data into ArcPro. Once you have added it, the next thing you will do is use the Optimize Hot Spot analysis. The last thing needed is this toolbox is the kriging tool. The kriging tool will allow you to see the Input data you are using to a greater margin.

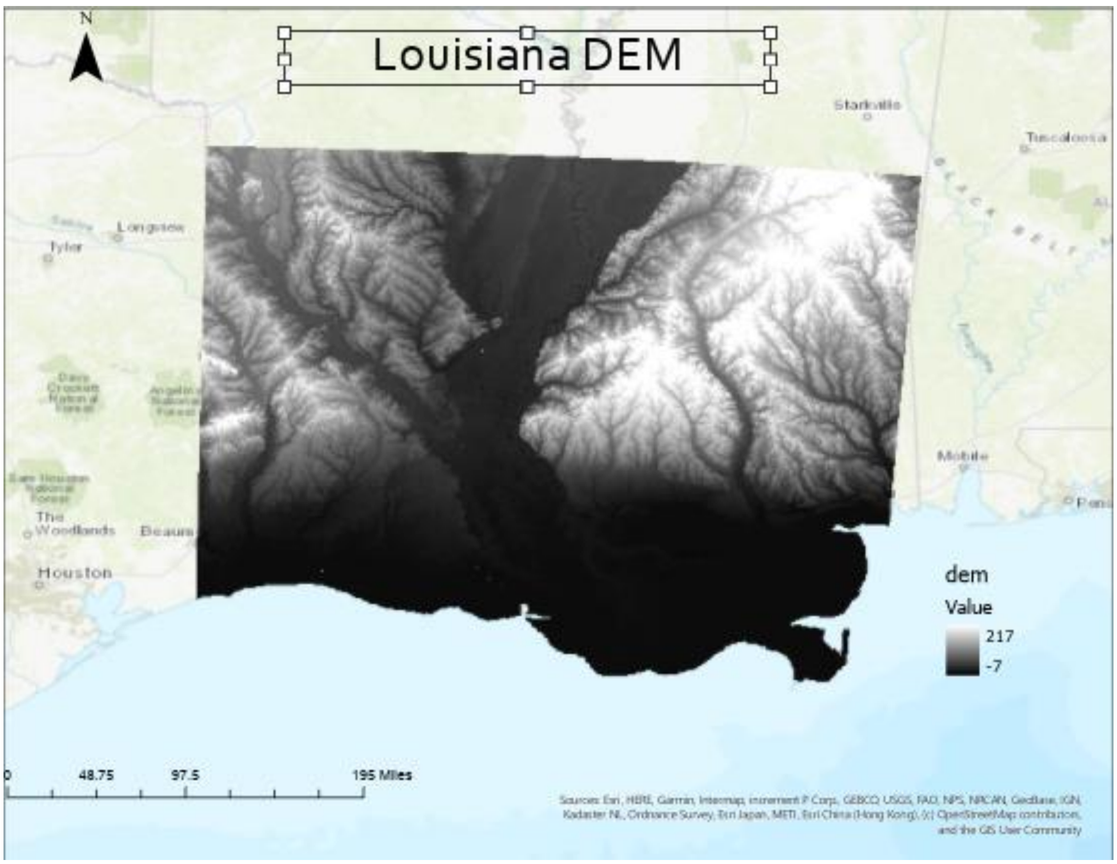

Figure 2.Louisiana DEM and site for my project.

Figure 3. Coordinates for Hurricane Katrina

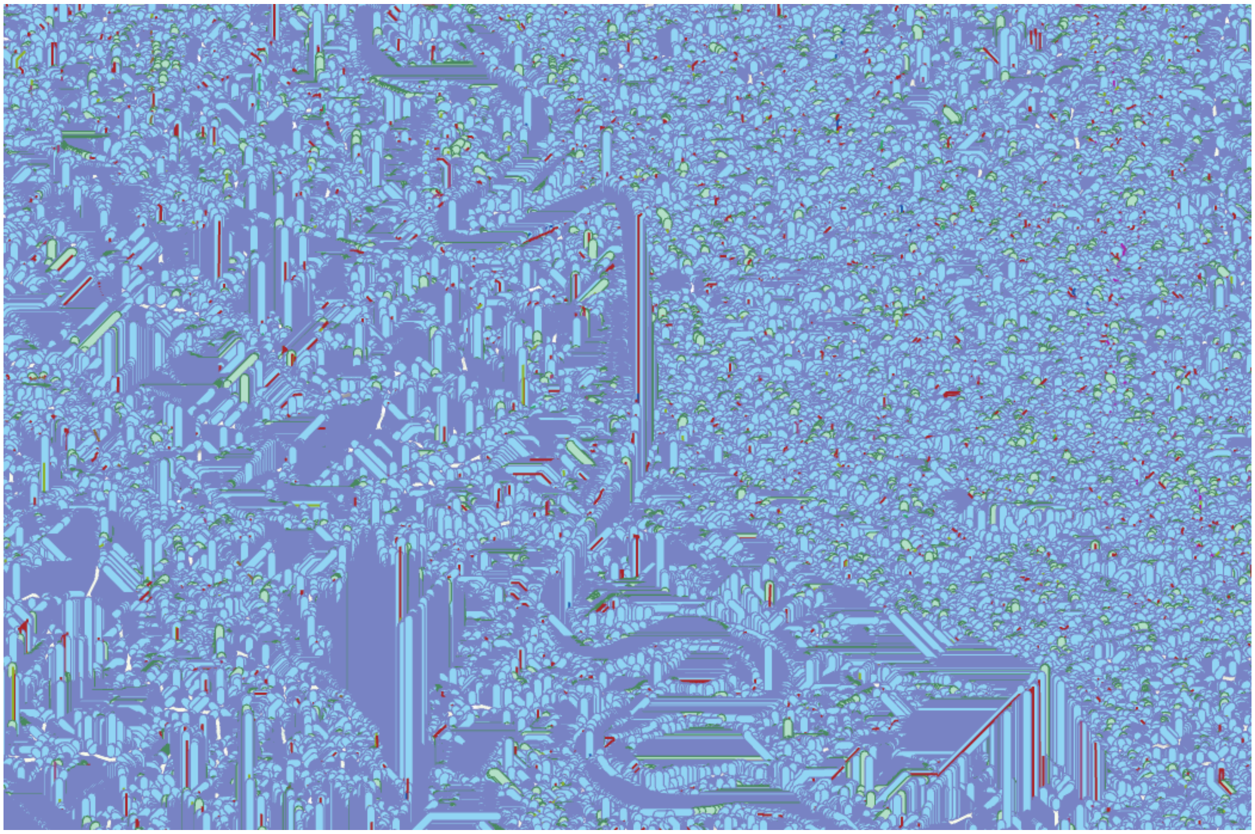

Figure 4. Stream to Features Map…The reason for the multiple colors is that the symbology has been separated. Thicker lines mean larger stream features while the thin lines mean smaller streams.

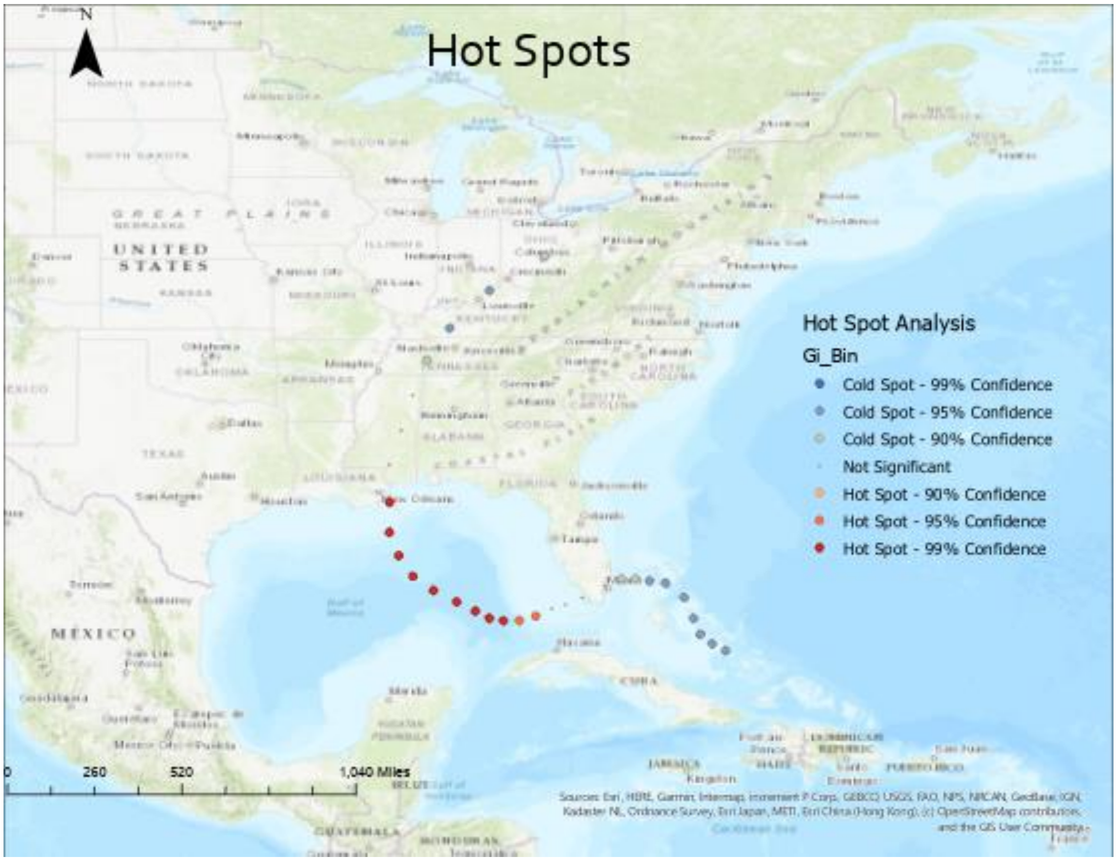

Figure 5. Hot Spot Analysis of the Wind Speeds of Hurricane Katrina as the storm progressed and degressed

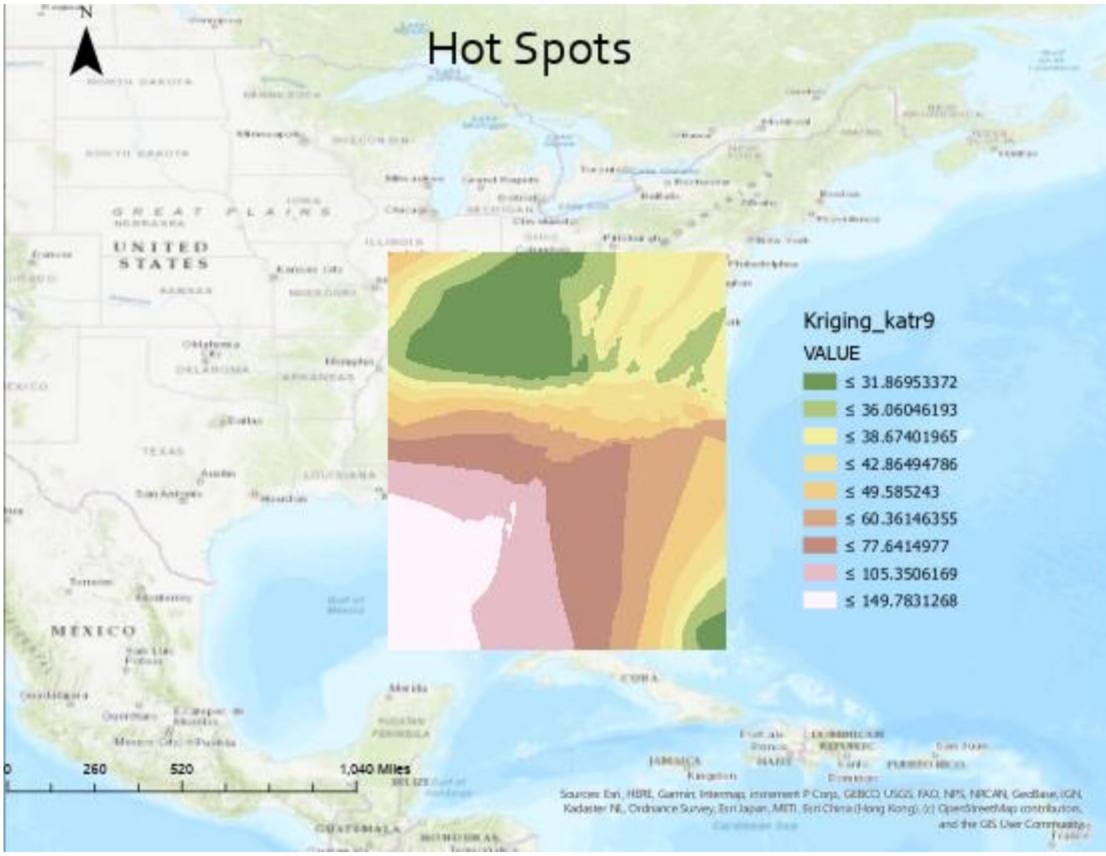

Figure 6. Kriging Variables of Wind speeds during Hurricane Katrina

CODE

```python
# -*- coding: utf-8 -*-

import arcpy
from arcpy import env
from arcpy.sa import *

class Toolbox(object):
    def __init__(self):
        """Define the toolbox (the name of the toolbox is the name of the
        .pyt file)."""
        self.label = "Katrina Tool"
        self.alias = ""

        # List of tool classes associated with this toolbox
        self.tools = [FloodAnalysisTool, HurricaneTool]


class FloodAnalysisTool(object):
    def __init__(self):
        """Define the tool (tool name is the name of the class)."""
        self.label = "FloodAnalysisTool"
        self.description = ""
        self.canRunInBackground = False

    def getParameterInfo(self):
        """Define parameter definitions"""
        # DEM
        dem = arcpy.Parameter(
                name="dem",
                displayName="DEM",
                direction="Input",
                datatype="GPRasterLayer",
                parameterType="Required")
        # Output Filled DEM
        filled = arcpy.Parameter(
                name="filled",
                displayName="Output Filled DEM",
                direction="Output",
                datatype="DERasterDataset",
                parameterType="Required")
```

```
        # Output Flow Direction
        fdir = arcpy.Parameter(
                name="fdir",
                displayName="Output Flow Direction",
                direction="Output",
                datatype="DERasterDataset",
                parameterType="Required")
        # Output Flow Accumulation
        facc = arcpy.Parameter(
                name="facc",
                displayName="Output Flow Accumulation",
                direction="Output",
                datatype="DERasterDataset",
                parameterType="Required")
        # Output Raster Calculator
        rast_calc = arcpy.Parameter(
                name="rast_calc",
                displayName="Output Raster Calculator",
                direction="Output",
                datatype="DERasterDataset",
                parameterType="Required")
        # Output Stream Link
        stream_link = arcpy.Parameter(
                name="stream_link",
                displayName="Output Stream Link",
                direction="Output",
                datatype="DERasterDataset",
                parameterType="Required")
        # Output Stream Order
        stream_order = arcpy.Parameter(
                name="stream_order",
                displayName="Output Stream Order",
                direction="Output",
                datatype="DERasterDataset",
                parameterType="Required")
```

```
    #Stream to Feature
    stream_feature = arcpy.Parameter(
            name="stream_feature",
            displayName="Stream to Feature",
            direction="Output",
            datatype="DEFeatureClass",
            parameterType="Required")

    params = [dem, filled, fdir, facc, rast_calc, stream_link, stream_order, stream_feature]
    return params

def isLicensed(self):
    """Set whether tool is licensed to execute."""
    return True

def updateParameters(self, parameters):
    """Modify the values and properties of parameters before internal
    validation is performed.  This method is called whenever a parameter
    has been changed."""
    return

def updateMessages(self, parameters):
    """Modify the messages created by internal validation for each tool
    parameter.  This method is called after internal validation."""
    return

def execute(self, parameters, messages):
    """The source code of the tool."""
    dem = parameters[0].ValueAsText
    filled = parameters[1].ValueAsText
    fdir = parameters[2].ValueAsText
    facc = parameters[3].ValueAsText
    rast_calc = parameters[4].ValueAsText
    stream_link = parameters[5].ValueAsText
    stream_order = parameters[6].ValueAsText
    stream_feature = parameters[7].ValueAsText
    return
```

```
class HurricaneTool(object):
    def __init__(self):
        """Define the tool (tool name is the name of the class)."""
        self.label = "Hurricane Tool"
        self.description = ""
        self.canRunInBackground = False

    def getParameterInfo(self):
        """Define parameter definitions"""
        # GPS Coordinates
        katrina_xytabletopoint = arcpy.Parameter(
                name="katrina_xytabletopoint",
                displayName="Katrina Coordinates",
                direction="Input",
                datatype="GPFeatureLayer",
                parameterType="Required")
        # Hot Spots
        hotspot = arcpy.Parameter(
                name="hotspot",
                displayName="Katrina Hot Spots",
                direction="Output",
                datatype="DEFeatureClass",
                parameterType="Required")
        # Kriging
        kriging = arcpy.Parameter(
                name="kriging",
                displayName="Kringing",
                direction="Output",
                datatype="DERasterDataset",
                parameterType="Required")

        params = [katrina_xytabletopoint, hotspot, kriging]
        return params

    def isLicensed(self):
        """Set whether tool is licensed to execute."""
        return True
```

```
def isLicensed(self):
    """Set whether tool is licensed to execute."""
    return True

def updateParameters(self, parameters):
    """Modify the values and properties of parameters before internal
    validation is performed.  This method is called whenever a parameter
    has been changed."""
    return

def updateMessages(self, parameters):
    """Modify the messages created by internal validation for each tool
    parameter.  This method is called after internal validation."""
    return

def execute(self, parameters, messages):
    """The source code of the tool."""
    katrina_coords = parameters[0].ValueAsText
    hotspot = parameters[1].ValueAsText
    kriging = parameters[2].ValueAsText

    # Set local variables
    inPointFeature = 'Katrina_XYTableToPoint.shp'
    outVarRaster = "D:\Louisiana\kriging"

    # Create KrigingModelOrdinary Object
    lagSize = 0.160400
    kModelOrdinary = KrigingModelOrdinary("LINEAR", lagSize)

    # Check out the ArcGIS Spatial Analyst extension license
    arcpy.CheckOutExtension("Spatial")

    #Execute Kriging
    outKrigingOrd = Kriging(inPointFeature, "ws", kModelOrdinary, 0.1604, outVarRaster)
    return
```

.

Discussion

This project was very interesting due to the knowledge I gained from researching this topic. One thing I noticed during this project was looking at the stream to features tool showed multiple areas that had large stream segments in areas that were below sea level. Areas in the south region of Louisiana had larger stream segments in certain areas that could have caused areas to obtain more water and cause streams and other areas to have to much water in areas causing flooding. Obviously with the storm's storm surge, water was going to rise but it is interesting to see that in some areas that the larger stream segments were in the southern part of the state and that could cause help cause flooding in those areas more than others. One thing I think I could've done differently was creating a water surface elevation model to know what the difference in water elevations were at different times during the storm surge. Another idea would to find the wave heights to see how the water was pushed onto the leeves. An article I read claims, "The water surface elevation and significant wave heights show the effect of the bathymetry and the detail on the mesh in the sharp gradients of both surface elevation and significant wave heights over the reef chain that divides the Atlantic Ocean from the Virgin Islands shelf.” (Dynamic Modeling of Surface Runoff and Storm Surge during Hurricane and Tropical Storm Events, Silva-Araya, 3.3). I like these models that were created because it shows the significant elevation and wave heights. I believe if I would’ve known about these models and created them, it would only benefit the research I conducted during this project.

Introduction

To conclude, I think that if the Core of Engineers in Louisiana has not already tried to communicate and create ideas to better the levee systems to help cause minimal flooding, the need to start now. Katrina was one of the most catastrophic storms to ever hit America. The flood damage was something unthinkable and if models of Katrina and other hurricanes are created to show the wind speed and flood models to find streams and how much water is flowing through them, floods like what were caused by Katrina could happen again.

References

- Burns, Jordan. *Risk Assessment Tools from FEMA's Hazus Program*. Jan. 2020, www.urisa.org/clientuploads/directory/Documents/The%20GIS%20Professional/2020/JanFeb2020.pdf.
- Cummings, Christina A., et al. “Using the Hazus-MH Flood Model to Evaluate Community Relocation as a Flood Mitigation Response to Terminal Lake Flooding: The Case of Minnewaukan, North Dakota, USA.” *Applied Geography*, vol. 32, no. 2, 2012, pp. 889–895., doi:10.1016/j.apgeog.2011.08.016.
- Merz, B. *Assessment of Economic Flood Damage*. 18 Aug. 2010, gfzpublic.gfz-potsdam.de/rest/items/item_241074_1/component/file_241073/content.
- Nastev, Miroslav, and Nikolay Todorov. “Hazus: A Standardized Methodology for Flood Risk Assessment in Canada.” *Canadian Water Resources Journal*, vol. 38, no. 3, 2013, pp. 223–231., doi:10.1080/07011784.2013.801599.
- *Risk MAP CDS Hazus Tools and Data*. Nov. 2019, www.fema.gov/media-library-data/1585923923582-818d18af850789cc3dd353468d05e9ea/ReleaseNotes.pdf.
- Royal, Andrew, and Margaret Walls. “Flood Risk Perceptions and Insurance Choice: Do Decisions in the Floodplain Reflect Overoptimism?” *Risk Analysis*, vol. 39, no. 5, 2018, pp. 1088–1104., doi:10.1111/risa.13240.
- Scawthorn, Charles. “HAZUS-MH Flood Loss Estimation Methodolgoy. I: Overview and Flood Hazard Characterization.” May 2006.
- Silva-Araya, Walter, et al. “Dynamic Modeling of Surface Runoff and Storm Surge during Hurricane and Tropical Storm Events.” *Hydrology*, vol. 5, no. 1, 2018, p. 13., doi:10.3390/hydrology5010013.
- Silva, Lucas Borges Leal Da, et al. “Multidimensional Flood Risk Management under Climate Changes: Bibliometric Analysis, Trends and Strategic Guidelines for Decision-Making in Urban Dynamics.” *International Journal of Disaster Risk Reduction*, vol. 50, 2020, p. 101865., doi:10.1016/j.ijdrr.2020.101865.
- Tate, Eric, et al. “Uncertainty and Sensitivity Analysis of the HAZUS-MH Flood Model.” *Natural Hazards Review*, vol. 16, no. 3, 2015, p. 04014030., doi:10.1061/(asce)nh.1527-6996.0000167.