AN AUTOMATED APPROACH FOR CALCULATING ENVIRONMENTAL IMPACTS OF TRANSMISSION LINE CONSTRUCTION USING PYTHON

A THESIS PRESENTED TO
THE DEPARTMENT OF HUMANITIES AND SOCIAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
MASTER OF SCIENCE

By
KYLE J. GALLAGHER

NORTHWEST MISSOURI STATE UNIVERSITY
MARYVILLE, MISSOURI
OCTOBER 2014

An Automated Approach for Calculating Environmental Impacts of Transmission Line Construction Using Python

Kyle J. Gallagher

Northwest Missouri State University

THESIS APPROVED

Thesis Advisor, Dr. Yanfen Le Date

Dr. Yi-Hwa Wu Date

Dr. Patricia Drews Date

Dean of Graduate School, Dr. Gregory Haddock Date

An Automated Approach for Calculating Environmental Impacts of Transmission Line Construction Using Python

ABSTRACT

The transmission line construction permitting process involves the requirement set forth by state and federal agencies that electric utility companies take into consideration the environmental impacts to sensitive areas such as wetlands and document the impact calculations within the permit. Arriving at the impact calculations with the use of GIS software can be a cumbersome process, entailing numerous geoprocessing steps done in a specific order to achieve accurate results. Accurate impact calculations are of extreme importance to the electric utility, who must attempt to minimize impacts and be a good steward of the environment, and to the regulatory permitting agencies, who ultimately sign off on the construction after specific conditions are met by the utility.

The study examines the possibility of constructing an automated Python script solution that takes a set of input dataset parameters, creates all applicable sensitive area datasets, calculates all impact calculations, and outputs all calculations to a set of pre-formatted Excel tables that are ultimately included in the permit application. This solution helps eliminate errors caused from manual execution of the workflow and allows for quick regeneration of impact calculations if input dataset parameters (e.g., design) are to change. A 13-mile, 138 kV transmission line rebuild project within the State of Maryland is used as the case study. Impacts to remove existing electric poles and construct new poles are calculated. While desired results are achieved, the solution is not fully automated due to a few minor tasks that are not easily achievable with Python code. The study highlights the ability of ArcGIS, Python, and Excel to talk to each other and exchange data, and presents a platform that could be used for other forms of spatial analyses.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

B&V	Black & Veatch
CAA	Critical Area Act
CEA	Cumulative Effects Assessment
CWA	Clean Water Act
CPCN	Certificate of Public Conveyance and Necessity
EPA	Environmental Protection Agency
EMC	Electric Membership Corporation
ERD	Environmental Review Document
Esri	Environmental Systems Research Institute
GIS	Geographic Information System
GPS	Global Positioning System
MD DNR	Maryland Department of Natural Resources
NAIP	National Agricultural Imagery Program
NC-CREWS	North Carolina Region Evaluation of Wetland Significance
NYPA	New York Port Authority
PHI	Pepco Holdings, Inc.
ROW	Right-of-Way
UTM	Universal Transverse Mercator
VEC	Valuable Ecosystem Component
WSSC	Wetlands of Special State Concern

ACKNOWLEDGMENTS

First and foremost, I would like to thank my wife Anni and daughter Norah for their everlasting support and sacrifice throughout this process. To my Black & Veatch team members, Scott McBurney and Jody Lima, thanks for helping me understand the complete picture of the environmental permitting and regulatory processes. Also thanks to Dana Small and Matt Savage of Pepco Holdings, Inc. for their approval and support of this research. To my thesis advisor, Dr. Le, and thesis committee members, Dr. Drews and Dr. Wu, I appreciate all of your guidance and feedback throughout this journey.

Chapter 1: Introduction

Wetlands play a valuable role in any ecosystem, helping to reduce flood damage, providing vital fish and wildlife habitats, preserving vegetation, and helping to improve water quality by filtering sediment, nutrients and pollutants. When activities such as transmission line construction take place, every effort should be made to preserve wetlands as much as possible by limiting direct impacts (EPA 2001). There are many national and state regulations in place to govern activities within wetlands, including Section 404 of the Clean Water Act (CWA) – Wetlands Compliance Monitoring (EPA 2009), the State of Maryland's Nontidal Wetlands Protection Act (Walbeck *et al.* 2011), and the State of Delaware's Wetland Act of 1973 (State of Delaware 2009). These regulations provide extensive guidance on how wetlands should be monitored and protected, and what kinds of impacts should be documented for potential construction activities.

Vegetation management is another activity performed by utility companies that manage right-of-way (ROW). It often comes with very large operational expenses (Kelly 2008) and represents a large portion of annual budgets. Reliability and affordability have been popular topics within the utility industry in recent years, especially in regard to vegetation management impact assessments (Chappell 2007).

Impacts to wetlands can be minimized with the use of matting that exerts minimal downward pressure from heavy construction vehicles and equipment. The use of matting in turn can help utility companies minimize their costs for wetland restoration and post-construction monitoring activities (Electric Light & Power 2010), while promoting proactive and responsible environmental stewardship (Pepco Holdings, Inc. 2010).

Impacts generally fall into one of two categories: temporary or permanent. Temporary impacts occur when mats are placed on top of herbaceous vegetation that will revegetate naturally. Permanent impacts occur when mats need to be placed in areas consisting of shrubs or brush that must be cleared prior to placement of mats. The impacts related to the removal or installation of utility poles are also included in permanent impact calculations (Black & Veatch 2011). While eliminating all impacts is usually an unattainable goal, every effort should be made to restore temporary impacts and minimize permanent impacts (Pepco Holdings, Inc. 2010). GIS offers the spatial analysis tools to calculate all the necessary impacts that a utility will need in order to acquire permitting for construction, execute construction plans, and limit operational expenses.

An impact occurs when matting is placed in wetlands, wetland buffers, Critical Area buffers, or riparian buffers. Wetland and watercourse boundaries are delineated in the field using stakes which are then surveyed using a high-accuracy GPS unit. Wetland and riparian buffers can then be created based on the final wetland and watercourse boundaries, respectively. In addition to protecting wetlands, the State of Maryland also has measures in place to protect the Chesapeake Bay Critical Areas. The Critical Area Act (CAA) of 1984 defined "Critical Areas" as "land within 1,000 feet of the Mean High Water Line of tidal waters or the landward edge of tidal wetlands and all waters of and lands under the Chesapeake Bay and its tributaries" (MD DNR nd). The Critical Area Commission (CAC) was formed to oversee development within these areas to ensure minimal environmental impacts and to conserve fish, wildlife, and plant habitats (MD DNR nd). Within Critical Areas, the CAA requires a 25 foot buffer around all non-tidal wetlands unless classified as Wetlands of Special State Concern (WSSC), which require a

100 foot wetland buffer (MD DNR 1997).

The access roads within the ROW provide the basis for laying mats within the aforementioned protected areas. Each mat is an 8x14 foot piece of wood or plastic composite material (Black & Veatch 2011) that can bear loads up to 300,000 lbs. and is typically constructed around the pole to allow enough room for construction vehicles and machinery (Black & Veatch 2011). Figure 1 shows a layout of mats along an access road for a transmission line rebuild project in New Jersey. Figure 2 provides an overview example of how the matting would be laid out within a wetland area.

This research project builds upon lessons learned from a project undertaken at Black & Veatch (B&V). B&V was tasked by Pepco Holdings, Inc. (PHI), an electric utility that operates in the Mid-Atlantic region, with producing an

Figure 1. Matting Used for Transmission Line Construction (source: Black & Veatch)

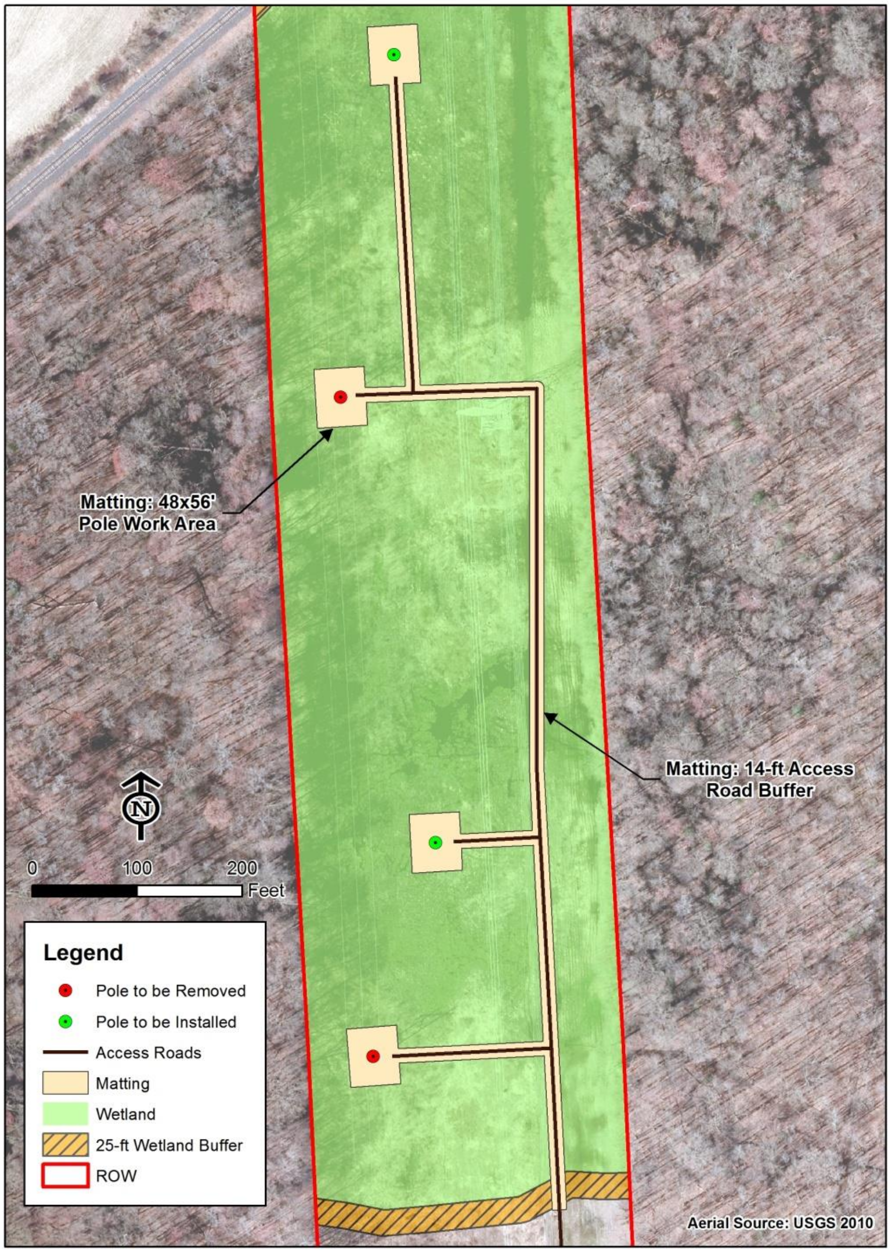


Figure 2. Matting Layout Example

Environmental Review Document (ERD) and an Individual Permit (IP) associated with the future rebuild of a 13-mile, 138kV transmission line within the State of Maryland. The ERD covered a wide array of potential environmental impacts associated with the future rebuild and provided the necessary environmental documentation for PHI to apply through the State of Maryland for a Certificate of Public Convenience and Necessity (CPCN) in order to undertake construction (Black & Veatch 2011). GIS was depended on extensively to provide, via Microsoft Excel tables, numerous impact calculations for wetlands, wetland buffers, and the defined construction ROW. The data were derived from a combination of Critical Area buffers, riparian buffers, and vegetation areas of various types in and around field activities, including a wetland delineation and vegetation assessment, and environmental GIS datasets available from the State of Maryland. The resulting impact calculations were detailed throughout the ERD and IP in text and tables. Many of the processes for producing impact calculations required frequent geoprocessing steps such as clips, intersects, and erases, all of which produced a large amount of intermediate data and overall was a very cumbersome process to undertake manually. Careful attention had to be given to the sequence of the steps in order to accurately produce the desired outcomes. In order to minimize time spent coming up with impact calculations, this research looked at creating a semi-automated solution that used a combination of ArcGIS ModelBuilder and the Python programming language to produce script tools to be used in ArcToolbox. With the use of user-input parameters, the script tools can quickly produce all necessary impact calculations, and deliver the results through Excel tables that can be easily transferred to the relative sections of the ERD or IP.

1.1 Research Objective

Wetland and vegetation impact calculations are a vital component of any environmental impact assessment or permitting process (Walbeck *et al.* 2011) and, in the case of an electric utility, can have cost implications for construction activities. The objective of this study was to construct, verify, and validate a set of Python scripts that take user-provided input parameters, process all necessary environmental impact calculations, and deliver the results in an Excel table. By leveraging the automation capabilities of Python, all calculations and necessary re-calculations of environmental impacts can be performed quickly and accurately simply by providing the necessary inputs. This will provide the utility companies with peace of mind that the calculations are accurate, dependable, and flexible in the event that any construction plan changes are made, and ultimately will help minimize construction expenses and environmental impacts.

1.2 Justification

There are numerous impact calculations produced during wetland and vegetation assessments, some of which may require many geoprocessing tasks to be performed in specific sequences. When inputs are altered due to an engineering design modification (e.g., a proposed pole location is moved), the calculations must be reproduced, placing a heavy emphasis on performing the manual geoprocessing tasks in the correct sequence in order to achieve accurate results. This can be a cumbersome process when performed manually, requires a high level of attention to detail, and can have drastic consequences if performed incorrectly. Furthermore, B&V has produced multiple ERD reports and permits in the State of Maryland and is currently scheduled to produce additional reports in the near future. Given that the same impact calculations are performed for each report or permit, the proposed research presents an opportunity to establish a template that can be reused for future use and potentially be modified for the requirements of other types of construction permits.

Chapter 2: Research Background

2.1 GIS for Environmental Impact Calculations

The initial literature review produced minimal academic research specific to the proposed research as there appears to be a lack of GIS components when it comes to environmental impact modeling and analysis. According to the GEOBASE bibliographical database, between January 1990 and February 2003, there were 1,360 references on Environmental Impact Assessments (EIAs), and only 58 (4.2%) incorporated GIS technology (Gunasekera 2004). Additionally, a LexisNexis search for "National Environmental Policy Act" from 1969-2010 netted 4,673 results, of which 1,007 (21.5%) referenced cumulative effects/impacts, and only 11 (1%) of the 1,007 referenced GIS (Atkinson and Canter 2011). The assumption could be made that since the use of GIS technology has greatly expanded since 2003 that these statistics are much higher. The example applications presented in the reviewed journal articles, however, did have a similar tone to this research paper as there were many references to popular terms such as "overlay" and "buffer zones" (Davidovic *et al.* 2010, Herrero-Jimenez 2012, Atkinson and Canter 2011). The review also revealed that numerous electric utility companies are utilizing the vast capabilities of GIS to undertake environmental management activities that are very similar to the tasks carried out by this research paper (Pepco Holdings, Inc. 2010, Kelly 2008, and Meehan 2007).

Atkinson and Canter (2011) explored how GIS technology was being used for environmental impact assessments, particularly in preparation of cumulative effects assessments (CEAs) and impact audits. It was noted that with the significant increase in GIS technology for environmental analysis over the last few decades, there has also been a steady increase in the types of solutions to carry out the analysis, such as models, scripts, algorithms, and visual applications. Map overlays, geoprocessing tools, and spatial statistic computations can all be used to calculate environmental impacts to Valuable Ecosystem Components (VECs), such as wetlands and threatened and endangered species habitats, and compare the numbers to certain thresholds of significance. A reviewed study by Muller *et al.* (2007) included an area-wide CEA (ACEA) conducted for the Denver Regional Council of Governments to compute predicted impervious areas by land use category up to the year 2020. Overlay-intersect GIS procedures were used to compute the areas, which were then transferred to an Excel file as a summary.

Andrews (1990) looked at the varying environmental impacts of access roads and utility corridors. It was noted that for each kilometer of transmission and distribution line within the U.S., approximately 61-98 acres of land were being compacted, representing a significant disturbance ratio caused by the roads. The access roads tend to isolate certain species on an “island” as their movement becomes limited due to the access roads acting as a barrier. Soil compaction can also be affected by construction activities such as clearing, leveling, and cut and fill, which in turn directly affect revegetation efforts and can lead to watercourse erosion. The impacts of these kinds of activities become magnified when wetlands are bisected by the access roads. Similarly, this research used pre-determined access roads that presented the least amount of disturbance to protected areas such as wetlands and Critical Areas.

A Microstation Geographics® GIS platform was used by Herrero-Jimenez (2012) to build a prototype that could identify and assess environmental impacts of engineering projects based on declarative and procedural knowledge bases. The declarative knowledge base contained spatial datasets of existing environmental conditions whereas the procedural knowledge base contained programming algorithms or processes used to analyze and calculate environmental impacts. The algorithms analyzed topological overlays of points, lines, and polygons, and produced intersections, unions, and buffer zones. It was determined that by defining the specific locations of project activities, such as access routes, the environmental impacts could be identified and represented much more thoroughly. A key note from Herrero-Jimenez's study was that it used pre-defined access routes to aid in the calculation of environmental impacts, which is also a requirement for the process being proposed in this study. In addition, the overlays used in Herrero-Jimenez's analysis are very similar to the geoprocessing tasks that were utilized to analyze the environmental impact calculation process developed in this study.

Various GIS datasets were overlaid and analyzed through a defined set of indicators to minimize environmental impacts of tourism at National Park Berchtesgaden in Bavaria, Germany. Limits of protected natural resource areas and their buffers, along with the boundaries of tourism areas and their corresponding carrying capacities as defined by the European Union were analyzed to show where tourism activities can be limited or promoted in order to minimize negative environmental impacts (Davidovic *et al.* 2010). In this study, the natural resource areas under protection were represented by wetlands, watercourses, and Critical Areas, all of which had a corresponding buffer zone used for impact analysis.

Flood risk impact analysis of several states and Local Government Areas (LGAs) within Nigeria was performed based on the October 2012 flooding of the Niger-Benue basin (Nkeki *et al.* 2013). A combination of Moderate Resolution Imaging Spectroradiometre (MODIS) satellite data, topographic data, hydrographic data, and population data were utilized in the spatial analysis. Numerous geoprocessing tools such as intersects and erases, along with several overlaying techniques, were used to generate low, medium, and high risk flood areas by state, LGA, and watershed. Population data was then factored into the analysis in order to generate reports and charts of the percent of submerged area by state and LGA. Ultimately, by identifying the areas of Nigeria that would experience the most impact from a flood, decision makers would be better prepared to allocate emergency resources. Much like flood risk areas were analyzed in conjunction with population data using a series of geoprocessing tools to create reports, the study undertaken in this paper looked at sensitive environmental areas and how construction impacts affected each area while using many of the same tools and creating similar reports to be used by decision makers (Nkeki *et al.* 2013).

GIS is also being utilized for wetland risk assessments. The North Carolina Coastal Region Evaluation of Wetland Significance (NC-CREWS) is a GIS-based assessment tool for evaluating and assigning risk ratings to state wetlands based on a wide variety of activities that may produce “lost” wetlands, such as dredging and filling for transmission line poles. The tool was developed to comply with Section 404 wetland permit applications. A total of 39 parameters are used for assigning risk ratings and include characteristics of watersheds, wetlands, water quality, soils, and habitats (Sutter *et al.* 1999).

2.2 GIS for Vegetation Management

Utility companies are turning to GIS to aid their operations in the areas of planning, environmental compliance, and construction monitoring. PHI (2010) employs the spatial analysis capabilities of GIS to assess habitat and species diversity and sensitive environmental areas (e.g., wetlands) within their regulated ROWs in order to establish appropriate access points, access roads, and construction equipment areas that will avoid these sensitive areas. PHI also uses GIS extensively for vegetation management plans by mapping all relative components of their ROWs and assigning the affected areas specific vegetation management recommendations that can be analyzed visually.

Habersham Electric Membership Corporation (EMC), a Georgia co-op that operates over 1,800 miles of overhead transmission lines, used to track ROW line-clearing activities manually using a paper wall map (Kelly 2008). This setup produced a high risk for the co-op in regard to losing details about ROW activities, which in turn could have led to poor customer service, a failure to comply with environmental regulations, and an overall lack of system reliability. In order to measure activity progress and analyze crew productivity, an automated procedure had to be adopted. Using the ArcGIS Engine Developer Kit, Habersham EMC implemented the Vegetation Management Solution, which receives inputs such as land base and electrical component datasets that allow for easy tracking, planning, and analysis of ROW line-clearing activities. This type of automated setup is helping Habersham minimize their operational expenses and maximize their return on investment (Kelly 2008).

The New York Power Authority (NYPA) developed an integrated vegetation management application using ArcGIS to help oversee activities of their approximately 16,000 acres of ROW. The application uses a simple interface to perform complex tasks that aid in planning efforts of ROW activities while complying with federal and state regulations (Meehan 2007).

2.3 Current Technology Capabilities

Esri's (2011b) ModelBuilder provides an easy-to-use platform for building workflows that allow for the stringing together of geoprocessing tools, where one tool's output is an input to the next tool. A model can then be exported to a Python script file (.py) where further code modifications can be made. A Python script can be imported to ArcToolbox as a geoprocessing tool where parameters can be defined to allow for dynamic user inputs. Kauffman (2007) used the ModelBuilder to Python approach in developing automated tools to prioritize thousands of wetlands in Oregon that considered numerous parameters using a weighted average system.

The ability to export table data from geospatial datasets to Excel tables is available through Esri's (2010) Table to Excel Python script tool that works with ArcGIS 10, and through out-of-the-box export tools built using the xlrd and xlwt Python modules that are now available with ArcGIS 10.2 (Esri 2013). The xlwt module can also be leveraged outside of the ArcGIS platform to allow for generating new and modifying existing Excel tables, and can be imported into a Python script in order to access specific classes and functions that include writing data to specific cells, cell formatting (Examples Generating Excel Documents 2011), borders, and formulas (Bernier 2009). The xlwt module also offers the added flexibility of customizing what and how data is output to an Excel file (Avraam 2009). Given that Esri's out-of-the-box scripts will only output the entire table of a geospatial dataset (Esri 2010), using the xlwt module outside of the ArcGIS platform presents the better option in order to customize how data should be outputted and what cells they should be outputted to.

The Python programming language presents a very simple and flexible platform for executing the impact calculation process in this research. Since the release of ArcGIS 10, the ArcPy module and the Python language as a whole have been tightly integrated for the purpose of carrying out geoprocessing tasks (Python 2012). This has helped provide a number of coding examples and other support documentation available on the Internet, which can be used to check proper usage, syntax, and structure. For the purposes of this research, Python's xlwt module presented a flexible solution for interfacing with ArcGIS by outputting data from geoprocessing tasks to new or existing Excel tables (Pinero 2010). The following are capabilities provided by the xlwt module that relate to this research:

- Ability to write tables to multiple worksheets within one Excel file
- Cell formatting: borders, alignment, font type/style, number formats (decimal places/percentage symbol)
- Cell formulas

Chapter 3: Methodology

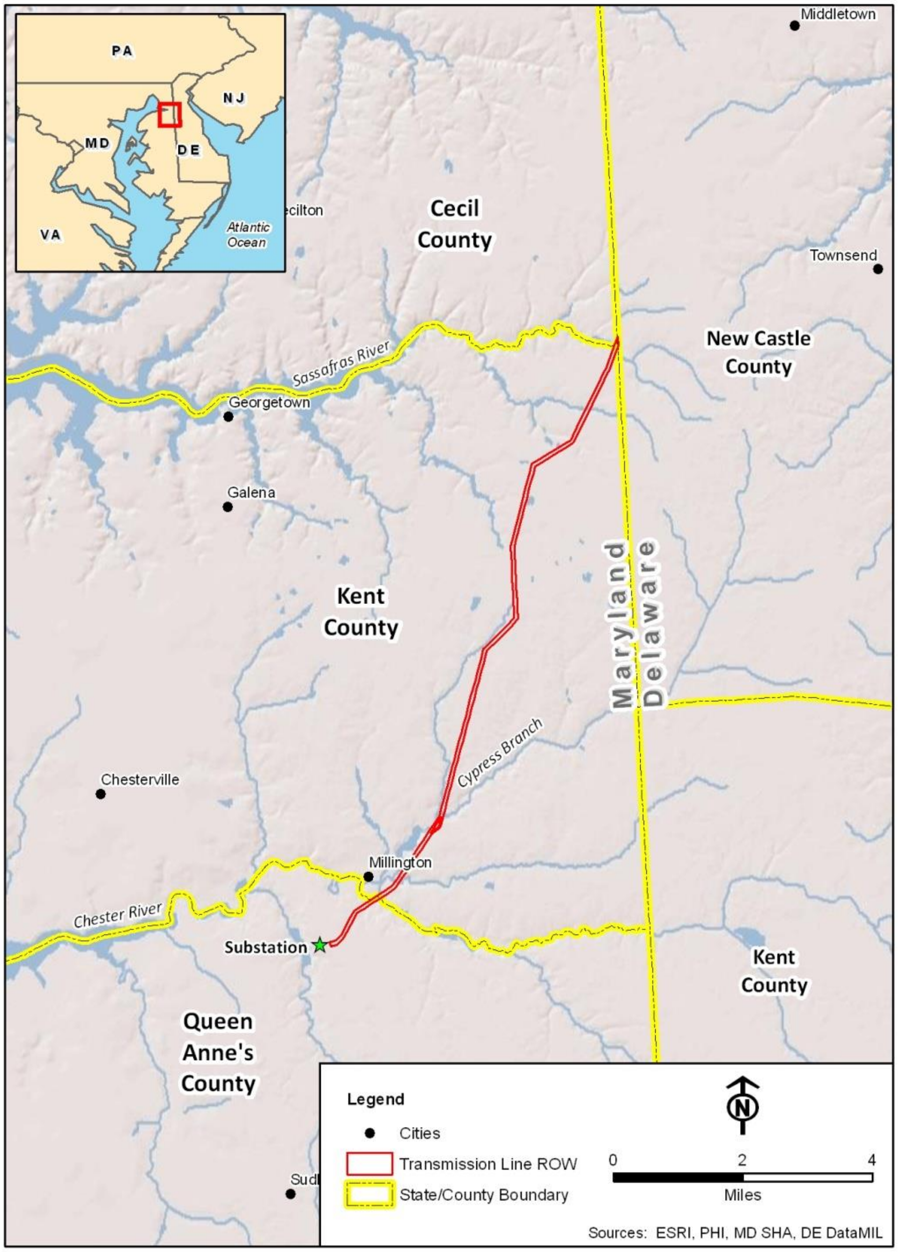
3.1 Study Area

The study area for testing the impact calculation scripts was confined to an existing 138kV transmission line within the Delmarva Peninsula, running from the Maryland/Delaware state line near the Cecil/Kent County, MD border, southwest to a substation outside of Millington, MD (Figure 3). The line is approximately 13 miles long and contains non-tidal wetlands from the Sassafras River, Upper Chester River, and Cypress Branch watersheds. The line will be rebuilt with new single monopoles being proposed 65 feet to the east of current H-frame structure locations. The analysis took into account the impacts of accessing poles, removing current poles, and installing new poles that are located within wetlands or buffers. Impacts were constrained to the transmission line's ROW boundary, which served as the primary area of interest. A ¼-mile ROW buffer (outside only) presented the secondary area of interest, where area calculations on State-regulated wetlands and their associated buffers were generated.

3.2 Description of Data

The following datasets were used in this research:

- Right-of-Way (ROW) (Pepco Holdings, Inc. 2011)
- Existing pole locations (Pepco Holdings, Inc. 2011)
- Proposed pole locations (Pepco Holdings, Inc. 2011)
- Access road centerlines (Pepco Holdings, Inc. 2011)
- Field-delineated Wetlands (Black & Veatch and True Measure Consulting, 2011)



Sources: ESRI, PHI, MD SHA, DE DataMIL

Figure 3. Study Area Map

- Field-delineated Vegetation Cover (Black & Veatch, 2011)
- State Wetlands for Cecil, Kent, and Queen Anne's County, MD (Maryland Department of Natural Resources, 1993)
- State Wetlands of Special State Concern (WSSC) (Maryland Department of Natural Resources, 1998)
- Chesapeake Bay Critical Areas for Cecil and Kent County, MD (Maryland Department of Natural Resources, 2011)
- Chesapeake Bay Critical Areas for Queen Anne's County, MD (Queen Anne's County, MD, 2011)

Although access road centerlines may fall inside or outside the ROW, impact calculations only considered roads inside the ROW. The field delineated wetland and watercourse datasets provided by B&V were reviewed and approved for use by field personnel who conducted the delineations. The Queen Anne's County Critical Areas dataset was not available from the MD DNR website and had to be obtained from the county's Planning and Zoning department. All datasets were projected to a custom UTM Zone 18N coordinate system (feet) used internally by PHI.

3.2.1 Required Data Fields

The following is a breakdown of the datasets that have required data fields necessary to carry out the impact calculation scripts:

Table 1. Required Data Fields

Dataset: Proposed pole locations		
Required Fields:		
Name	**Type**	**Purpose**
Diameter	Double	Poles at turns have 8.5-ft diameters, all others are 3.5-ft diameter; will affect permanent impact calculations

Dataset: Field-Delineated Wetlands		
Required Fields:		
Name	**Type**	**Purpose**
Num	Short	Represents the wetland complex ID number
Type	Text	Set to "WL" to distinguish as wetland when merged with other sensitive area datasets
Shape_Area	Double	Area of wetland complex in square feet, used in wetland and wetland buffer impact calculation tables

Dataset: Field-Delineated Watercourses		
Required Fields:		
Name	**Type**	**Purpose**
Num	Short	Represents the watercourse complex ID number
Type	Text	Set to "WC" to distinguish as watercourse when merged with other sensitive area datasets
Shape_Area	Double	Area of watercourse complex in square feet, used in watercourse/riparian buffer impact calculation table

Dataset: Vegetation Cover		
Required Fields:		
Name	**Type**	**Purpose**
Veg_Class	Text	Vegetation class type, used for vegetation impact calculations table
Shape_Area	Double	Area of vegetation area in square feet

3.3 Overview

The solution to this research is broken into two Python scripts with a minimal amount of manual effort in between the execution of each script. A basic workflow of the entire process is shown in Figure 4. The first script, documented in Section 3.5.1, takes the required datasets described in Section 3.2 and creates all buffers, access road matting, and a 48’ x 56’ area of matting centered on each existing and proposed pole that falls within an environmentally sensitive area. The manual effort in between scripts, documented in Section 3.5.2, includes:

1. Extending the access road matting towards the upland side of a sensitive area to ensure complete coverage
2. Properly orienting the pole matting areas to be parallel with the direction of the ROW, or to a “best-fit” location based on surrounding sensitive areas.

The second script, documented in Section 3.5.3, calculates all impacts and other relevant statistics and outputs the results to an Excel file consisting of tables for each of the following:

Within ROW:

- Field-delineated Wetlands
- Combined 25-ft Field-Delineated Wetland and Wetland in Critical Area Buffers
- Watercourse & 25-ft Riparian Buffer
- Vegetation Cover

Within ¼-mile ROW buffer (outside ROW only):

- State Wetlands
- 25-ft State Wetlands Buffer
- Wetlands of Special State Concern (WSSC)
- 100-ft WSSC Buffer

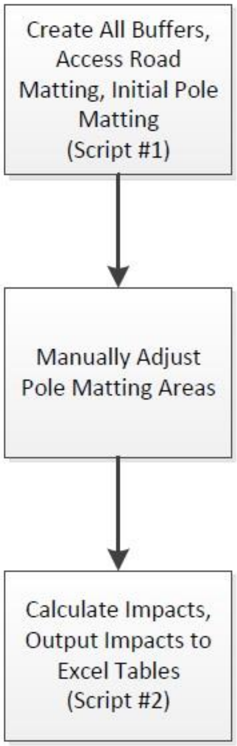


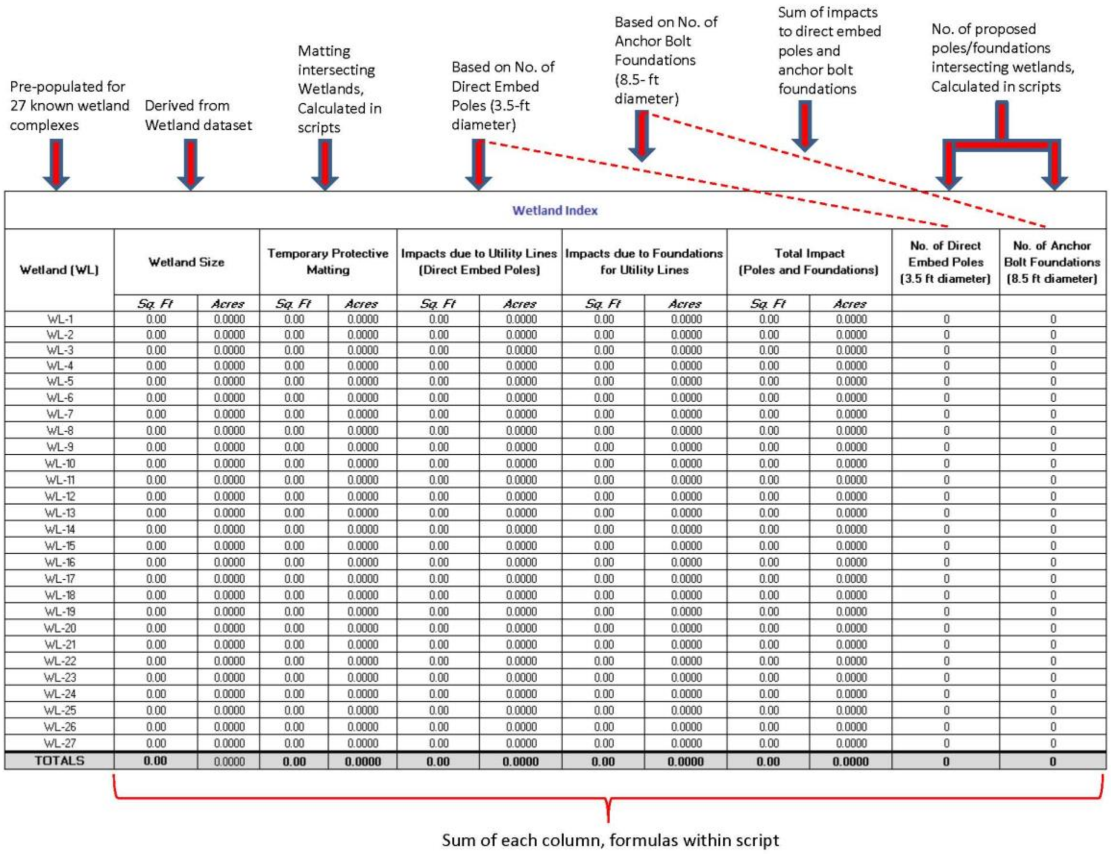
Figure 4. Basic Process Workflow

3.4 Output to Excel Tables

The format of each of the Excel tables is predefined since they had previously been manually created. What numbers are output to which cells, and any subsequent calculations that take place as a result, are all carried out by the script. It should be noted that all square feet values carry two decimal places, all acres values carry four decimal places, and the number of poles/foundations are whole numbers. All values are manually set to zero prior to running the script, as the script will only populate the appropriate rows for wetlands with impacts, leaving wetlands without impacts set to zero. Figure 5 provides an overview of the impact calculation table used for wetlands, and how each column is to be calculated. The same table setup is used for the combined 25-ft wetland and wetland in Critical Area buffers, and the combined watercourse and 25-ft riparian buffer.

The tables used for the vegetation cover and miscellaneous calculations have different setups. The vegetation cover table, shown in Figure 6, shows a breakdown of each vegetation cover class's total area, temporary impacts, and how many existing and proposed structures lie within the particular class. This setup provides a full account of the entire ROW and allows PHI to make more informed decisions as it relates to vegetation management activities that will come up during the structure removal and installation phases of construction.

The miscellaneous calculations table, shown in Figure 7, is not as extensive as the other tables as it represents simple, straightforward calculations related to field-delineated wetlands, Critical Areas, State wetlands, and State Wetlands of Special State Concern. These calculations are used in various sections of the ERD document to provide a general overview of areas within and outside the vicinity of the ROW.



Pre-populated for 27 known wetland complexes

Derived from Wetland dataset

Matting intersecting Wetlands, Calculated in scripts

Based on No. of Direct Embed Poles (3.5-ft diameter)

Based on No. of Anchor Bolt Foundations (8.5- ft diameter)

Sum of impacts to direct embed poles and anchor bolt foundations

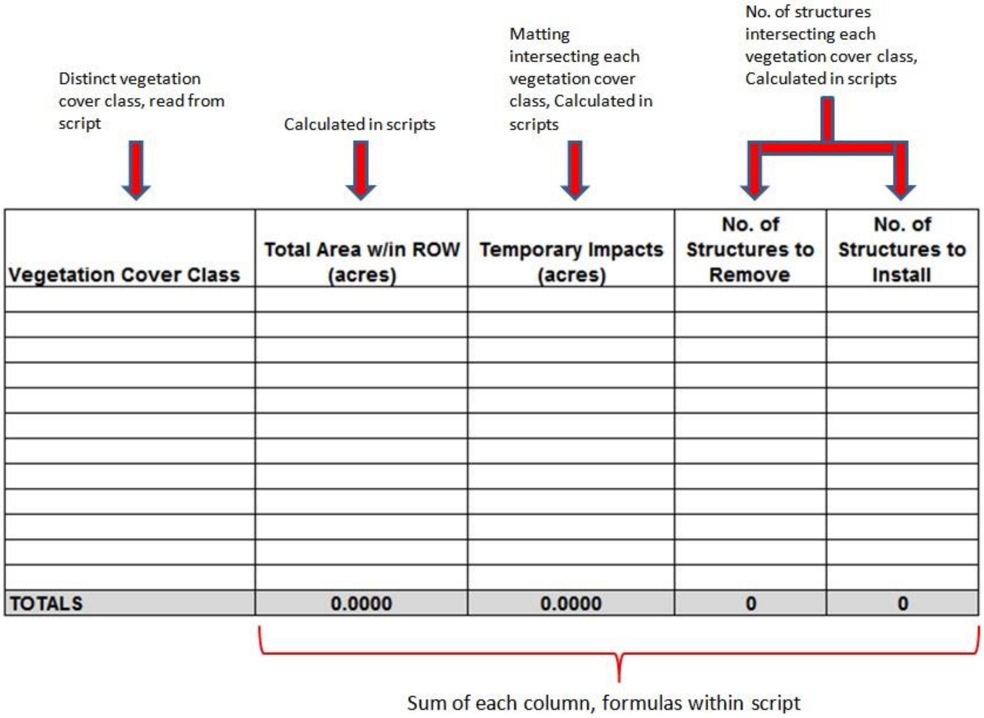
No. of proposed poles/foundations intersecting wetlands, Calculated in scripts

Wetland Index

Wetland (WL)	Wetland Size		Temporary Protective Matting		Impacts due to Utility Lines (Direct Embed Poles)		Impacts due to Foundations for Utility Lines		Total Impact (Poles and Foundations)		No. of Direct Embed Poles (3.5 ft diameter)	No. of Anchor Bolt Foundations (8.5 ft diameter)
	Sq. Ft	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*		
WL-1	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-2	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-3	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-4	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-5	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-6	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-7	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-8	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-9	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-10	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-11	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-12	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-13	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-14	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-15	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-16	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-17	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-18	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-19	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-20	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-21	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-22	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-23	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-24	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-25	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-26	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-27	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
TOTALS	**0.00**	0.0000	**0.00**	**0.0000**	**0.00**	**0.0000**	**0.00**	**0.0000**	**0.00**	**0.0000**	**0**	**0**

Sum of each column, formulas within script

Figure 5. Impacts Table Structure



Distinct vegetation cover class, read from script

Calculated in scripts

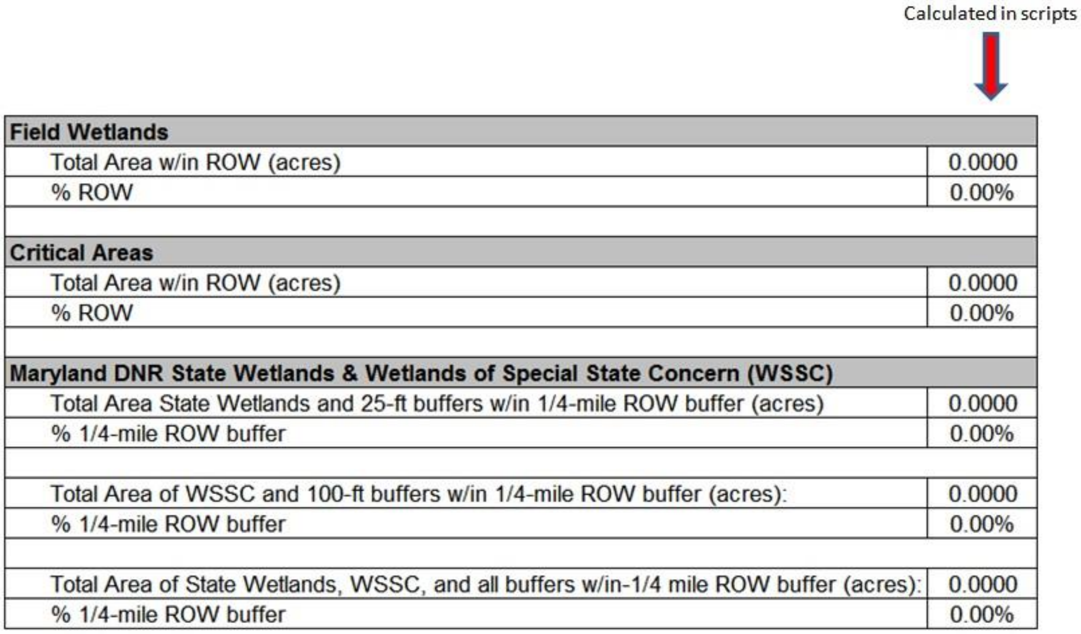
Matting intersecting each vegetation cover class, Calculated in scripts

No. of structures intersecting each vegetation cover class, Calculated in scripts

Vegetation Cover Class	Total Area w/in ROW (acres)	Temporary Impacts (acres)	No. of Structures to Remove	No. of Structures to Install
TOTALS	**0.0000**	**0.0000**	**0**	**0**

Sum of each column, formulas within script

Figure 6. Vegetation Impacts Table Structure



Calculated in scripts

Field Wetlands	
Total Area w/in ROW (acres)	0.0000
% ROW	0.00%
Critical Areas	
Total Area w/in ROW (acres)	0.0000
% ROW	0.00%
Maryland DNR State Wetlands & Wetlands of Special State Concern (WSSC)	
Total Area State Wetlands and 25-ft buffers w/in 1/4-mile ROW buffer (acres)	0.0000
% 1/4-mile ROW buffer	0.00%
Total Area of WSSC and 100-ft buffers w/in 1/4-mile ROW buffer (acres):	0.0000
% 1/4-mile ROW buffer	0.00%
Total Area of State Wetlands, WSSC, and all buffers w/in-1/4 mile ROW buffer (acres):	0.0000
% 1/4-mile ROW buffer	0.00%

Figure 7. Miscellaneous Calculations Table Structure

3.5 Models and Python Scripts

3.5.1 Script #1 – Establish Sensitive Areas and Initial Matting Placement

The field-delineated wetlands and watercourses make up the primary sensitive areas for analysis. Prior to producing any impact calculations, a number of geoprocessing steps were executed to establish the following additional sensitive areas within the ROW:

- 25-ft Field-Delineated Wetland Buffer
- 25-ft Wetland in Critical Area Buffer
- 25-ft Riparian Buffer

In addition, the following areas were also established within a ¼-mile buffer outside of the ROW:

- Clipped Maryland DNR Wetlands
- Clipped 25-ft Maryland DNR Wetland Buffer
- Clipped Maryland WSSC
- Clipped 100-ft Maryland WSSC Buffer

ModelBuilder was first utilized to test the workflow with hard-coded variables before being exported to a Python script that used dynamic user-input variables. The first model was used to generate all sensitive areas, access road matting, and initial pole matting. This model is shown in Figures 8 and 9. It has been broken into four pieces due to its size and paper size restrictions. Once the buffers were established within this model, they were then merged with the wetland and watercourse datasets into an aggregate sensitive areas dataset. This operation presented two advantages 1) the merged dataset would contain a "Type" field that distinguishes each polygon as one of the five sensitive areas, and 2) regarding the creation of access road centerlines with the merged dataset, the resulting output was continuous lines spanning multiple, adjacent sensitive areas instead of lines broken at each sensitive area boundary. The affected access road centerlines were then buffered by 7 feet with a flat end type to create the initial 14-ft wide access road matting. The process for creating the initial pole matting areas is detailed on Figure 13. The steps to build the ¼-mile buffer outside the ROW and the Maryland state wetland datasets and their associated buffers are included in Figure 10. The user interface of the exported Python script tool showing the required user-input parameters is shown in Figure 11.

Close examination of the first model (Figures 8 and 9) shows an existing hierarchy associated with the creation of the buffers, hence the numerous instances of the Erase tool to address overlapping sensitive areas. Wetlands and watercourses will always be present as they take precedence, but their buffers may not always get generated if it is to be generated in the same location as another overlapping sensitive area. The hierarchy is as follows:

1. Wetland/Watercourse
2. 25-ft Wetland in Critical Area Buffer
3. 25-ft Wetland Buffer
4. 25-ft Riparian Buffer

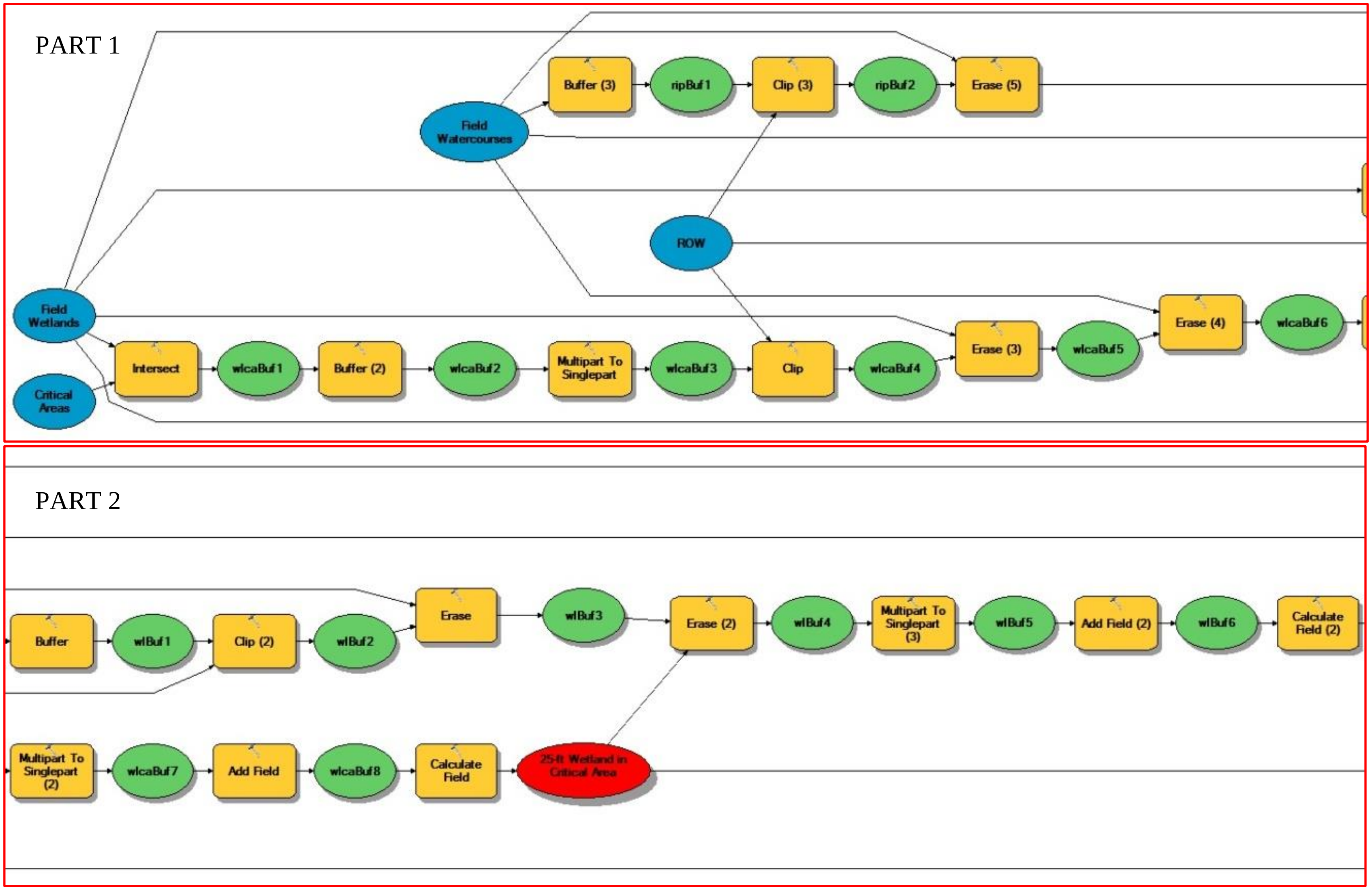


Figure 8. Model for Buffer Construction and Access Road Matting – Parts 1 and 2

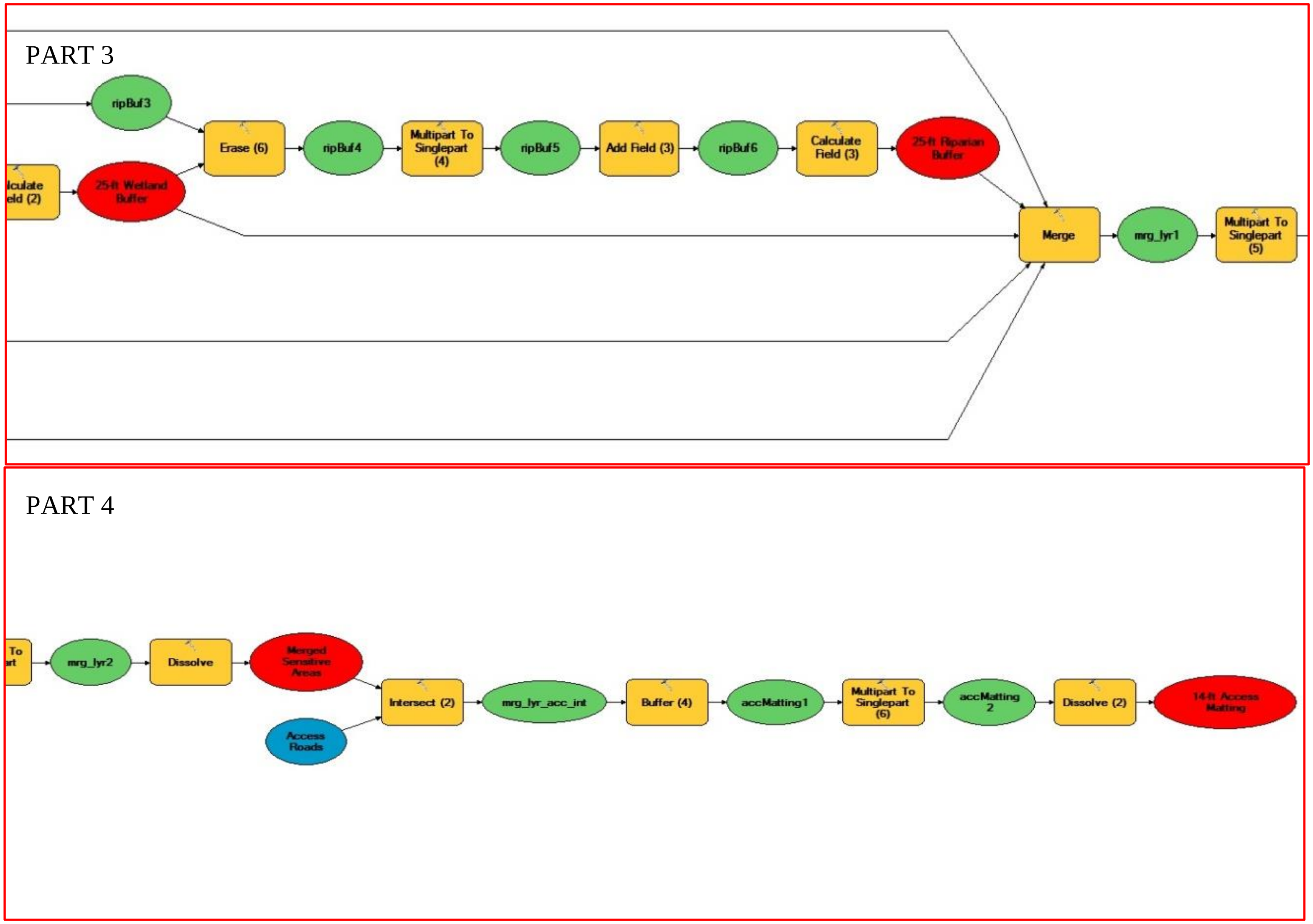


Figure 9. Model for Buffer Construction and Access Road Matting – Parts 3 and 4

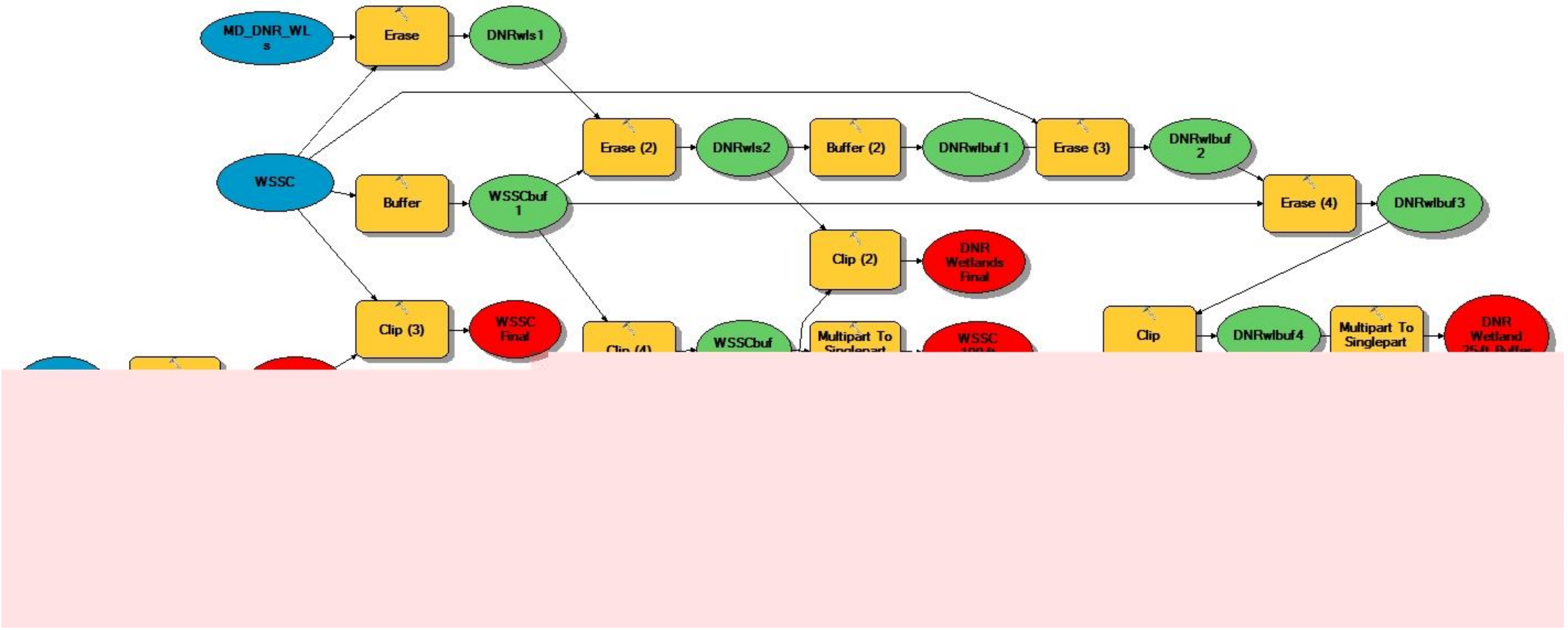


Figure 10. Model for State Buffer Construction

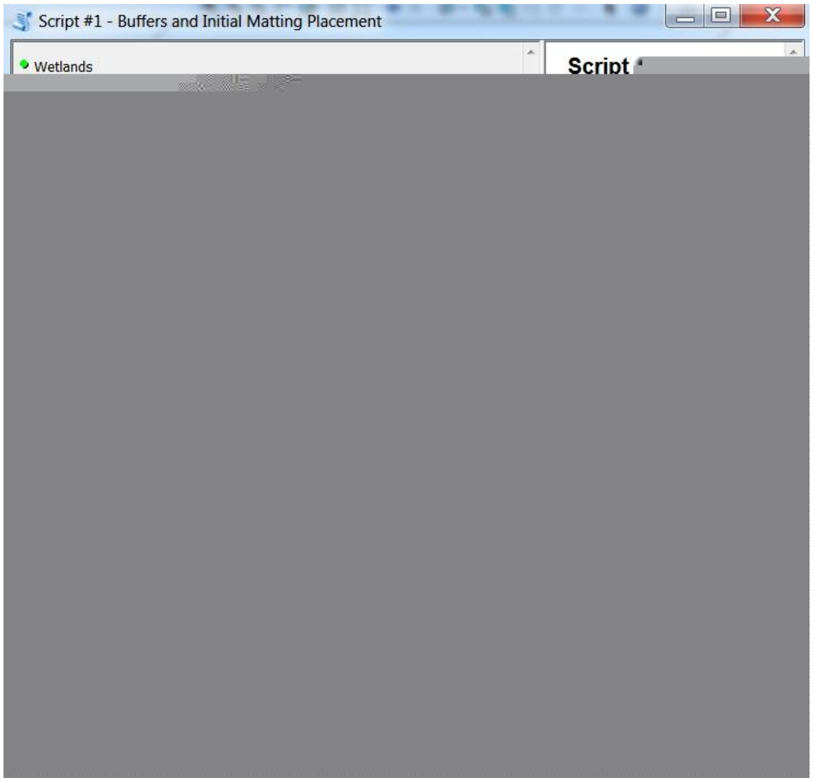


Figure 11. Script Tool User Interface

Figure 12 illustrates an example of the hierarchal concept. In this example, a watercourse is bordered by wetlands on each of its banks. In this case, the watercourse's riparian buffer would be erased with the overlapping wetlands and wetland buffers because those areas take precedence. Therefore, impact calculations would only be generated for the wetland and not the riparian buffer, in order to avoid overlapping areas and double-counting of the impacts.

The process for producing the 48' x 56' work area of initial matting around existing and proposed poles within sensitive areas could not be accomplished using ModelBuilder and instead was achieved using custom coding in a Python script that used arrays, lists, and geometry objects (Esri 2011a). The first step involved using the Select

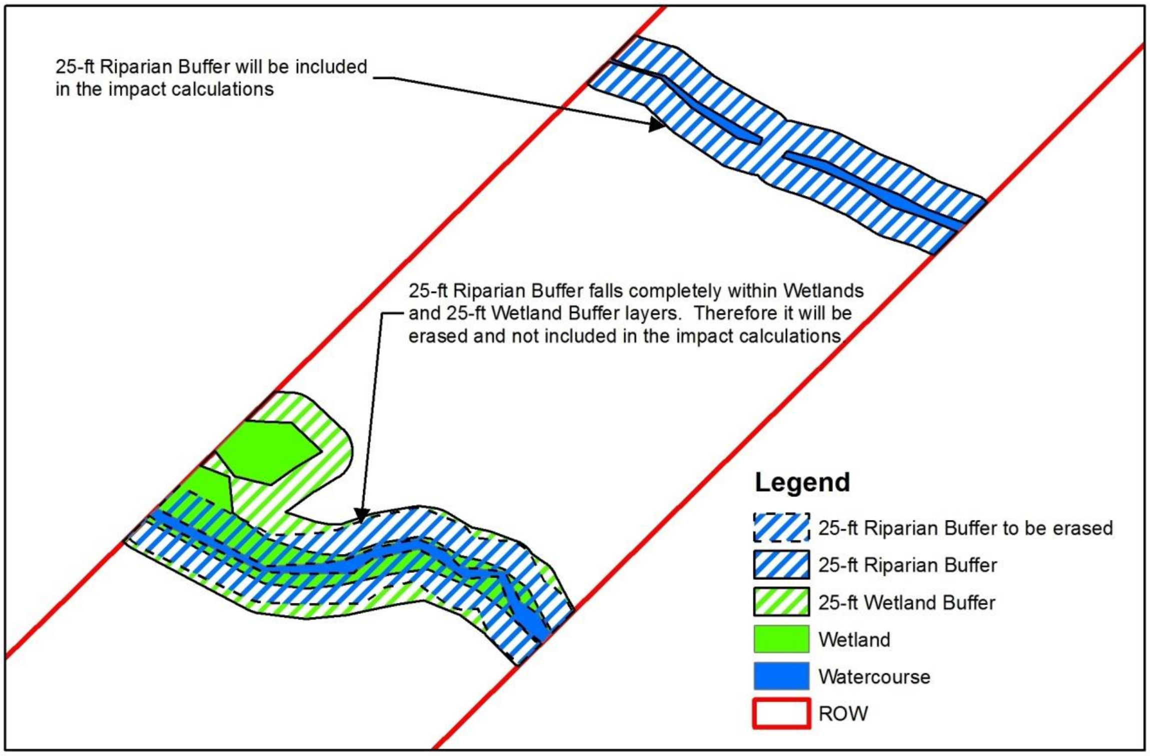


Figure 12. Buffer Hierarchy Example

Layer By Location geoprocessing tool to select existing and proposed poles that were located in sensitive areas. The Arcpy SearchCursor function was then used to look at each affected pole, read its X and Y coordinates, construct four properly-spaced corner points around the pole, and connect the points to create a polygon representing the 48x56 foot matted work area. The resulting pole matting areas were then copied into the access road matting dataset to complete the initial matting placement. Figure 13 displays a layout of the four corner points and resultant matted work area. This marked the completion of the first script.

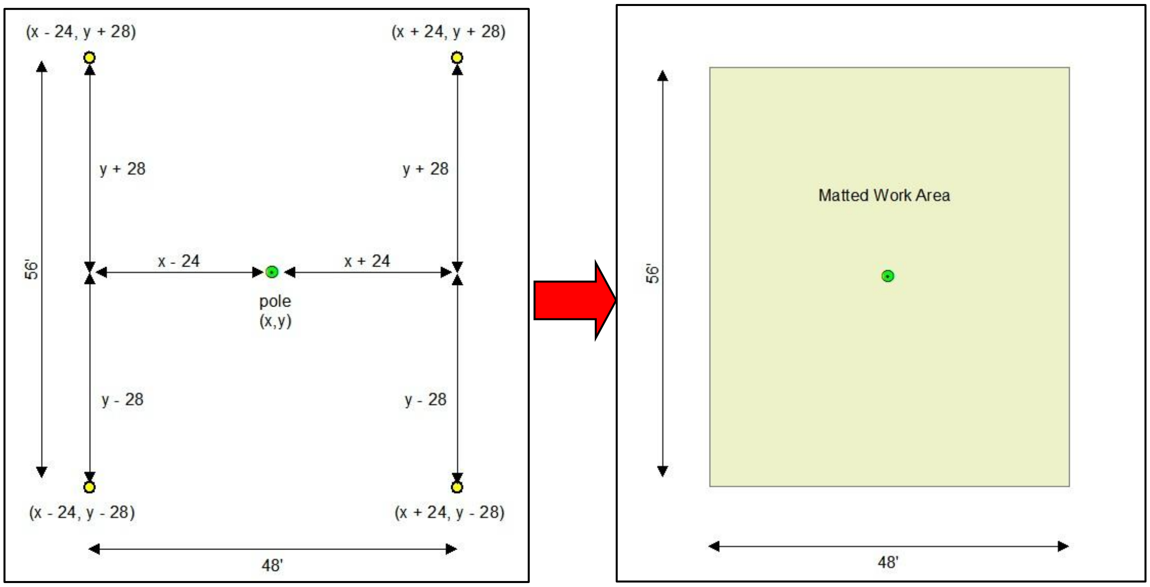


Figure 13. Initial Pole Matting Area Design Layout

3.5.2 Manual Matting Adjustments

The next step involved two manual steps that could not be accomplished with a script:

1. Extending the access road matting polygons towards the upland side of a sensitive area to ensure complete coverage prior to calculating impacts. This was accomplished by selecting and moving the two vertices at either end of the access matting simultaneously in order to preserve the 14-foot matting width (Figures 14 and 15).
2. Properly orienting the pole matting areas to be parallel with the direction of the ROW (Figure 16), as this is typically how the mats are laid out by construction crews in the field. The pole matting area may also be oriented to a “best-fit” location based on surrounding sensitive areas (Figure 17).

Once these two steps were performed, the matting placement was considered final and the impact calculations script was executed.

Figure 14. Extension of Access Road Matting Example - Part 1

Figure 15. Extension of Access Road Matting Example – Part 2

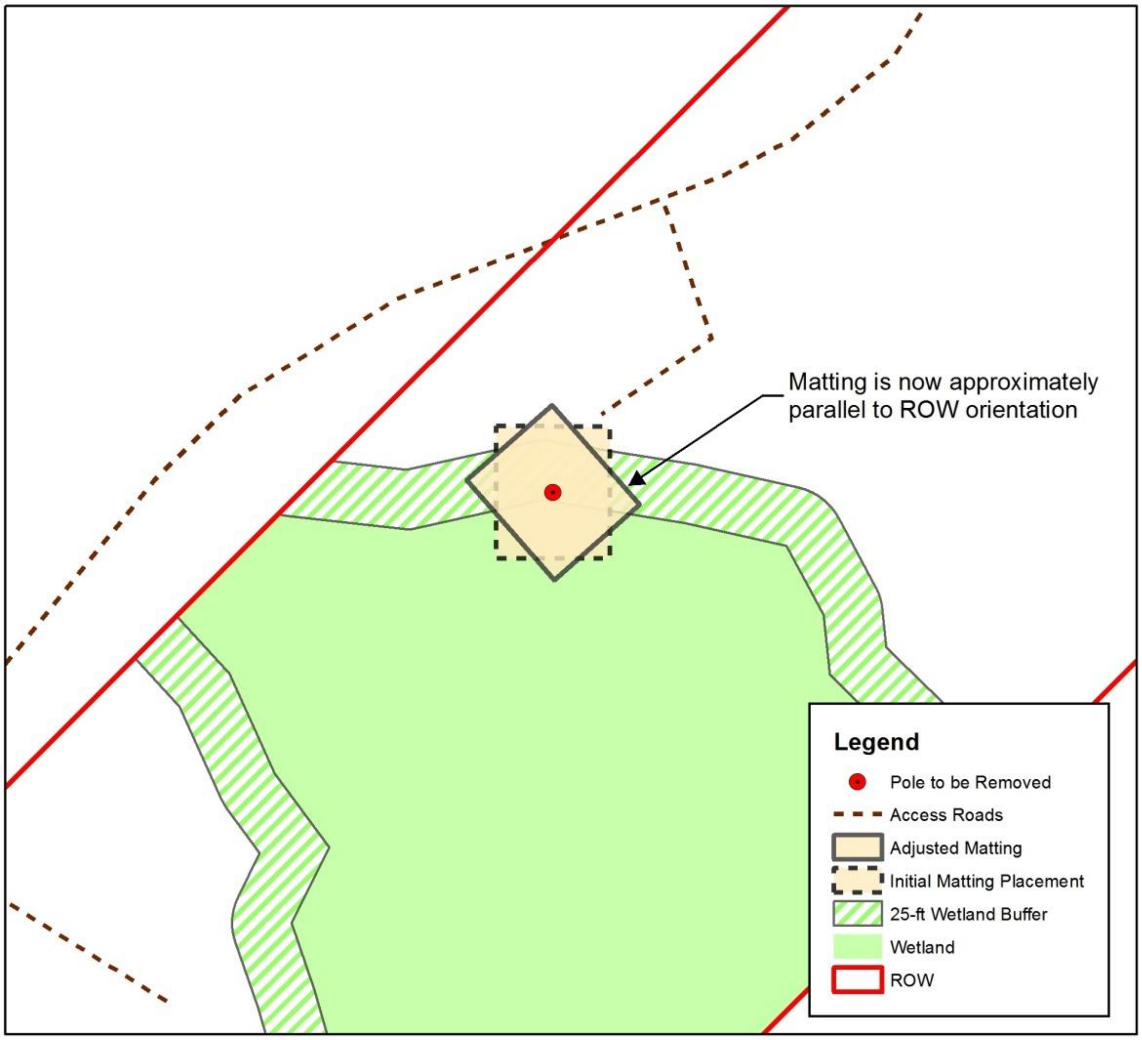


Figure 16. Orienting of Pole Matting Area Example

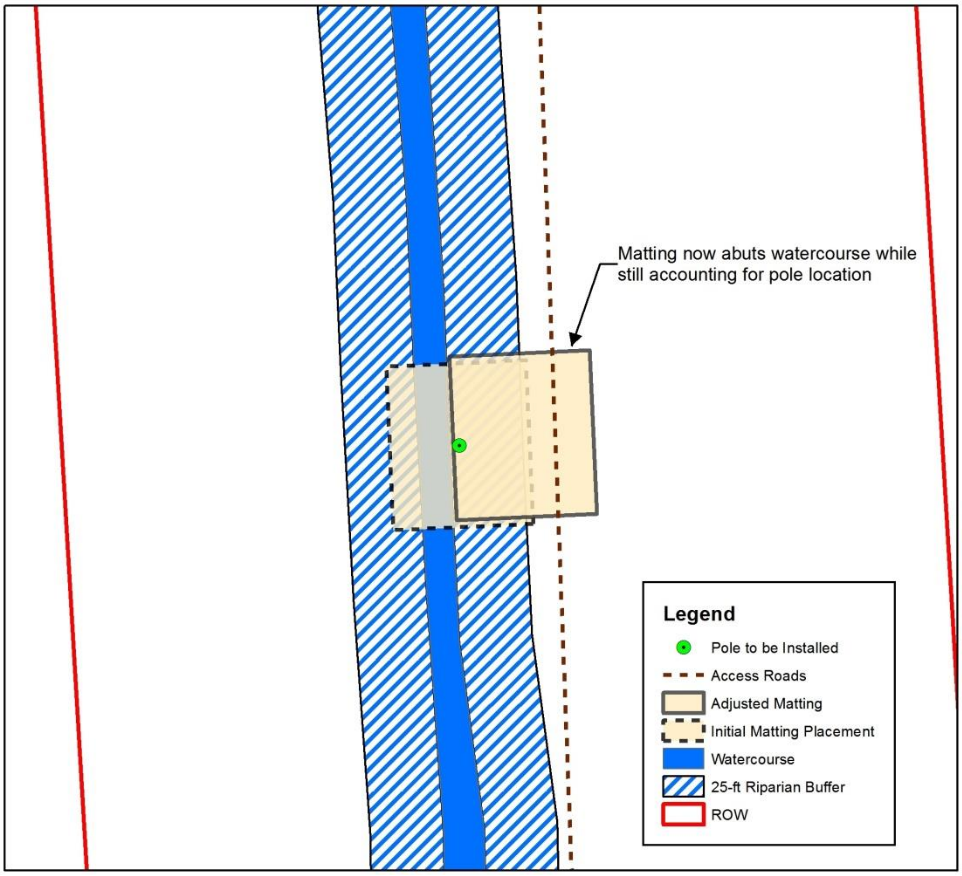
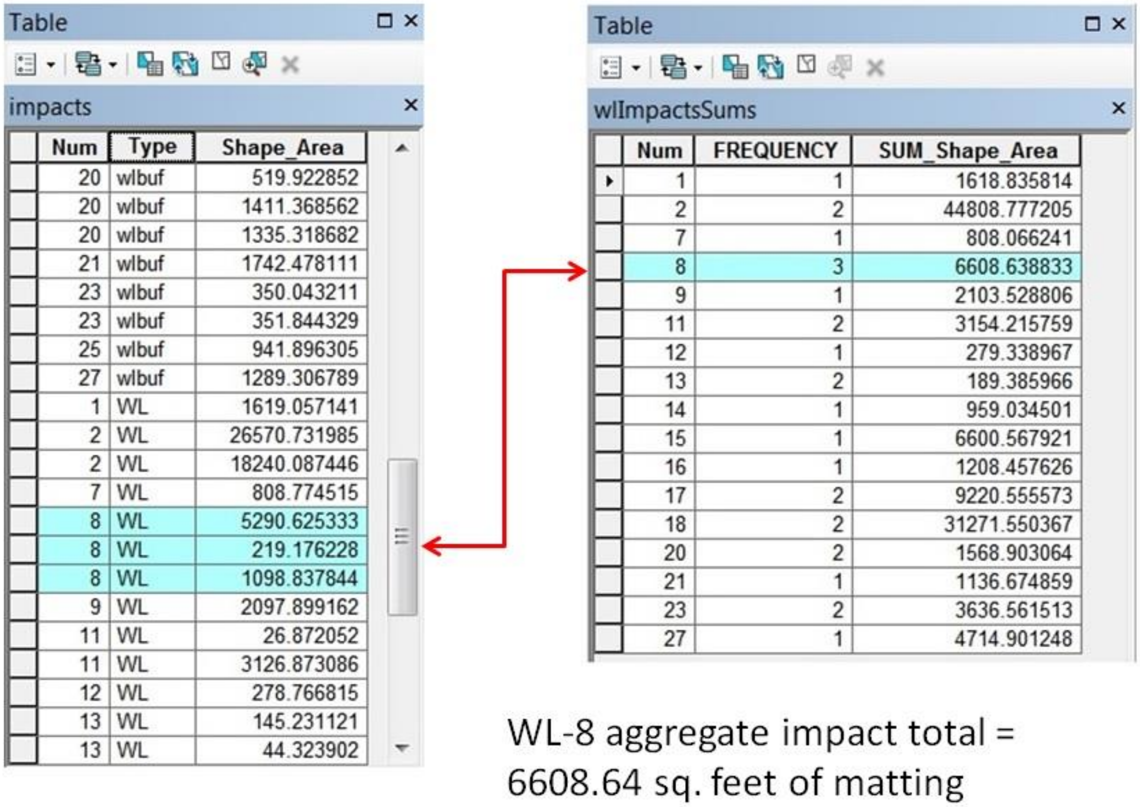


Figure 17. "Best-fit" Location Adjustment of Pole Matting Area Example

3.5.3 Script #2 – Calculate Impacts and Output to Excel Tables

The second script performed all impact calculations and output them to the appropriate table within a user-defined Excel file. First, the pole matting and access road matting datasets that were manually edited were merged into one dissolved dataset to account for overlapping areas. At this point, the matting and sensitive areas datasets were intersected with resulting polygons classified by sensitive area type. It is important to note that the sensitive areas dataset has a field called “Num” that carries the ID number for the corresponding wetland or watercourse. This ID number tells Python which row to output a number in the Excel file, as each row represents a unique wetland or watercourse

ID. When the “Num”, “Type”, and “Shape_Area” fields are analyzed together, aggregate matting area calculations can be made for each of the sensitive area types by wetland ID or watercourse ID and output to summary tables. In Figure 18, the attribute table on the left lists each impacts instance as a result of the intersection along with the three aforementioned fields. The attribute table on the right shows the aggregation of the wetland impacts by ID number.



impacts

Num	Type	Shape_Area
20	wlbuf	519.922852
20	wlbuf	1411.368562
20	wlbuf	1335.318682
21	wlbuf	1742.478111
23	wlbuf	350.043211
23	wlbuf	351.844329
25	wlbuf	941.896305
27	wlbuf	1289.306789
1	WL	1619.057141
2	WL	26570.731985
2	WL	18240.087446
7	WL	808.774515
8	WL	5290.625333
8	WL	219.176228
8	WL	1098.837844
9	WL	2097.899162
11	WL	26.872052
11	WL	3126.873086
12	WL	278.766815
13	WL	145.231121
13	WL	44.323902

wlImpactsSums

Num	FREQUENCY	SUM_Shape_Area
1	1	1618.835814
2	2	44808.777205
7	1	808.066241
8	3	6608.638833
9	1	2103.528806
11	2	3154.215759
12	1	279.338967
13	2	189.385966
14	1	959.034501
15	1	6600.567921
16	1	1208.457626
17	2	9220.555573
18	2	31271.550367
20	2	1568.903064
21	1	1136.674859
23	2	3636.561513
27	1	4714.901248

WL-8 aggregate impact total = 6608.64 sq. feet of matting

Figure 18. Aggregate Impacts by Wetland/Watercourse ID and Sensitive Area Type

In a separate process, poles to be installed located within each of the sensitive areas were split into different datasets using a series of spatial join operations. Figure 19 shows an overview model of the aforementioned processes, all of which lead into the customized Python code operations and Excel table outputs.

The xlwt module write function was utilized to output data and apply formulas to specific cells. The write function syntax is as follows:

Worksheet.write(row #, column #, data to write, style to use)

The function uses a zero-based numbering system. For example, to write the number 16 to cell A1 (first row, first column) using the style2B style, the code would be:

Worksheet.write(0, 0, 16, style2B)

Given the zero-based numbering system and the fact that no numbers would be output until the fourth row of any table as a result of the table title and column headings, careful attention had to be paid to which row numbers were used in output code statements.

Three impact tables had to be populated:

1. Wetlands
2. Combined 25-ft Wetland and Wetland in Critical Area Buffer
3. Combined Watercourse and 25-ft Riparian Buffer

Populating these tables used the exact same process since each table used the same columns as Figure 5. The populating process using the Wetlands table as an example is described below.

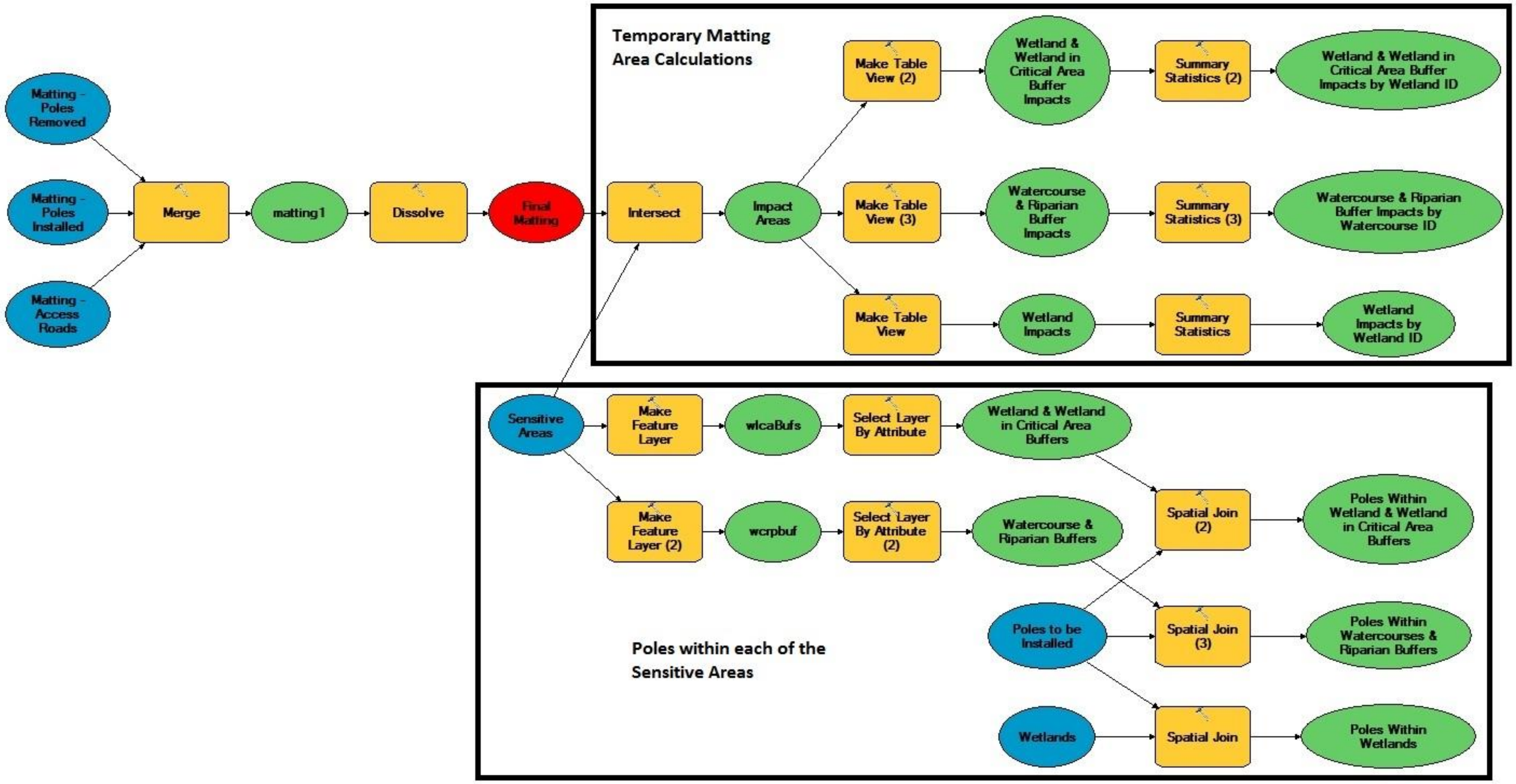


Figure 19. Model for Developing Aggregate Impacts of and Pole Counts within Sensitive Areas

The first step involved populating the Wetland Size columns. The summary table produced from the wetland dataset was used to aggregate wetland area totals and populate these columns. Second, the Temporary Protective Matting columns were populated by using the SearchCursor method to look at each row of the wetland impact area summary table, determine the wetland ID and aggregate area sum, and output accordingly.

The third step involved looking at the pole counts within wetlands in order to calculate permanent impacts. This process involved not only gathering the IDs of the affected wetlands but also determining how many poles of each diameter had to be accounted for. A majority of the poles to be installed along the circuit are to be 3.5 feet in diameter, whereas diameters for poles at turns (angle structures) are 8.5 feet. By using a combination of Python lists, the SearchCursor method, and iterations of for loops and if loops, the correct number of poles was able to be determined and output to the proper column of the Excel table (No. of Direct Embed Poles or No. of Anchor Bolt Foundations). Once these pole counts were in place, the columns for permanent impacts (Impacts due to Utility Lines and Impacts due to Foundations for Utility Lines) were calculated using a simple area formula of:

pole count * πr^2 (r = radius, either 1.75 or 4.25)

Once permanent impacts were calculated, the Total Impact column could be calculated by summing the two permanent impacts columns together. At this point, formulas were used to calculate sums of every column, completing the Wetland Impacts table.

Note: Cells within all output Excel tables were preset to zero, using two decimal places for square feet, four decimal places for acreages (Afanasiev 2013), and whole numbers for pole counts. This helped the script minimize the number of cells to populate as only those IDs that had impacts were populated and all others remained at zero. In addition, any square footage calculation output to a table had a corresponding acreage calculation that was easily determined by dividing the square footage by 43,560.

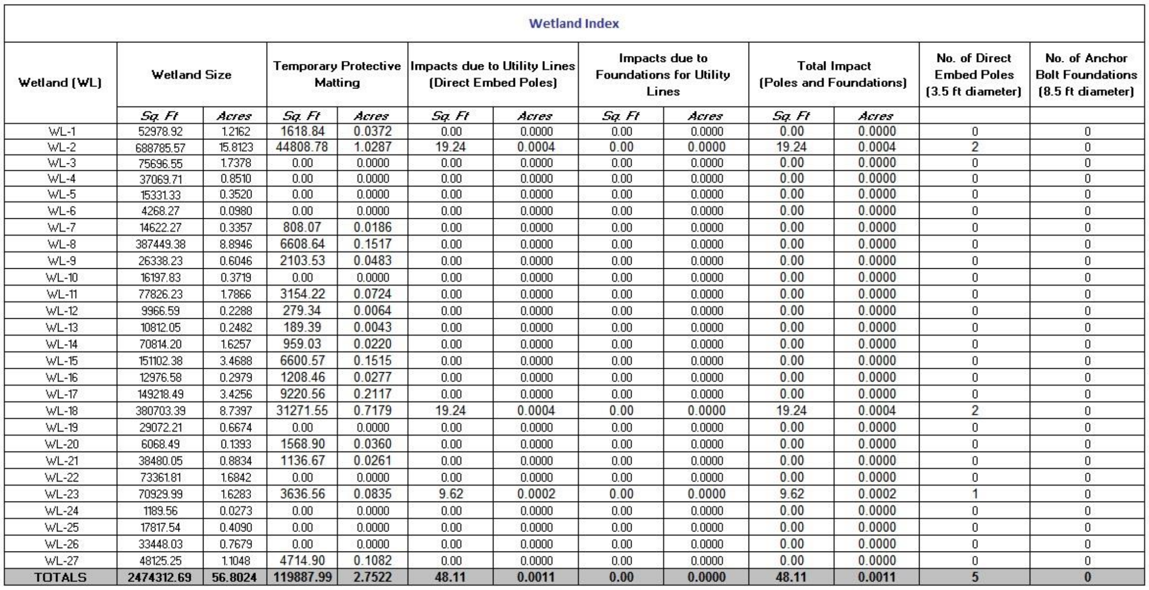
Chapter 4: Results and Discussion

4.1 Final Output Excel Tables

The output table results for the wetlands (Tables 2 and 3), combined 25-ft wetland and wetland in Critical Area buffers (Tables 4 and 5), and combined watercourse and 25-ft riparian buffers (Tables 6 and 7) are included below along with a comparison to each table's respective original permit calculations. Overall, the script produces the desired outputs for all tables but also highlights some errors that were made with the original manual permit calculations that were generated manually. It should be noted that Temporary Protective Matting calculations are not expected to be 100% accurate due to the manual steps of extending matting paths through sensitive areas and rotating matted work areas around structures.

The wetlands are the most significant sensitive area in the permitting process and therefore it is imperative that accurate impact calculations are developed for these areas. The output table (Tables 2 and 3) developed from the script reflects the same totals developed from the original manual calculations.

Table 2. Wetland Impacts Table Output



Wetland Index

Wetland (WL)	Wetland Size		Temporary Protective Matting		Impacts due to Utility Lines (Direct Embed Poles)		Impacts due to Foundations for Utility Lines		Total Impact (Poles and Foundations)		No. of Direct Embed Poles (3.5 ft diameter)	No. of Anchor Bolt Foundations (8.5 ft diameter)
	Sq. Ft	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*		
WL-1	52978.92	1.2162	1618.84	0.0372	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-2	688785.57	15.8123	44808.78	1.0287	19.24	0.0004	0.00	0.0000	19.24	0.0004	2	0
WL-3	75696.55	1.7378	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-4	37069.71	0.8510	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-5	15331.33	0.3520	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-6	4268.27	0.0980	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-7	14622.27	0.3357	808.07	0.0186	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-8	387449.38	8.8946	6608.64	0.1517	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-9	26338.23	0.6046	2103.53	0.0483	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-10	16197.83	0.3719	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-11	77826.23	1.7866	3154.22	0.0724	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-12	9966.59	0.2288	279.34	0.0064	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-13	10812.05	0.2482	189.39	0.0043	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-14	70814.20	1.6257	959.03	0.0220	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-15	151102.38	3.4688	6600.57	0.1515	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-16	12976.58	0.2979	1208.46	0.0277	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-17	149218.49	3.4256	9220.56	0.2117	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-18	380703.39	8.7397	31271.55	0.7179	19.24	0.0004	0.00	0.0000	19.24	0.0004	2	0
WL-19	29072.21	0.6674	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-20	6068.49	0.1393	1568.90	0.0360	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-21	38480.05	0.8834	1136.67	0.0261	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-22	73361.81	1.6842	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-23	70929.99	1.6283	3636.56	0.0835	9.62	0.0002	0.00	0.0000	9.62	0.0002	1	0
WL-24	1189.56	0.0273	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-25	17817.54	0.4090	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-26	33448.03	0.7679	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-27	48125.25	1.1048	4714.90	0.1082	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
TOTALS	**2474312.69**	**56.8024**	**119887.99**	**2.7522**	**48.11**	**0.0011**	**0.00**	**0.0000**	**48.11**	**0.0011**	**5**	**0**

Table 3. Wetland Impacts Analysis

	Wetland Size		Temporary Protective Matting		Impacts due to Utility Lines (Direct Embed Poles)		Impacts due to Foundations for Utility Lines		Total Impact (Poles and Foundations)		No. of Direct Embed Poles (3.5 ft diameter)	No. of Anchor Bolt Foundations (8.5 ft diameter)
	Sq. Ft	**Acres**	**Sq. Ft**	**Acres**	**Sq. Ft**	**Acres**	**Sq. Ft**	**Acres**	**Sq. Ft**	**Acres**		
SCRIPT TOTALS	**2474312.69**	**56.8024**	**119887.99**	**2.7522**	**48.11**	**0.0011**	**0.00**	**0.0000**	**48.11**	**0.0011**	**5**	**0**
PERMIT TOTALS	**2474312.69**	**56.8024**	**119740.34**	**2.7489**	**48.11**	**0.0011**	**0.00**	**0.0000**	**48.11**	**0.0011**	**5**	**0**
ACCURACY	**100%**	**100%**	**99.9%**	**99.9%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

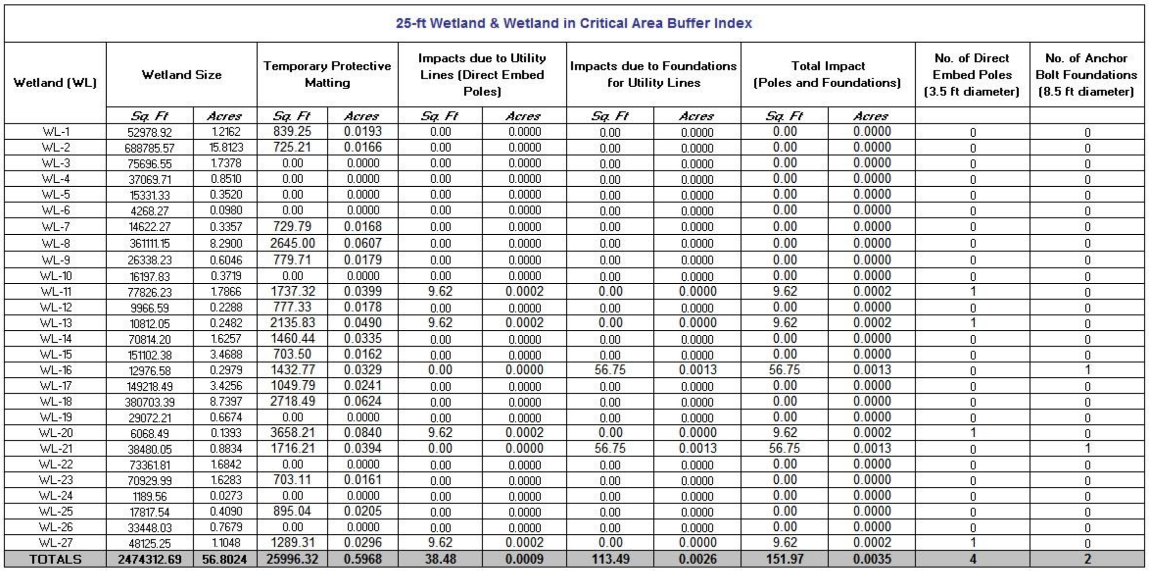
The output tables (Tables 4 and 5) for the combined 25-ft wetland and wetland in Critical Area buffer reflect most of the original manual calculations, but highlight a few minor errors associated with the permanent impact columns. All manual errors are a result of cells not using a formula and instead being input as text. The manual errors include the following:

- WL-13 Impacts due to Foundations for Utility Lines had originally equaled 9.62 s.f. and 0.0002 acres; however, the No. of Anchor Bolt Foundations (8.5 ft. diameter) column equaled zero. This results in the Total Impact columns being higher by 9.62 s.f. and 0.0002 acres, respectively.

- The acreages for WL-16 and WL-21 Impacts due to Foundations for Utility Lines had both originally equaled "0.0009*", with the asterisk footnoting that these 8.5-ft diameter poles were assumed to be angle structures and therefore warranting the larger diameter. However, the 0.0009 number was derived using a 7-ft diameter, which was used early in the project before receiving further clarification from the client on actual pole diameters. The two 0.0009 acreage calculations for WL-16 and WL-21 along with the bottom line total for Total Impact were never updated when the diameters were updated and instead were left as text.

The resulting net effects of these errors, when comparing the manual permit calculations to the script calculations, are +0.0008 acres for Impacts due to Foundations for Utility Lines, and -9.64 s.f. and +0.0006 acres for Total Impacts.

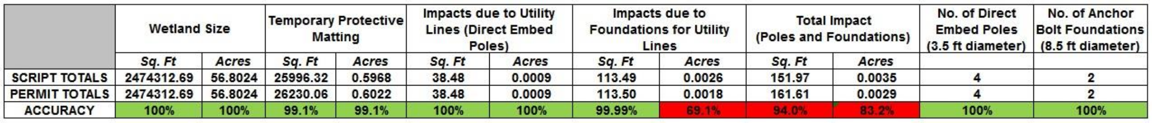
Table 4. Combined 25-ft Wetland and Critical Area Buffers Impact Table Output



25-ft Wetland & Wetland in Critical Area Buffer Index

Wetland (WL)	Wetland Size		Temporary Protective Matting		Impacts due to Utility Lines (Direct Embed Poles)		Impacts due to Foundations for Utility Lines		Total Impact (Poles and Foundations)		No. of Direct Embed Poles (3.5 ft diameter)	No. of Anchor Bolt Foundations (8.5 ft diameter)
	Sq. Ft	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*		
WL-1	52978.92	1.2162	839.25	0.0193	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-2	688785.57	15.8123	725.21	0.0166	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-3	75696.55	1.7378	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-4	37069.71	0.8510	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-5	15331.33	0.3520	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-6	4268.27	0.0980	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-7	14622.27	0.3357	729.79	0.0168	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-8	361111.15	8.2900	2645.00	0.0607	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-9	26338.23	0.6046	779.71	0.0179	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-10	16197.83	0.3719	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-11	77826.23	1.7866	1737.32	0.0399	9.62	0.0002	0.00	0.0000	9.62	0.0002	1	0
WL-12	9966.59	0.2288	777.33	0.0178	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-13	10812.05	0.2482	2135.83	0.0490	9.62	0.0002	0.00	0.0000	9.62	0.0002	1	0
WL-14	70814.20	1.6257	1460.44	0.0335	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-15	151102.38	3.4688	703.50	0.0162	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-16	12976.58	0.2979	1432.77	0.0329	0.00	0.0000	56.75	0.0013	56.75	0.0013	0	1
WL-17	149218.49	3.4256	1049.79	0.0241	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-18	380703.39	8.7397	2718.49	0.0624	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-19	29072.21	0.6674	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-20	6068.49	0.1393	3658.21	0.0840	9.62	0.0002	0.00	0.0000	9.62	0.0002	1	0
WL-21	38480.05	0.8834	1716.21	0.0394	0.00	0.0000	56.75	0.0013	56.75	0.0013	0	1
WL-22	73361.81	1.6842	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-23	70929.99	1.6283	703.11	0.0161	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-24	1189.56	0.0273	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-25	17817.54	0.4090	895.04	0.0205	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-26	33448.03	0.7679	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WL-27	48125.25	1.1048	1289.31	0.0296	9.62	0.0002	0.00	0.0000	9.62	0.0002	1	0
TOTALS	**2474312.69**	**56.8024**	**25996.32**	**0.5968**	**38.48**	**0.0009**	**113.49**	**0.0026**	**151.97**	**0.0035**	**4**	**2**

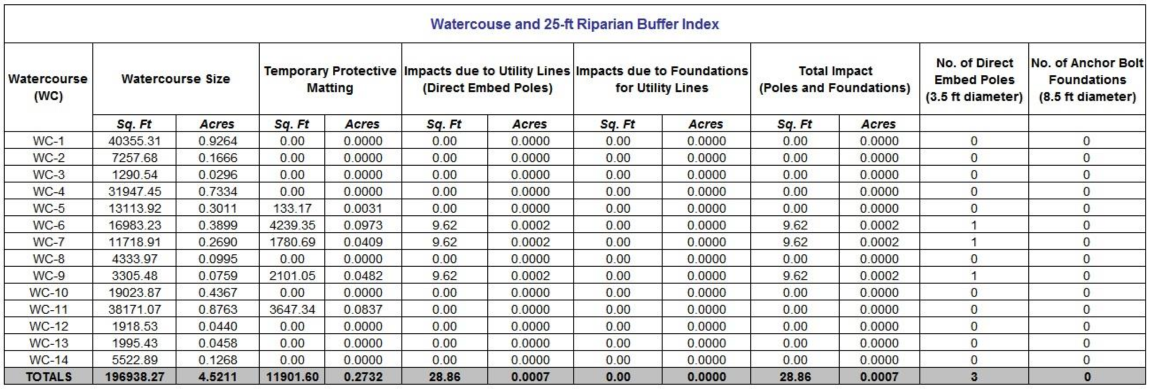
Table 5. Combined 25-ft Wetland and Critical Area Buffers Impact Analysis



	Wetland Size		Temporary Protective Matting		Impacts due to Utility Lines (Direct Embed Poles)		Impacts due to Foundations for Utility Lines		Total Impact (Poles and Foundations)		No. of Direct Embed Poles (3.5 ft diameter)	No. of Anchor Bolt Foundations (8.5 ft diameter)
	Sq. Ft	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*		
SCRIPT TOTALS	**2474312.69**	**56.8024**	**25996.32**	**0.5968**	**38.48**	**0.0009**	**113.49**	**0.0026**	**151.97**	**0.0035**	**4**	**2**
PERMIT TOTALS	**2474312.69**	**56.8024**	**26230.06**	**0.6022**	**38.48**	**0.0009**	**113.50**	**0.0018**	**161.61**	**0.0029**	**4**	**2**
ACCURACY	**100%**	**100%**	**99.1%**	**99.1%**	**100%**	**100%**	**99.99%**	**69.1%**	**94.0%**	**83.2%**	**100%**	**100%**

The output tables (Tables 6 and 7) for the combined watercourse and 25-ft riparian buffer do not reflect the original manual calculations; however, the root cause for the errors can be traced to WC-5. The Temporary Protective Matting impact calculation for WC-5 had originally been 1654.62 s.f. and 0.0380 acres, but most of these impacts had already been attributed to WL-13 and its 25-ft buffer area, which take precedence in the sensitive areas hierarchy. By only including the watercourse area being impacted, the Temporary Protective Matting totals for WC-5 lower to 133.17 s.f. and 0.0031 acres. A second error was associated with the pole to be installed within the 25-ft riparian buffer of WC-5. However, this pole is already counted in the WL-13 buffer total, which takes precedence.

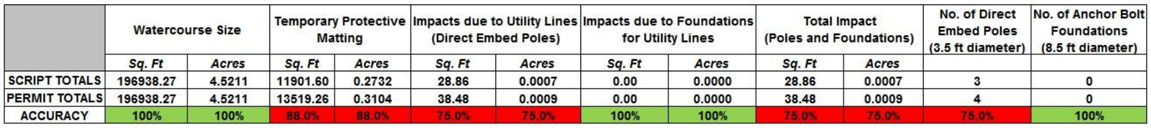
Table 6. Combined Watercourse and 25-ft Riparian Buffer Impacts Table Output



Watercouse and 25-ft Riparian Buffer Index

Watercourse (WC)	Watercourse Size		Temporary Protective Matting		Impacts due to Utility Lines (Direct Embed Poles)		Impacts due to Foundations for Utility Lines		Total Impact (Poles and Foundations)		No. of Direct Embed Poles (3.5 ft diameter)	No. of Anchor Bolt Foundations (8.5 ft diameter)
	Sq. Ft	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*		
WC-1	40355.31	0.9264	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-2	7257.68	0.1666	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-3	1290.54	0.0296	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-4	31947.45	0.7334	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-5	13113.92	0.3011	133.17	0.0031	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-6	16983.23	0.3899	4239.35	0.0973	9.62	0.0002	0.00	0.0000	9.62	0.0002	1	0
WC-7	11718.91	0.2690	1780.69	0.0409	9.62	0.0002	0.00	0.0000	9.62	0.0002	1	0
WC-8	4333.97	0.0995	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-9	3305.48	0.0759	2101.05	0.0482	9.62	0.0002	0.00	0.0000	9.62	0.0002	1	0
WC-10	19023.87	0.4367	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-11	38171.07	0.8763	3647.34	0.0837	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-12	1918.53	0.0440	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-13	1995.43	0.0458	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
WC-14	5522.89	0.1268	0.00	0.0000	0.00	0.0000	0.00	0.0000	0.00	0.0000	0	0
TOTALS	**196938.27**	**4.5211**	**11901.60**	**0.2732**	**28.86**	**0.0007**	**0.00**	**0.0000**	**28.86**	**0.0007**	**3**	**0**

Table 7. Combined Watercourse and 25-ft Riparian Buffer Impacts Analysis



	Watercourse Size		Temporary Protective Matting		Impacts due to Utility Lines (Direct Embed Poles)		Impacts due to Foundations for Utility Lines		Total Impact (Poles and Foundations)		No. of Direct Embed Poles (3.5 ft diameter)	No. of Anchor Bolt Foundations (8.5 ft diameter)
	Sq. Ft	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*	*Sq. Ft*	*Acres*		
SCRIPT TOTALS	**196938.27**	**4.5211**	**11901.60**	**0.2732**	**28.86**	**0.0007**	**0.00**	**0.0000**	**28.86**	**0.0007**	**3**	**0**
PERMIT TOTALS	**196938.27**	**4.5211**	**13519.26**	**0.3104**	**38.48**	**0.0009**	**0.00**	**0.0000**	**38.48**	**0.0009**	**4**	**0**
ACCURACY	**100%**	**100%**	**88.0%**	**88.0%**	**75.0%**	**75.0%**	**100%**	**100%**	**75.0%**	**75.0%**	**75.0%**	**100%**

The resulting net effects of these errors, when comparing the manual permit calculations to the script calculations, are -1617.66 s.f. and -0.0372 acres for Temporary Protective Matting, -9.62 s.f. and -0.0002 acres for Impacts due to Utility Lines and Total Impact, and -1 for No. of Direct Embed Poles (3.5-ft diameter). The area of WC-5 containing the errors is detailed in Figure 20.

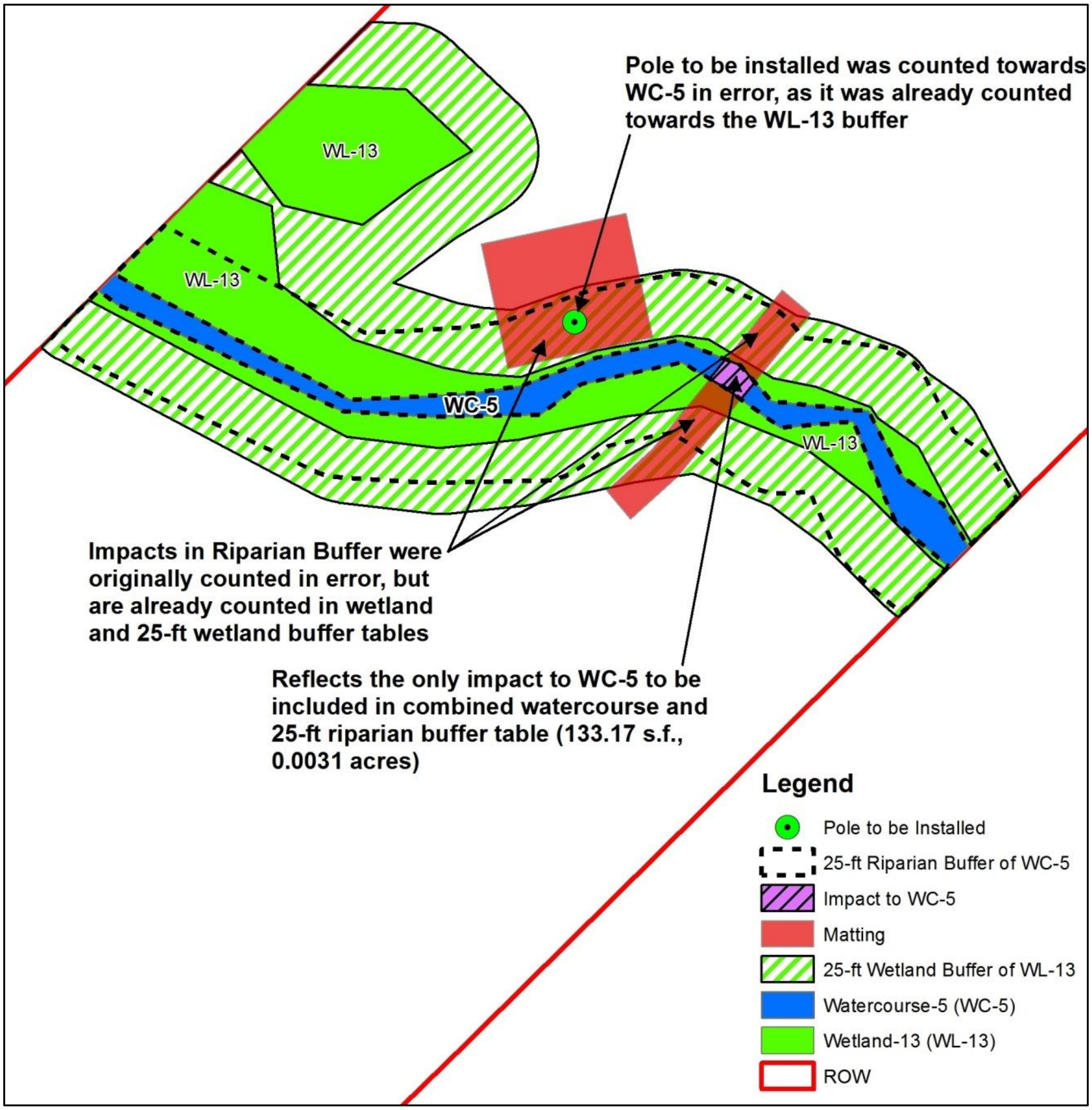
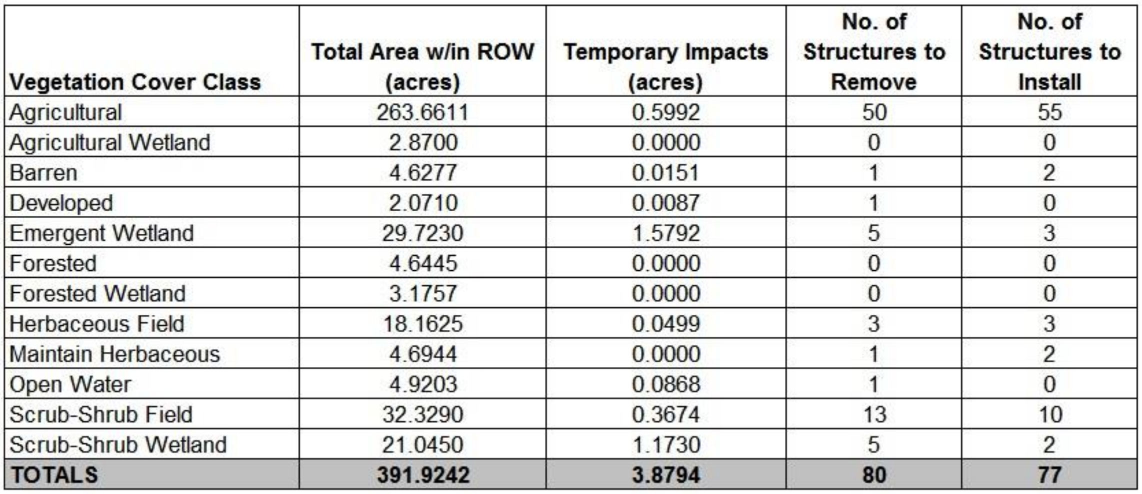


Figure 20. WC-5 Manual Impact Errors

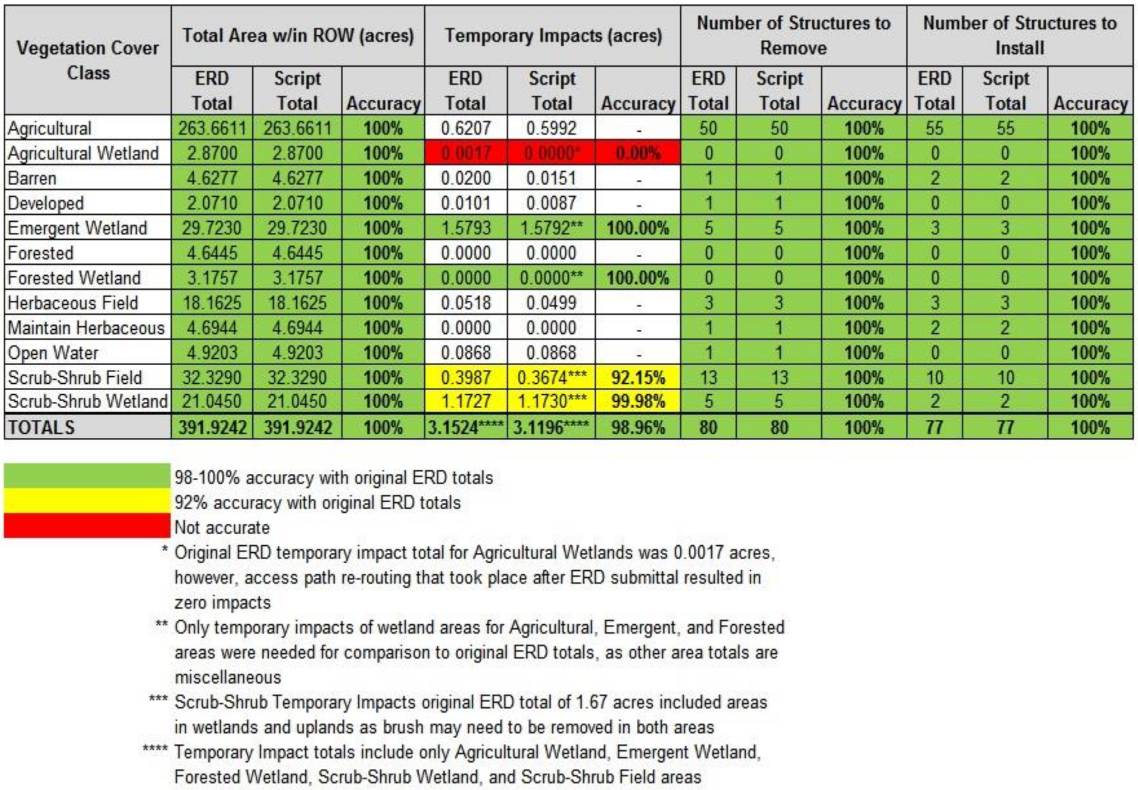
The output tables (Tables 8 and 9) for vegetation cover closely reflect the totals included in the ERD text. The total area of each vegetation cover type within the ROW and the number of poles to install and remove within each vegetation cover type are 99% accurate when compared to the original manual calculations. For temporary impacts, the only vegetation cover totals needed for comparison to the original ERD totals are those in wetlands and Scrub-Shrub Fields. As previously stated, the wetlands reflect the sensitive areas of most significance. Temporary impacts for the Scrub-Shrub Field class are grouped with the Scrub-Shrub Wetlands class to highlight all areas where brush may need to be removed for access, which helps aid PHI in their vegetation management approach during construction.

Table 8. Vegetation Cover Impacts Table Output



Vegetation Cover Class	Total Area w/in ROW (acres)	Temporary Impacts (acres)	No. of Structures to Remove	No. of Structures to Install
Agricultural	263.6611	0.5992	50	55
Agricultural Wetland	2.8700	0.0000	0	0
Barren	4.6277	0.0151	1	2
Developed	2.0710	0.0087	1	0
Emergent Wetland	29.7230	1.5792	5	3
Forested	4.6445	0.0000	0	0
Forested Wetland	3.1757	0.0000	0	0
Herbaceous Field	18.1625	0.0499	3	3
Maintain Herbaceous	4.6944	0.0000	1	2
Open Water	4.9203	0.0868	1	0
Scrub-Shrub Field	32.3290	0.3674	13	10
Scrub-Shrub Wetland	21.0450	1.1730	5	2
TOTALS	**391.9242**	**3.8794**	**80**	**77**

Table 9. Vegetation Cover Impacts Analysis



Vegetation Cover Class	Total Area w/in ROW (acres)			Temporary Impacts (acres)			Number of Structures to Remove			Number of Structures to Install		
	ERD Total	Script Total	Accuracy	ERD Total	Script Total	Accuracy	ERD Total	Script Total	Accuracy	ERD Total	Script Total	Accuracy
Agricultural	263.6611	263.6611	**100%**	0.6207	0.5992	-	50	50	**100%**	55	55	**100%**
Agricultural Wetland	2.8700	2.8700	**100%**	0.0017	0.0000*	**0.00%**	0	0	**100%**	0	0	**100%**
Barren	4.6277	4.6277	**100%**	0.0200	0.0151	-	1	1	**100%**	2	2	**100%**
Developed	2.0710	2.0710	**100%**	0.0101	0.0087	-	1	1	**100%**	0	0	**100%**
Emergent Wetland	29.7230	29.7230	**100%**	1.5793	1.5792**	**100.00%**	5	5	**100%**	3	3	**100%**
Forested	4.6445	4.6445	**100%**	0.0000	0.0000	-	0	0	**100%**	0	0	**100%**
Forested Wetland	3.1757	3.1757	**100%**	0.0000	0.0000**	**100.00%**	0	0	**100%**	0	0	**100%**
Herbaceous Field	18.1625	18.1625	**100%**	0.0518	0.0499	-	3	3	**100%**	3	3	**100%**
Maintain Herbaceous	4.6944	4.6944	**100%**	0.0000	0.0000	-	1	1	**100%**	2	2	**100%**
Open Water	4.9203	4.9203	**100%**	0.0868	0.0868	-	1	1	**100%**	0	0	**100%**
Scrub-Shrub Field	32.3290	32.3290	**100%**	0.3987	0.3674***	**92.15%**	13	13	**100%**	10	10	**100%**
Scrub-Shrub Wetland	21.0450	21.0450	**100%**	1.1727	1.1730***	**99.98%**	5	5	**100%**	2	2	**100%**
TOTALS	**391.9242**	**391.9242**	**100%**	**3.1524******	**3.1196******	**98.96%**	**80**	**80**	**100%**	**77**	**77**	**100%**

🟩	98-100% accuracy with original ERD totals
🟨	92% accuracy with original ERD totals
🟥	Not accurate

* Original ERD temporary impact total for Agricultural Wetlands was 0.0017 acres, however, access path re-routing that took place after ERD submittal resulted in zero impacts

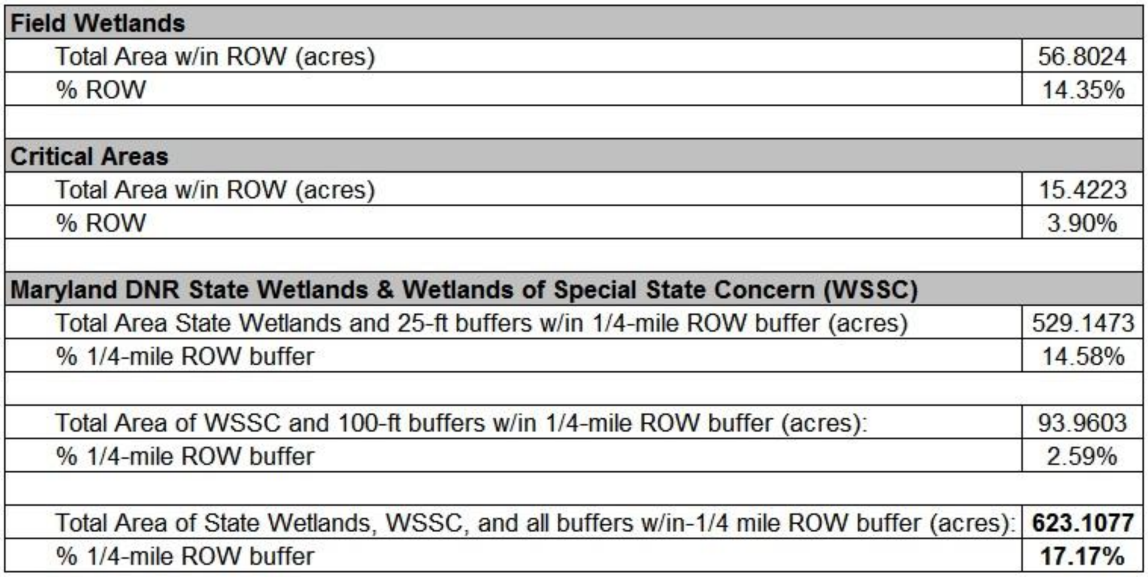
** Only temporary impacts of wetland areas for Agricultural, Emergent, and Forested areas were needed for comparison to original ERD totals, as other area totals are miscellaneous

*** Scrub-Shrub Temporary Impacts original ERD total of 1.67 acres included areas in wetlands and uplands as brush may need to be removed in both areas

**** Temporary Impact totals include only Agricultural Wetland, Emergent Wetland, Forested Wetland, Scrub-Shrub Wetland, and Scrub-Shrub Field areas

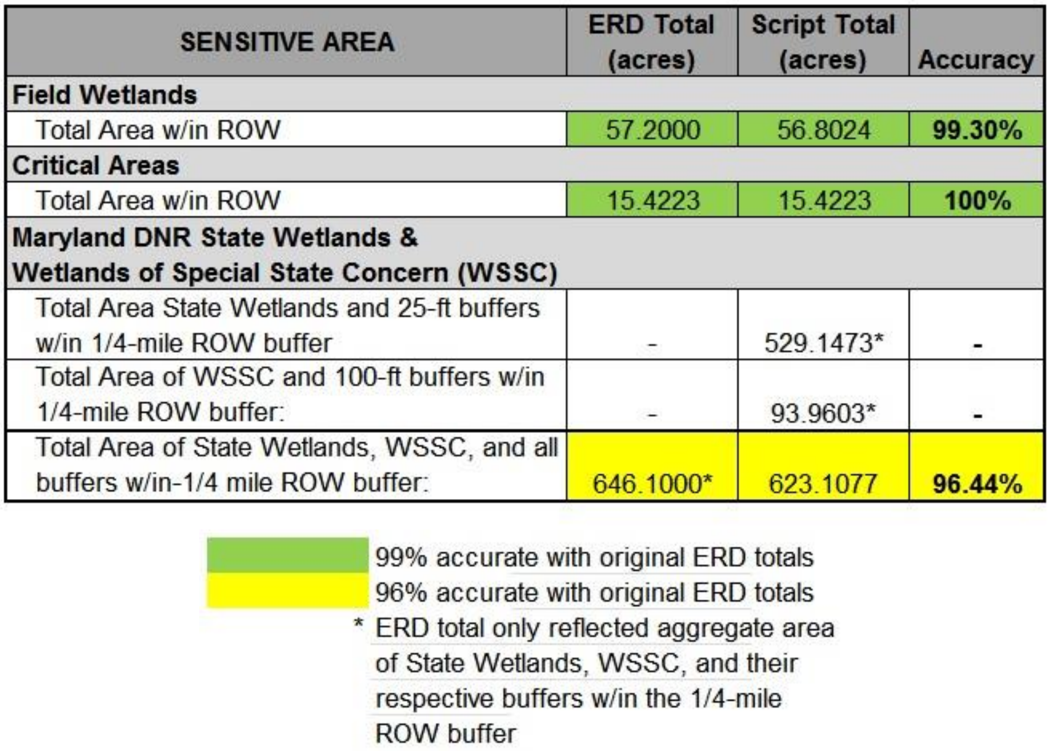
The output tables (Tables 10 and 11) containing miscellaneous calculations mostly reflect the original manual calculations with the exception of the aggregate area total of MD State wetlands and associated 25-ft buffers, and MD WSSC and associated

Table 10. Miscellaneous Impacts Table Output



Field Wetlands	
Total Area w/in ROW (acres)	56.8024
% ROW	14.35%
Critical Areas	
Total Area w/in ROW (acres)	15.4223
% ROW	3.90%
Maryland DNR State Wetlands & Wetlands of Special State Concern (WSSC)	
Total Area State Wetlands and 25-ft buffers w/in 1/4-mile ROW buffer (acres)	529.1473
% 1/4-mile ROW buffer	14.58%
Total Area of WSSC and 100-ft buffers w/in 1/4-mile ROW buffer (acres):	93.9603
% 1/4-mile ROW buffer	2.59%
Total Area of State Wetlands, WSSC, and all buffers w/in-1/4 mile ROW buffer (acres):	**623.1077**
% 1/4-mile ROW buffer	**17.17%**

Table 11. Miscellaneous Impacts Table Analysis



SENSITIVE AREA	ERD Total (acres)	Script Total (acres)	Accuracy
Field Wetlands			
Total Area w/in ROW	57.2000	56.8024	**99.30%**
Critical Areas			
Total Area w/in ROW	15.4223	15.4223	**100%**
Maryland DNR State Wetlands & Wetlands of Special State Concern (WSSC)			
Total Area State Wetlands and 25-ft buffers w/in 1/4-mile ROW buffer	-	529.1473*	**-**
Total Area of WSSC and 100-ft buffers w/in 1/4-mile ROW buffer:	-	93.9603*	**-**
Total Area of State Wetlands, WSSC, and all buffers w/in-1/4 mile ROW buffer:	646.1000*	623.1077	**96.44%**

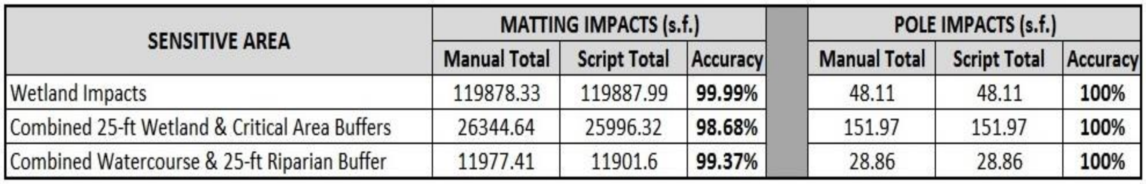
🟩 99% accurate with original ERD totals
🟨 96% accurate with original ERD totals
* ERD total only reflected aggregate area of State Wetlands, WSSC, and their respective buffers w/in the 1/4-mile ROW buffer

buffers. Given that the calculation is still 96% accurate, this indicates there may have been a small error in the manual geoprocessing tasks originally undertaken to produce the calculations.

4.2 Process Verification – Manual vs. Automated Calculations

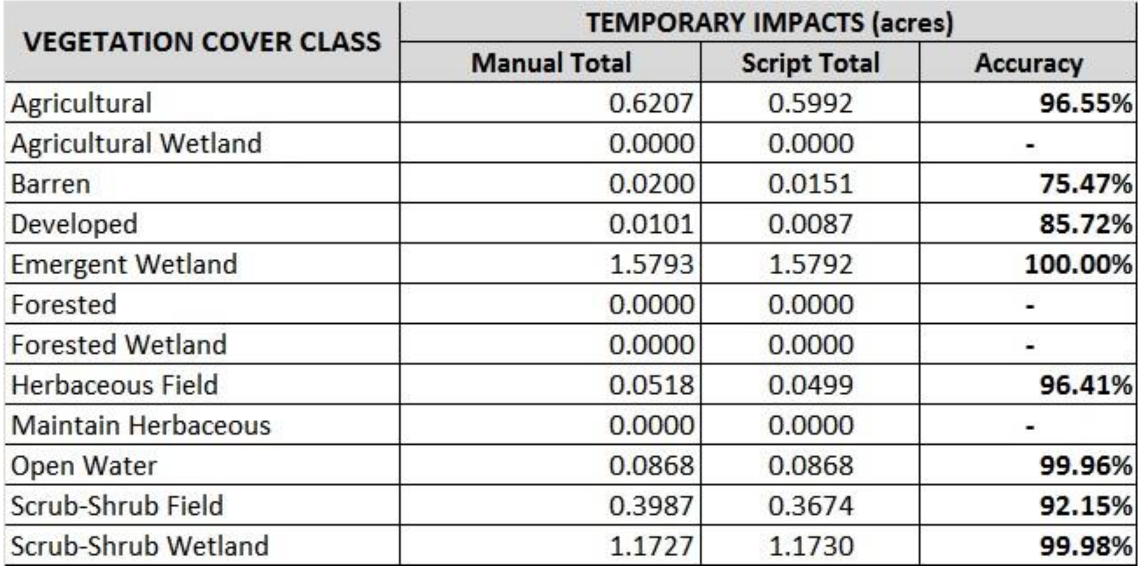
In order to verify the results of the automated calculations, a manual process was executed using the same step-by-step approach established for the two scripts. The manual calculations were then used to verify the automated calculations and determine whether the scripts were constructed correctly. Impact calculations of sensitive areas were generated for both matting and poles, which was the breakdown shown in the output Excel tables for wetlands (Table 2), combined 25-ft wetland and Critical Area buffers (Table 4), and combined watercourse and 25-ft riparian buffer (Table 6). An acceptable tolerance for the manual totals of sensitive areas was set at +/- 2% of the automated script totals. The only expected deviations in the totals would be derived from how far access matting was manually extended into uplands from wetland buffers and how pole matting work areas were manually rotated after placement. The resulting totals for matting and pole impacts to wetlands, combined 25-ft wetland and Critical Area buffer areas, and combined watercourse and 25-ft riparian buffer areas (Table 12) all fell within +/- 2% of the script totals. The resulting totals for vegetation impacts (Table 13) were acceptable, as deviations in totals were derived from how access matting was manually extended into upland areas from wetland buffers. The resulting totals for miscellaneous calculations (Table 14) matched exactly with those generated in the scripts. The manual verification process took approximately 2 ½ hours to complete, compared to approximately 10 minutes with the automated process (including manual matting adjustments). This certainly highlighted the importance of an automated solution.

Table 12. Manual Verification Results of Sensitive Area Impacts



SENSITIVE AREA	MATTING IMPACTS (s.f.)				POLE IMPACTS (s.f.)		
	Manual Total	Script Total	Accuracy		Manual Total	Script Total	Accuracy
Wetland Impacts	119878.33	119887.99	**99.99%**		48.11	48.11	**100%**
Combined 25-ft Wetland & Critical Area Buffers	26344.64	25996.32	**98.68%**		151.97	151.97	**100%**
Combined Watercourse & 25-ft Riparian Buffer	11977.41	11901.6	**99.37%**		28.86	28.86	**100%**

Table 13. Manual Verification Results of Vegetation Impacts



VEGETATION COVER CLASS	TEMPORARY IMPACTS (acres)		
	Manual Total	Script Total	Accuracy
Agricultural	0.6207	0.5992	**96.55%**
Agricultural Wetland	0.0000	0.0000	**-**
Barren	0.0200	0.0151	**75.47%**
Developed	0.0101	0.0087	**85.72%**
Emergent Wetland	1.5793	1.5792	**100.00%**
Forested	0.0000	0.0000	**-**
Forested Wetland	0.0000	0.0000	**-**
Herbaceous Field	0.0518	0.0499	**96.41%**
Maintain Herbaceous	0.0000	0.0000	**-**
Open Water	0.0868	0.0868	**99.96%**
Scrub-Shrub Field	0.3987	0.3674	**92.15%**
Scrub-Shrub Wetland	1.1727	1.1730	**99.98%**

Table 14. Manual Verification Results of Miscellaneous Calculations

SENSITIVE AREA	Script Total (acres)	Manual Total (acres)	Accuracy
Field Wetlands			
Total Area w/in ROW	56.8024	56.8024	**100%**
Critical Areas			
Total Area w/in ROW	15.4223	15.4223	**100%**
Maryland DNR State Wetlands & Wetlands of Special State Concern (WSSC)			
Total Area State Wetlands and 25-ft buffers w/in 1/4-mile ROW buffer	529.1473	529.1473	**100%**
Total Area of WSSC and 100-ft buffers w/in 1/4-mile ROW buffer:	93.9603	93.9603	**100%**
Total Area of State Wetlands, WSSC, and all buffers w/in-1/4 mile ROW buffer:	623.1077	623.1077	**100%**

4.3 Process Validation

The impact calculations used in this research serve the purposes of both the regulatory permitting phase and the construction phase. For permitting, the impacts help demonstrate to state and federal regulatory agencies how much total acreage of sensitive areas will be impacted throughout construction (Small 2013). For construction, the total impacts in square feet are used as the basis to solicit bids from matting contractors for the entire transmission line project. The contractors also take into consideration an analysis of construction plan maps, the construction schedule, and access routes to schedule the installation and removal of matting throughout the project. Since all mats will not be on the ground at the same time, the contractor is able to use and re-use the matting throughout the project to achieve cost efficiencies for PHI (Savage 2013). For the transmission line rebuild project used for this research, the total amount of matting used as the basis for soliciting bids from matting contractors was 228,306.57 square feet (McBurney 2014). Of this total, only 210,114.22 square feet fall within Maryland, which is the study area for this research, while the remaining falls within Delaware. Also deducted from this total is 49,794.78 square feet of additional matting used for two large transmission conductor pulling locations that traverse wetlands and wetland buffers (Black & Veatch 2013), leaving the final matting total at 160,319.44 square feet. Pulling locations were left out of the scope of this research as they are added very late in the construction preparation process and require custom sizes and very specific ROW orientation. The resulting matting total of 160,319.44 square feet is within +/- 10% (1.6%) of the total impacts calculation generated in this research (157,785.91 s.f.). By meeting the needs of the regulatory permitting and construction stakeholders by providing accurate calculations generated through the use of scripts, this research's solution has been validated.

Chapter 5: Conclusions and Future Research

5.1 Conclusions

The objective of this study was to construct, verify, and validate a set of Python scripts that take user-provided input parameters, process all necessary environmental impact calculations, and output the results as tables within multiple worksheets of an Excel file. By leveraging the automation capabilities of Python, all calculations and necessary re-calculations of environmental impacts are performed quickly and accurately simply by providing the necessary inputs. This would provide the utility companies with peace of mind that the calculations are accurate, dependable, and flexible in the event that any construction plan changes are made, and ultimately would help minimize construction expenses and environmental impacts.

The resulting impact calculations were exactly what was desired at the start of this research; however, the means to get to that point took a slightly different path. ModelBuilder and custom Python code were used extensively to produce all of the sensitive areas, place matting, generate impact calculations, and place the calculations into an Excel table. The repetitive task of testing the Python code showcased the time-saving component of an automated solution, as providing new input parameters for re-calculations was a quick and easy process. The ability of having a line of communication between ArcGIS, Python, and Excel was vital to this research and an area I believe deserves further exploitation in the GIS industry. The resulting impact calculations were on par with what was desired and helped show that a script solution can provide great benefit in achieving the correct results in a timely fashion in order to meet the needs of both regulatory permitting and construction stakeholders.

Ultimately, the scripts generated in this research did not provide a fully automated solution as desired. The manual matting adjustments that had to be made in between execution of the two scripts highlighted the dynamic nature of a transmission line in general and all of the varying environmental and planimetric scenarios that would need to be accounted for to achieve a fully automated solution. Future research is needed to deeper examine these scenarios and to develop a plan for addressing them accordingly within the realm of a coding environment.

The takeaway from this research is that the ArcGIS-Python-Excel line of communication can be utilized for many different applications of spatial analysis. Most people in the GIS industry use ArcGIS and Excel on a regular, if not daily basis. It is the Python component that may not be the most clear to the average GIS user. Python presents a wealth of tools that can be leveraged to do very specific tasks and provide results in a flexible manner. The ArcPy package introduced at ArcGIS version 10 continues to expand with each version as more tools are being provided to accomplish a multitude of various spatial analysis tasks. While my research did not achieve complete automation, there are no doubt many other applications that could be automated using a similar ArcGIS-Python-Excel setup as used in this research.

5.2 Future Research

There are three items that need to be investigated more in-depth from a coding perspective to determine if it is possible to convert this research from a semi-automated solution to a completely automated solution:

1) The extension of access road centerlines to the upland side of a wetland buffer

2) The orientation of the pole matting work area to be parallel with the ROW
3) The addition of large transmission conductor pulling locations that may require matting

During the execution of the first script to create buffers and initial matting, the access road centerlines were clipped to the outer extent of the merged sensitive areas dataset. Buffering the resulting centerlines to create the access road matting did not provide full matting coverage of the sensitive areas, and therefore the resulting matting had to be manually extended to achieve full coverage. To try and avoid this manual step, the Extend Line editing tool was investigated to determine if it could be used within the script. However, the tool will only extend (or snap) a line to the nearest intersecting feature, without any control over which feature it is extended to (Esri 2012). Extending to the nearest feature was not the desired outcome so the decision was made to make it a manual step.

A conceptual solution for this task would be to look at the start (S) and end vertices (E) of each access road centerline and the first vertex after (S+1) and prior (E-1) to the start and end vertices, respectively. By looking at the XY coordinates of S and S+1, likewise E and E-1, the angle of the line drawn between the two vertices could be determined and used to place a vertex a certain distance both before the start vertex and after the end vertex while maintaining the same angle. The buffer to represent access road matting can then be generated and would provide complete coverage of wetland buffers.

The ability to perform simple editing of vertices (e.g. Add Vertex, Remove Vertex) in ArcMap is currently only available through the Edit Sketch Properties window, as these specific tools are not available as geoprocessing tools. The most likely solution to achieve automation of the access road centerline task would be to use the Insert and Update cursors in Python to convert the access road centerlines into Point objects consisting of an array of XY coordinates representing the line's vertices. Once the angles for adding new vertices to the start and end of each line are determined, the new vertices can be written to the geometry of each Point object representing each access road centerline, and new lines can be generated and used to create the buffers representing the access road matting (Esri 2013).

The pole matting work areas are to be oriented to where they are parallel with the ROW. The typical access road to a pole is also parallel with the ROW, therefore orienting pole matting work area this way will help ensure a perpendicular meeting of access road matting with pole matting. When the script generated the pole matting work areas, they were centered on the pole with no rotation, therefore necessitating the manual rotation adjustment. A conceptual solution would involve, for each affected pole, looking at the two closest consecutive vertices of the ROW line to the work area, determining the angle between the two vertices, and applying the angle to the rotation of the matted work area. Research would be necessary for finding out how to arrive at the two closest consecutive ROW line vertices, most likely involving a proximity analysis. There would be some obstacles to overcome with this solution, including the following:

- Poles located at turns where the ROW bends and numerous vertices exist along the ROW line

- Areas where parts of the matting gets placed inside watercourses and would need to be manually moved out (e.g. “Best-fit” scenario – Figure 17)
- Areas where placing the matting parallel to the ROW would not be appropriate due to various access restrictions

The addition of large (e.g., 60x360’) transmission conductor pulling locations at turns will need to be considered earlier in the process for future research. These areas can be generated with a script and centered on a pole/structure much like the pole matting work areas; however, they would need to be manually rotated to a specific orientation, typically in line with one of the circuits at a turn. There is also the possibility that a pulling location polygon would need to be truncated or cut in the middle to avoid coverage of a road, watercourse, forested area, etc. Accounting for these kinds of pulling location scenarios could lead to the script ultimately being a semi-automated solution, as was the case with this research.

Consideration should also be given to future users’ level of familiarity with Python code. Some users may not have programming experience and therefore may not be able to modify the Python code to improve or maintain the tool. To compensate for this, a ModelBuilder-only approach could be used to handle the numerous geoprocessing steps found in the first script, as input parameters could be set within the model. A Python script file used to output the impacts to Excel tables could then be inserted into the model and executed at the appropriate time. However, the script file would eventually need some level of customization for the specific circuit being studied.

Therefore, providing detailed documentation manuals for future users of the script file would be of benefit and help prevent any inexperienced user pitfalls.

Appendix A: Script #1 “buffersMatting1” Python Code

```
# -*- coding: utf-8 -*-
# ---------------------------------------------------------------------------
# buffersMatting1.py
# Created on: 2013-03-09 12:58:40.00000
#   (generated by ArcGIS/ModelBuilder)
# Description:
# ---------------------------------------------------------------------------

# Set the necessary product code
# import arcinfo

# Import modules
import arcpy
from arcpy import env

arcpy.env.overwriteOutput = 'True'

# Input Parameters
wls = arcpy.GetParameterAsText(0)
wcs = arcpy.GetParameterAsText(1)
cas = arcpy.GetParameterAsText(2)
mdWLS = arcpy.GetParameterAsText(3)
mdWSSC = arcpy.GetParameterAsText(4)
pa = arcpy.GetParameterAsText(5)
accRds = arcpy.GetParameterAsText(6)
polesInstall = arcpy.GetParameterAsText(7)
polesRemove = arcpy.GetParameterAsText(8)
outputFolder = arcpy.GetParameterAsText(9)
scratchFolder = arcpy.GetParameterAsText(10)

#Scratch variables
mdWLS_Erase = scratchFolder + "\\mdWLS_Erase"
mdWSSC_Buffer1 = scratchFolder + "\\mdWSSC_Buffer1"
mdWLS_Erase_Erase = scratchFolder + "\\mdWLS_Erase_Erase"
mdWLS_Erase_Erase1 = scratchFolder + "\\mdWLS_Erase_Erase1"
mdWLS_Erase_Erase2 = scratchFolder + "\\mdWLS_Erase_Erase2"
mdWLS_Erase_Erase3 = scratchFolder + "\\mdWLS_Erase_Erase3"
mdWLS_Buffer1 = scratchFolder + "\\mdWLS_Buffer1"
mdWSSC_Buffer2 = scratchFolder + "\\mdWSSC_Buffer2"
wl_25ft_buf1 = scratchFolder + "\\wl_25ft_buf1"
wl_25ft_buf1_Clip = scratchFolder + "\\wl_25ft_buf1_Clip"
wl_25ft_buf1_wc_erase = scratchFolder + "\\wl_25ft_buf1_wc_erase"
wlca_buffer_25ft = scratchFolder + "\\wlca_buf_25ft"
```

```
wl_25ft_buf1_wc_erase_Erase = scratchFolder + "\\wl_25ft_buf1_wc_erase_Erase"
wl_ca_int = scratchFolder + "\wl_ca_int"
wl_ca_int_25ft_buf = scratchFolder + "\wl_ca_int_25ft_buf"
wl_ca_int_25ft_buf_Multipart = scratchFolder + "\\wl_ca_int_25ft_buf_Multipart"
wl_ca_int_25ft_buf_Multipart1 = scratchFolder + "\\wl_ca_int_25ft_buf_Multipart1"
wl_ca_int_25ft_buf_Multipart2 = scratchFolder + "\\wl_ca_int_25ft_buf_Multipart2"
wl_ca_int_25ft_buf_Multipart3 = scratchFolder + "\\wl_ca_int_25ft_buf_Multipart3"
wc_25ft_buf1 = scratchFolder + "\\wc_25ft_buf1"
wc_25ft_buf1_Clip = scratchFolder + "\\wc_25ft_buf1_Clip"
wc_25ft_buf1_Clip_Erase = scratchFolder + "\\wc_25ft_buf1_Clip_Erase"
wl_buffer_25ft = scratchFolder + "\\wl_buf_25ft"
wc_25ft_buf1_Clip_Erase_Eras = scratchFolder + "\\wc_25ft_buf1_Clip_Erase_Eras"
mrg_lyr1 = scratchFolder + "\\mrg_lyr1"
acc_matting_14ft_1 = scratchFolder + "\\acc_matting_14ft_1"
acc_matting_14ft_2 = scratchFolder + "\\acc_matting_14ft_2"

#Output variables
pa_buffer = outputFolder + "\\pa_Buffer"
mdWLS_clip = outputFolder + "\\mdWLS_clip"
mdWLS_buffer = outputFolder + "\\mdWLS_buffer"
mdWSSC_clip = outputFolder + "\\mdWSSC_clip"
mdWSSC_buffer = outputFolder + "\\mdWSSC_buffer"
wl_ca_25ft_buf = outputFolder + "\\wl_ca_25ft_buf"
wl_25ft_buf = outputFolder + "\\wl_25ft_buf"
wc_25ft_buf = outputFolder + "\\wc_25ft_buf"
mrg_lyr = outputFolder + "\\mrg_lyr"
mrg_lyr_Dissolve = outputFolder + "\\mrg_lyr_Dissolve"
mrg_lyr_acc_int = outputFolder + "\\mrg_lyr_acc_int"
acc_matting_14ft = outputFolder + "\\acc_matting_14ft"
acc_matting_14ft_byArea = outputFolder + "\\acc_matting_14ft_byArea"


# MARYLAND DNR 25-FT WETLAND BUFFER & MARYLAND 100-FT
          WETLAND OF SPECIAL STATE
# CONCERN (WSSC) BUFFER (both outside Project Area Right-of Way)

# Erase MD DNR Wetlands with MD WSSC
arcpy.Erase_analysis(mdWLS, mdWSSC, mdWLS_Erase, "")

# Buffer MD WSSC by 100 ft
arcpy.Buffer_analysis(mdWSSC, mdWSSC_Buffer1, "100 Feet", "OUTSIDE_ONLY",\
                  "ROUND", "ALL", "")

# Erase MD DNR Wetlands with WSSC buffer
arcpy.Erase_analysis(mdWLS_Erase, mdWSSC_Buffer1, mdWLS_Erase_Erase, "")
```

```
# Buffer Project Area by 1/4 mile to provide clip extent of MD DNR Wetlands and
# MD WSSC
arcpy.Buffer_analysis(pa, pa_buffer, "0.25 Miles", "OUTSIDE_ONLY", "ROUND",\
                "NONE", "")

# Clip MD DNR Wetlands with 1/4 mile project area buffer
arcpy.Clip_analysis(mdWLS_Erase_Erase, pa_buffer, mdWLS_clip, "")

# Clip MD WSSC with 1/4 mile project area buffer
arcpy.Clip_analysis(mdWSSC, pa_buffer, mdWSSC_clip, "")

# Create 25-ft MD DNR Wetland buffer
arcpy.Buffer_analysis(mdWLS_Erase_Erase, mdWLS_Erase_Erase1, "25 Feet",\
                "OUTSIDE_ONLY", "ROUND", "ALL", "")

# Erase MD DNR Wetland buffer with MD WSSC
arcpy.Erase_analysis(mdWLS_Erase_Erase1, mdWSSC, mdWLS_Erase_Erase2, "")

# Erase MD DNR Wetland buffer with MD WSSC buffer
arcpy.Erase_analysis(mdWLS_Erase_Erase2, mdWSSC_Buffer1,
         mdWLS_Erase_Erase3, "")

# Clip MD DNR Wetland buffer with 1/4 mile project area buffer
arcpy.Clip_analysis(mdWLS_Erase_Erase3, pa_buffer, mdWLS_Buffer1, "")

# Explode MD DNR Wetland buffer to create single part features
arcpy.MultipartToSinglepart_management(mdWLS_Buffer1, mdWLS_buffer)

# Clip MD WSSC buffer to 1/4 mile project area buffer
arcpy.Clip_analysis(mdWSSC_Buffer1, pa_buffer, mdWSSC_Buffer2, "")

# Explode MD WSSC buffer to create single part features
arcpy.MultipartToSinglepart_management(mdWSSC_Buffer2, mdWSSC_buffer)


# WETLAND IN CRITICAL AREA (WLCA) BUFFER
# Intersect Critical Areas (CAs) and surveyed wetlands (wetlands in CAs)
arcpy.Intersect_analysis([cas, wls], wl_ca_int, "ALL", "", "INPUT")

# Buffer wetlands in CAs by 25 ft (WLCA buffer)
arcpy.Buffer_analysis(wl_ca_int, wl_ca_int_25ft_buf, "25 Feet", "OUTSIDE_ONLY",\
                "ROUND", "LIST", "Num")

# Explode WLCA buffer to form single part features
arcpy.MultipartToSinglepart_management(wl_ca_int_25ft_buf,\
                          wl_ca_int_25ft_buf_Multipart)
```

```
# Clip WLCA buffer with Project Area
arcpy.Clip_analysis(wl_ca_int_25ft_buf_Multipart, pa,\
                wl_ca_int_25ft_buf_Multipart1, "")

# Erase WLCA buffer with surveyed wetlands
arcpy.Erase_analysis(wl_ca_int_25ft_buf_Multipart1, wls,\
                wl_ca_int_25ft_buf_Multipart2, "")

# Erase WLCA buffer with surveyed watercourses
arcpy.Erase_analysis(wl_ca_int_25ft_buf_Multipart2, wcs,\
                wl_ca_int_25ft_buf_Multipart3, "")

# Explode WLCA buffer to form single part features
arcpy.MultipartToSinglepart_management(wl_ca_int_25ft_buf_Multipart3,\
                            wl_ca_25ft_buf)

# Add Type field to WLCA buffer
arcpy.AddField_management(wl_ca_25ft_buf, "Type", "TEXT", "", "", "5", "",\
                    "NULLABLE", "NON_REQUIRED", "")

# Calculate Type field in WLCA buffer to "cabuf"
arcpy.CalculateField_management(wl_ca_25ft_buf, "Type", "\"cabuf\"", "VB", "")

# WETLAND BUFFER
# Buffer surveyed wetlands by 25 ft (wetland buffer)
arcpy.Buffer_analysis(wls, wl_25ft_buf1, "25 Feet", "OUTSIDE_ONLY", "ROUND", +\
                "LIST", "Num")

# Clip wetland buffer with Project Area
arcpy.Clip_analysis(wl_25ft_buf1, pa, wl_25ft_buf1_Clip, "")

# Erase wetland buffer with surveyed watercourses
arcpy.Erase_analysis(wl_25ft_buf1_Clip, wcs, wl_25ft_buf1_wc_erase, "")

# Erase wetland buffer with WLCA buffer
arcpy.Erase_analysis(wl_25ft_buf1_wc_erase, wl_ca_25ft_buf,\
                wl_25ft_buf1_wc_erase_Erase, "")

# Explode wetland buffer to form single part features
arcpy.MultipartToSinglepart_management(wl_25ft_buf1_wc_erase_Erase, wl_25ft_buf)

# Add Type field to wetland buffer
arcpy.AddField_management(wl_25ft_buf, "Type", "TEXT", "", "", "5", "",\
                    "NULLABLE", "NON_REQUIRED", "")
```

```
# Calculate Type field in wetland buffer to "wlbuf"
arcpy.CalculateField_management(wl_25ft_buf, "Type", "\"wlbuf\"", "VB", "")

# RIPARIAN BUFFER
# Buffer surveyed watercourses by 25 ft (riparian buffer)
arcpy.Buffer_analysis(wcs, wc_25ft_buf1, "25 Feet", "OUTSIDE_ONLY", "ROUND",\
                "LIST", "Num")

# Clip riparian buffer with Project Area
arcpy.Clip_analysis(wc_25ft_buf1, pa, wc_25ft_buf1_Clip, "")

# Erase riparian buffer with surveyed wetlands
arcpy.Erase_analysis(wc_25ft_buf1_Clip, wls, wc_25ft_buf1_Clip_Erase, "")

# Erase riparian buffer with wetland buffer
arcpy.Erase_analysis(wc_25ft_buf1_Clip_Erase, wl_25ft_buf,\
               wc_25ft_buf1_Clip_Erase_Eras, "")

# Explode riparian buffer to form single part features
arcpy.MultipartToSinglepart_management(wc_25ft_buf1_Clip_Erase_Eras,\
                            wc_25ft_buf)

# Add Type field to riparian buffer
arcpy.AddField_management(wc_25ft_buf, "Type", "TEXT", "", "", "5", "",\
                  "NULLABLE", "NON_REQUIRED", "")

# Calculate Type field in riparian buffer to "rpbuf"
arcpy.CalculateField_management(wc_25ft_buf, "Type", "\"rpbuf\"", "VB", "")


# Merge surveyed wetlands, surveyed watercourses, wetland buffer, riparian
# buffer, and WLCA buffer into single dataset to be used for impact calculations
# (sensitive areas)
arcpy.Merge_management([wc_25ft_buf,wl_25ft_buf,wl_ca_25ft_buf,wcs,wls], \
                mrg_lyr1,"Num \"Num\" true true false 2 Short 0 0 ,First,+\
                #,wcs,Num,-1,-1,wls,Num,-1,-1,wc_25ft_buf,Num,-1,-1,+\
                wl_25ft_buf,Num,-1,-1,wl_ca_25ft_buf,Num,-1,-1;+\
                Shape_length \"Shape_length\" true true false 0 Double +\
                0 0 ,First,#,wcs,Shape_Length,-1,-1,wls,Shape_Length,-1, +\
                -1,wc_25ft_buf,Shape_length,-1,-1,wc_25ft_buf,+\
                Shape_length,-1,-1,wl_25ft_buf,Shape_length,-1,-1, +\
                wl_25ft_buf,Shape_length,-1,-1,wl_ca_25ft_buf, +\
                Shape_length,-1,-1,wl_ca_25ft_buf,Shape_length,-1,-1;+\
                Shape_area \"Shape_area\" true true false 0 Double 0 0 ,+\
                First,#,wcs,Shape_Area,-1,-1,wls,Shape_Area,-1,-1,+\
                wc_25ft_buf,Shape_area,-1,-1,wc_25ft_buf,Shape_area,-1,+\
```

```
          -1,wl_25ft_buf,Shape_area,-1,-1,wl_25ft_buf,Shape_area,+\
          -1,-1,wl_ca_25ft_buf,Shape_area,-1,-1,wl_ca_25ft_buf,+\
          Shape_area,-1,-1;Type \"Type\" true true false 5 Text 0 +\
          0 ,First,#,wcs,Type,-1,-1,wls,Type,-1,-1,wc_25ft_buf,+\
          Type,-1,-1,wl_25ft_buf,Type,-1,-1,wl_ca_25ft_buf,Type,+\
          -1,-1;ID \"ID\" true true false 10 Text 0 0 ,First,#,+\
          wcs,ID,-1,-1,wls,ID,-1,-1;ET_ID \"ET_ID\" true true +\
          false 100 Text 0 0 ,First,#,wcs,ET_ID,-1,-1,wls,ET_ID,+\
          -1,-1;Pre \"Pre\" true true false 20 Text 0 0 ,First,#,+\
          wcs,Pre,-1,-1,wls,Pre,-1,-1;Area_Acres \"Area_Acres\" +\
          true true false 8 Double 0 0 ,First,#,wcs,Area_Acres,-1,+\
          -1,wls,Area_Acres,-1,-1;Comment \"Comment\" true true +\
          false 20 Text 0 0 ,First,#,wcs,Comment,-1,-1,wls,Comment,+\
          -1,-1;BUFF_DIST \"BUFF_DIST\" true true false 0 Double 0 +\
          0 ,First,#,wc_25ft_buf,BUFF_DIST,-1,-1,wl_25ft_buf,+\
          BUFF_DIST,-1,-1,wl_ca_25ft_buf,BUFF_DIST,-1,-1;+\
          Shape_length_1 \"Shape_length_1\" true true false 0 +\
          Double 0 0 ,First,#,wc_25ft_buf,Shape_length_1,-1,-1,+\
          wl_25ft_buf,Shape_length_1,-1,-1,wl_ca_25ft_buf,+\
          Shape_length_1,-1,-1;Shape_area_1 \"Shape_area_1\" true +\
          true false 0 Double 0 0 ,First,#,wc_25ft_buf,Shape_area_1,+\
          -1,-1,wl_25ft_buf,Shape_area_1,-1,-1,wl_ca_25ft_buf,+\
          Shape_area_1,-1,-1;Shape_length_12 \"Shape_length_12\" +\
          true true false 0 Double 0 0 ,First,#,wc_25ft_buf,+\
          Shape_length_12,-1,-1,wl_25ft_buf,Shape_length_12,-1,-1,+\
          wl_ca_25ft_buf,Shape_length_12,-1,-1;Shape_area_12 +\
          \"Shape_area_12\" true true false 0 Double 0 0 ,First,#,+\
          wc_25ft_buf,Shape_area_12,-1,-1,wl_25ft_buf,Shape_area_12,+\
          -1,-1,wl_ca_25ft_buf,Shape_area_12,-1,-1;ORIG_FID +\
          \"ORIG_FID\" true true false 0 Long 0 0 ,First,#,+\
          wc_25ft_buf,ORIG_FID,-1,-1,wl_25ft_buf,ORIG_FID,-1,-1,+\
          wl_ca_25ft_buf,ORIG_FID,-1,-1;Shape_length_12_13 +\
          \"Shape_length_12_13\" true true false 0 Double 0 0 +\
          ,First,#,wl_ca_25ft_buf,Shape_length_12_13,-1,-1;+\
          Shape_area_12_13 \"Shape_area_12_13\" true true false 0 +\
          Double 0 0 ,First,#,wl_ca_25ft_buf,Shape_area_12_13,-1,-1")

# Explode sensitive areas to form single part features
arcpy.MultipartToSinglepart_management(mrg_lyr1, mrg_lyr)

# Dissolve sensitive areas
arcpy.Dissolve_management(mrg_lyr, mrg_lyr_Dissolve)

# ACCESS ROAD MATTING
# Intersect access roads with dissolved sensitive areas to determine access road
# segments that require matting placement
```

```
arcpy.Intersect_analysis([mrg_lyr_Dissolve, accRds], mrg_lyr_acc_int, "ALL", "",\
                    "INPUT")

# Create 14-ft wide matting area along affected access road segments
arcpy.Buffer_analysis(mrg_lyr_acc_int, acc_matting_14ft_1, "7 Feet", "FULL", \
                  "FLAT")

# Explode access road matting areas to form single part features
arcpy.MultipartToSinglepart_management(acc_matting_14ft_1, acc_matting_14ft_2)

# Dissolve
arcpy.Dissolve_management(acc_matting_14ft_2, acc_matting_14ft)



#POLE MATTING AREAS

#Output matting datasets
piMatting = outputFolder + "\\piMatting"
prMatting = outputFolder + "\\prMatting"

#POLES INSTALLED - Construct 48x56' matting area centered on each pole to be
#installed that falls within a sensitive area

#Convert polesInstall to Feature Layer
arcpy.MakeFeatureLayer_management(polesInstall, 'pi_lyr')

#Select by Location: polesInstall that intersect sensitive areas (mrg_lyr) -
#determines which poles need matting
arcpy.SelectLayerByLocation_management('pi_lyr', 'intersect', mrg_lyr)

#When using SearchCursor to iterate through each row of pi_lyr, grab X and Y
#coordinates of each location
fields = ['SHAPE@X', 'SHAPE@Y']

#Create empty array to store XY coordinate pairs for the 4 point locations
#around each pole in pi_lyr used to construct matting polygon
array = arcpy.Array()

#Create empty list to store new matting polygons around each pole in pi_lyr
featureList = []

#For each row in pi_lyr, establish NW, NE, SE, and SW coordinates used to
#construct matting polygon and construct Point objects from coordinates
with arcpy.da.SearchCursor('pi_lyr', fields) as cursor:
    for row in cursor:
```

```
        array = arcpy.Array()

        xNW = row[0] - 24
        yNW = row[1] + 28
        xNE = row[0] + 24
        yNE = row[1] + 28
        xSE = row[0] + 24
        ySE = row[1] - 28
        xSW = row[0] - 24
        ySW = row[1] - 28

        pointList = [[xNW,yNW], [xNE,yNE], [xSE,ySE], [xSW,ySW]]

        #New Point object
        point = arcpy.Point()

        #add XY coordinate pairs from pointList into array
        for pt in pointList:
            point.X = pt[0]
            point.Y = pt[1]
            pnt = arcpy.Point(point.X, point.Y)
            #array = arcpy.ARRAY(pnt)
            array.add(pnt)

        #In order to properly close new polygon,add in first point of array again
        array.add(array.getObject(0))

        #Create new matting polygon
        boundPolygon = arcpy.Polygon(array)

        #Add new matting polygon to working list
        featureList.append(boundPolygon)

        #array.removeAll()

#Once new matting polygons have been created for pole, add polygons to new
#piMatting feature class
arcpy.CopyFeatures_management(featureList, piMatting)


#POLES REMOVED - Construct 48x56' matting area centered on each pole to be
#removed that falls within a sensitive area
#Convert polesRemove to Feature Layer
arcpy.MakeFeatureLayer_management(polesRemove, 'pr_lyr')

#Select by Location: polesRemove that intersect sensitive areas (mrg_lyr) -
```

```
#determines which poles need matting
arcpy.SelectLayerByLocation_management('pr_lyr', 'intersect', mrg_lyr)

#When using SearchCursor to iterate through each row of pr_lyr, grab X and Y
#coordinates of each location
fields = ['SHAPE@X', 'SHAPE@Y']

#Create empty array to store XY coordinate pairs for the 4 point locations
#around each pole in pr_lyr used to construct matting polygon
array = arcpy.Array()

#Create empty list to store new matting polygons around each pole in pr_lyr
featureList = []

#for each row in pi_lyr, establish NW, NE, SE, and SW coordinates used to
#construct matting polygon and construct Point objects from coordinates
with arcpy.da.SearchCursor('pr_lyr', fields) as cursor:
    for row in cursor:
        #array = arcpy.Array()

        xNW = row[0] - 24
        yNW = row[1] + 28
        xNE = row[0] + 24
        yNE = row[1] + 28
        xSE = row[0] + 24
        ySE = row[1] - 28
        xSW = row[0] - 24
        ySW = row[1] - 28

        pointList = [[xNW,yNW], [xNE,yNE], [xSE,ySE], [xSW,ySW]]

        #New Point object
        point = arcpy.Point()

        #add XY coordinate pairs from pointList into array
        for pt in pointList:
            point.X = pt[0]
            point.Y = pt[1]
            pnt = arcpy.Point(point.X, point.Y)
            array = arcpy.ARRAY(pnt)
            #array.add(pnt)

        #In order to properly close new polygon,add in first point of array again
        array.add(array.getObject(0))

        #Create new matting polygon
```

```
    boundPolygon = arcpy.Polygon(array)

    #Add new matting polygon to working list
    featureList.append(boundPolygon)

    array.removeAll()

#Once new matting polygons have been created for pole, add polygons to new
#piMatting feature class
arcpy.CopyFeatures_management(featureList, prMatting)

print "COMPLETE"
```

Appendix B: Script #2 “calcsOutput2” Python Code

```
# -*- coding: utf-8 -*-
# ---------------------------------------------------------------------------
# calcsOutput2.py
# Created on: 2013-07-28 00:20:15.00000
#   (generated by ArcGIS/ModelBuilder)
# Description:
# ---------------------------------------------------------------------------

# Import arcpy, xlrd, xlwt, and xlutils modules
import arcpy, xlwt, math
from arcpy import env

from xlutils.copy import copy
from xlrd import open_workbook
from xlwt import easyxf, Workbook, Formula, XFStyle, Borders, Pattern, Font

arcpy.env.overwriteOutput = 'True'

# Input Parameters
workingGdb = arcpy.GetParameterAsText(0)
#numWL = arcpy.GetParameterAsText(1) #27
##numWC = arcpy.GetParameterAsText(2) #14

# Local variables:
acc_matting_14ft = workingGdb + "\\Outputs\\acc_matting_14ft"
piMatting = workingGdb + "\\Outputs\\piMatting"
prMatting = workingGdb + "\\Outputs\\prMatting"
mrg_lyr = workingGdb + "\\Outputs\\mrg_lyr"
wlcaBufs = workingGdb + "\\Outputs\\wlcaBufs"
matting_mrg = workingGdb + "\\Outputs\\matting_mrg"
impacts = workingGdb + "\\Outputs\\impacts"
matting_mrg_Dissolve = workingGdb + "\\Outputs\\matting_mrg_Dissolve"
wlImpacts = "wlImpacts"
wlImpactsSums = workingGdb + "\\wlImpactsSums"
wlcabufImpacts = "wlcabufImpacts"
wlcabufImpactsSums = workingGdb + "\\wlcabufImpactsSums"
wcrpbufImpacts = "wcrpbufImpacts"
wcrpbufImpactsSums = workingGdb + "\\wcrpbufImpactsSums"
polesInstall = workingGdb + "\\Inputs\\polesInstall"
wls = workingGdb + "\\Inputs\\wls"
wlcaBufs = workingGdb + "\\Outputs\\wlca_buffers_mrg_25ft"
wcrpBufs = workingGdb + "\\Outputs\\wc_rpbuffers25ft_mrg"
poleWLCounts = workingGdb + "\\Outputs\\poleWLCounts"
```

```
poleWLBufCounts = workingGdb + "\\Outputs\\poleWLBufCounts"
poleWCripBufCounts = workingGdb + "\\Outputs\\poleWCripBufCounts"
pa = workingGdb + "\\Inputs\\projArea"
wlsAreaSum = workingGdb + "\\wlsAreaSum"
mdWLSAreaSum = workingGdb + "\\mdWLSAreaSum"
mdWLSbufAreaSum = workingGdb + "\\mdWLSbufAreaSum"
pa_buffer = workingGdb + "\\Outputs\\pa_buffer"
cas = workingGdb + "\\Inputs\\critAreas"
casPAareaSum = workingGdb + "\\casPAareaSum"
casPAclip = workingGdb + "\\Scratch\\casPAclip"
mdWLS_clip = workingGdb + "\\Outputs\\mdWLS_clip"
mdWLS_buf = workingGdb + "\\Outputs\\mdWLS_buffer"
mdWSSC_clip = workingGdb + "\\Outputs\\mdWSSC_clip"
mdWSSC_buf = workingGdb + "\\Outputs\\mdWSSC_buffer"
mdWSSC_buffer = workingGdb + "\\Outputs\\mdWSSC_buffer"
mdWSSCareaSum = workingGdb + "\\mdWSSCareaSum"
mdWSSCbuf_areaSum = workingGdb + "\\mdWSSCbuf_areaSum"
veg = workingGdb + "\\Inputs\\veg"
vegAreaSums = workingGdb + "\\vegAreaSums"
vegMatInt = workingGdb + "\\Outputs\\vegMatInt"
vegAreaImpactSums = workingGdb + "\\vegAreaImpactSums"
polesRemove = workingGdb + "\\Inputs\\polesRemove"
prVegSpJoin = workingGdb + "\\Outputs\\prVegSpJoin"
prVegTypeCounts = workingGdb + "\\prVegTypeCounts"
polesInstall = workingGdb + "\\Inputs\\polesInstall"
piVegSpJoin = workingGdb + "\\Outputs\\piVegSpJoin"
piVegTypeCounts = workingGdb + "\\piVegTypeCounts"

# Merge matting layers for access paths, poles removed, and poles installed
arcpy.Merge_management([acc_matting_14ft, piMatting, prMatting], matting_mrg, \
                "Shape_Length \"Shape_Length\" false true true 8 Double +\
                0 0 ,First,#,piMatting,Shape_Length,-1,-1,prMatting, +\
                Shape_Length,-1,-1,acc_matting_14ft,Shape_Length,-1,-1;+\
                Shape_Area \"Shape_Area\" false true true 8 Double 0 0 , +\
                First,#,piMatting,Shape_Area,-1,-1,prMatting,Shape_Area, +\
                -1,-1,acc_matting_14ft,Shape_Area,-1,-1")

# Dissolve merged matting layer to remove overlapping areas
arcpy.Dissolve_management(matting_mrg, matting_mrg_Dissolve, "", "", \
                  "SINGLE_PART", "DISSOLVE_LINES")

# Intersect sensitive areas merged layer with new matting layer
arcpy.Intersect_analysis([mrg_lyr, matting_mrg_Dissolve], impacts, "ALL", "", \
                 "INPUT")
```

```
fieldArgs = "OBJECTID OBJECTID VISIBLE NONE;FID_mrg_lyr FID_mrg_lyr
          VISIBLE +\
NONE;Shape Shape VISIBLE NONE;Num Num VISIBLE NONE;ORIG_FID
          ORIG_FID VISIBLE +\
NONE;Type Type VISIBLE NONE;ID ID VISIBLE NONE;ET_ID ET_ID VISIBLE
          NONE;Pre +\
Pre VISIBLE NONE;Area_Acres Area_Acres VISIBLE NONE;Comment Comment
          VISIBLE +\
NONE;Shape_Length Shape_Length VISIBLE NONE;Shape_Area Shape_Area
          VISIBLE NONE; +\
Shape_length Shape_length VISIBLE NONE;Shape_area Shape_area VISIBLE NONE"

# Make table view for impacted wetland areas
arcpy.MakeTableView_management(impacts, wlImpacts, "\"Type\" = 'WL'", "", \
                          fieldArgs)

# For each Wetland ID (Num field), summarize the wetland impacts Shape_Area field
# to a gdb table
arcpy.Statistics_analysis(wlImpacts, wlImpactsSums, "Shape_Area SUM", "Num")

# Make table view for combined impacted wetland buffer and wetland in Critical
# Area buffer areas
arcpy.MakeTableView_management(impacts, wlcabufImpacts, "\"Type\" in ('wlbuf', +\
'cabuf')", "", fieldArgs)

# For each Wetland ID (Num field), summarize the combined wetland buffer and
# wetland in Critical Area buffer impacts Shape_Area field
arcpy.Statistics_analysis(wlcabufImpacts, wlcabufImpactsSums, "Shape_Area SUM",\
                      "Num")

# Make table view for combined impacted watercourse and riparian buffer areas
arcpy.MakeTableView_management(impacts, wcrpbufImpacts, "\"Type\" in ('WC', +\
'rpbuf')", "", fieldArgs)

# For each Watercourse ID (Num field), summarize the combined watercourse and
# riparian buffer impacts Shape_Area field
arcpy.Statistics_analysis(wcrpbufImpacts, wcrpbufImpactsSums, "Shape_Area SUM",\
                      "Num")

## Use mrg_lyr and include Type field - use output to populate pole count/impacts
## cells
# Get counts for number of poles to be installed in each wetland complex
arcpy.SpatialJoin_analysis(polesInstall, wls, poleWLCounts, "JOIN_ONE_TO_ONE", \
                      "KEEP_COMMON", "ET_ID \"ET_ID\" true true false 4 Long \
                      0 0 ,First,#," + polesInstall + ",ET_ID,-1,-1;Radius +\
                      \"Radius\" true true false 8 Double 0 0 ,First,#," \
```

```
                    polesInstall + ",Radius,-1,-1;Num \"Num\" true true +\
                    false 2 Short 0 0 ,First,#," + wls + ",Num,-1,-1; +\
                    Shape_Area \"Shape_Area\" false true true 8 Double 0 +\
                    0 ,First,#," + wls + ",Shape_Area,-1,-1", "INTERSECT",\
                    "", "")

# Make Feature Layer of combined wetland buffer and wetland in Critical Area
# complexes
arcpy.MakeFeatureLayer_management(mrg_lyr, wlcaBufs)
arcpy.SelectLayerByAttribute_management(wlcaBufs, "NEW_SELECTION", "\"Type\"
        in +\
('wlbuf', 'cabuf')")

# Get counts for number of poles to be installed in each wetland buffer/wetland
# in Critical Area buffer complex
arcpy.SpatialJoin_analysis(polesInstall, wlcaBufs, poleWLBufCounts, \
                    "JOIN_ONE_TO_ONE", "KEEP_COMMON", "ET_ID \"ET_ID\" +\
                    true true false 4 Long 0 0 ,First,#," + polesInstall + \
                    ",ET_ID,-1,-1;Radius \"Radius\" true true false 8 +\
                    Double 0 0 ,First,#," + polesInstall + ",Radius,-1,+\
                    -1;Num \"Num\" true true false 2 Short 0 0 ,First,#," \
                    + wlcaBufs + ",Num,-1,-1;Shape_Area \"Shape_Area\" +\
                    false true true 8 Double 0 0 ,First,#," + wlcaBufs + \
                    ",Shape_Area,-1,-1", "INTERSECT", "", "")

# Make Feature Layer of combined watercourse and riparian buffer complexes
arcpy.MakeFeatureLayer_management(mrg_lyr, wcrpBufs)
arcpy.SelectLayerByAttribute_management(wcrpBufs, "NEW_SELECTION", "\"Type\"
        in +\
('WC', 'rpbuf')")

# Get counts for number of poles to be installed in each watercourse/riparian buffer
        complex
arcpy.SpatialJoin_analysis(polesInstall, wcrpBufs, poleWCripBufCounts, +\
                    "JOIN_ONE_TO_ONE", "KEEP_COMMON", "Radius \"Radius\" \
                    true true false 8 Double 0 0 ,First,#," + polesInstall \
                    + ",Radius,-1,-1;Num \"Num\" true true false 2 Short +\
                    0 0 ,First,#," + wcrpBufs + ",Num,-1,-1;Type \"Type\" +\
                    true true false 5 Text 0 0,First,#," + wcrpBufs + ", +\
                    Type,-1,-1", "INTERSECT", "", "")

##EXCEL TABLE OUTPUT
filePath = "E:\NW_GradSchool\Thesis\THESIS\output_tables\C-T_IP_Tables_blank.xls"

#Open existing template Workbook
rb = open_workbook(filePath,formatting_info=True)
```

```
r_sheet = rb.sheet_by_index(0)
wb = copy(rb)

#Create cell style objects to address formatting for font type, bold, alignment,
# borders, and decimal places
fnt = Font()
fnt.name = 'Arial'
fntBold = Font()
fntBold.name = 'Arial'
fntBold.bold = True

align = xlwt.Alignment()
align.horz = xlwt.Alignment.HORZ_CENTER

borders = Borders()
borders.left = Borders.THIN
borders.right = Borders.THIN
borders.top = Borders.THIN
borders.bottom = Borders.THIN

style = XFStyle()
style.num_format_str = '0'
style.font = fnt
style.borders = borders
style.alignment = align

styleB = XFStyle()
styleB.num_format_str = '0'
styleB.font = fntBold
styleB.borders = borders
styleB.alignment = align

style2 = XFStyle()
style2.num_format_str = '0.00'
style2.font = fnt
style2.borders = borders
style2.alignment = align

style2perc = XFStyle()
style2perc.num_format_str = '0.00%'
style2perc.font = fnt
style2perc.borders = borders
style2perc.alignment = align

style2percB = XFStyle()
style2percB.num_format_str = '0.00%'
```

```
style2percB.font = fntBold
style2percB.borders = borders
style2percB.alignment = align

style2B = XFStyle()
style2B.num_format_str = '0.00'
style2B.font = fntBold
style2B.borders = borders
style2B.alignment = align

style4 = XFStyle()
style4.num_format_str = '0.0000'
style4.font = fnt
style4.borders = borders
style4.alignment = align

style4B = XFStyle()
style4B.num_format_str = '0.0000'
style4B.font = fntBold
style4B.borders = borders
style4B.alignment = align

styleText = XFStyle()
styleText.font = fnt
styleText.borders = borders

#Populate Wetland Impacts table - Temporary Matting columns
w_sheet = wb.get_sheet(0)

fields = ["Num", "SUM_Shape_Area"]
with arcpy.da.SearchCursor(wlImpactsSums, fields) as cursor:
    for row in cursor:
        wlNum = row[0]
        xlRow = wlNum + 2
        w_sheet.write(xlRow,3,row[1],style2)
        acres = row[1]/43560
        w_sheet.write(xlRow,4,acres,style4)
#change row number to
w_sheet.write(30,3,Formula('sum(d4:d30)'),style2B)
w_sheet.write(30,4,Formula('sum(e4:e30)'),style4B)

# Populate Wetland Impacts table - Number of Installed Poles/Pole Impacts columns
# Add wetland IDs into a Python list
wlIDList = []

fields = ["Num"]
```

```
with arcpy.da.SearchCursor(poleWLCounts, fields) as cursor:
    for row in cursor:
        if row[0] not in wlIDList:
            wlIDList.append(row[0])
        else:
            pass

fields = ["Num", "Radius"]
for ID in wlIDList:
    numPolesSmall = 0
    numPolesLarge = 0
    with arcpy.da.SearchCursor(poleWLCounts, fields) as cursor:
        for row in cursor:
            if ID == row[0]:
                if row[1] == 3.5:
                    numPolesSmall += 1
                elif row[1] == 8.5:
                    numPolesLarge += 1
            else:
                pass
    xlRow = ID + 2
    if numPolesSmall > 0:
        w_sheet.write(xlRow,11,numPolesSmall,style)
        poleImpacts = numPolesSmall * math.pi * 1.75**2
        w_sheet.write(xlRow,5,poleImpacts,style2)
        acres = poleImpacts / 43560
        w_sheet.write(xlRow,6,acres,style4)
    else:
        w_sheet.write(xlRow,5,0,style2)
        w_sheet.write(xlRow,6,0,style4)

    if numPolesLarge > 0:
        w_sheet.write(xlRow,12,numPolesLarge,style)
        poleImpacts = numPolesLarge * math.pi * 4.25**2
        w_sheet.write(xlRow,7,poleImpacts,style2)
        acres = poleImpacts / 43560
        w_sheet.write(xlRow,8,acres,style4)
    else:
        w_sheet.write(xlRow,7,0,style2)
        w_sheet.write(xlRow,8,0,style4)

for i in range(3,30):
    w_sheet.write(i,9,Formula("$F$%d+$H$%d" % (i+1, i+1)),style2)
    w_sheet.write(i,10,Formula("$G$%d+$I$%d" % (i+1, i+1)),style4)

#Sum total impacts from poles and foundations
```

```
w_sheet.write(30,5,Formula('sum(f4:f30)'),style2B)
w_sheet.write(30,6,Formula('sum(g4:g30)'),style4B)
w_sheet.write(30,7,Formula('sum(h4:h30)'),style2B)
w_sheet.write(30,8,Formula('sum(i4:i30)'),style4B)
w_sheet.write(30,9,Formula('sum(j4:j30)'),style2B)
w_sheet.write(30,10,Formula('sum(k4:k30)'),style4B)

#Sum pole count columns
w_sheet.write(30,11,Formula('sum(L4:L30)'),styleB)
w_sheet.write(30,12,Formula('sum(m4:m30)'),styleB)

#----------------------------------------------------------------------------

#Populate Wetland Buffer Impacts table - Temporary Matting columns
w_sheet = wb.get_sheet(1)

fields = ["Num", "SUM_Shape_Area"]
with arcpy.da.SearchCursor(wlcabufImpactsSums, fields) as cursor:
    for row in cursor:
        wlNum = row[0]
        xlRow = wlNum + 2
        w_sheet.write(xlRow,3,row[1],style2)
        acres = row[1]/43560
        w_sheet.write(xlRow,4,acres,style4)
#change row number to
w_sheet.write(30,3,Formula('sum(d4:d30)'),style2B)
w_sheet.write(30,4,Formula('sum(e4:e30)'),style4B)

# Populate Wetland Buffer Impacts table - Number of Installed Poles/Pole Impacts
# columns
# Add wetland IDs into a Python list
wlBufIDList = []

fields = ["Num"]
with arcpy.da.SearchCursor(poleWLBufCounts, fields) as cursor:
    for row in cursor:
        if row[0] not in wlBufIDList:
            wlBufIDList.append(row[0])
        else:
            pass

fields = ["Num", "Radius"]
for ID in wlBufIDList:
    numPolesSmall = 0
    numPolesLarge = 0
    with arcpy.da.SearchCursor(poleWLBufCounts, fields) as cursor:
```

```
        for row in cursor:
            if ID == row[0]:
                if row[1] == 3.5:
                    numPolesSmall += 1
                elif row[1] == 8.5:
                    numPolesLarge += 1
            else:
                pass
    xlRow = ID + 2
    if numPolesSmall > 0:
        w_sheet.write(xlRow,11,numPolesSmall,style)
        poleImpacts = numPolesSmall * math.pi * 1.75**2
        w_sheet.write(xlRow,5,poleImpacts,style2)
        acres = poleImpacts / 43560
        w_sheet.write(xlRow,6,acres,style4)
    else:
        w_sheet.write(xlRow,5,0,style2)
        w_sheet.write(xlRow,6,0,style4)

    if numPolesLarge > 0:
        w_sheet.write(xlRow,12,numPolesLarge,style)
        poleImpacts = numPolesLarge * math.pi * 4.25**2
        w_sheet.write(xlRow,7,poleImpacts,style2)
        acres = poleImpacts / 43560
        w_sheet.write(xlRow,8,acres,style4)
    else:
        w_sheet.write(xlRow,7,0,style2)
        w_sheet.write(xlRow,8,0,style4)

for i in range(3,30):
    w_sheet.write(i,9,Formula("$F$%d+$H$%d" % (i+1, i+1)),style2)
    w_sheet.write(i,10,Formula("$G$%d+$I$%d" % (i+1, i+1)),style4)

#Sum total impacts from poles and foundations
w_sheet.write(30,5,Formula('sum(f4:f30)'),style2B)
w_sheet.write(30,6,Formula('sum(g4:g30)'),style4B)
w_sheet.write(30,7,Formula('sum(h4:h30)'),style2B)
w_sheet.write(30,8,Formula('sum(i4:i30)'),style4B)
w_sheet.write(30,9,Formula('sum(j4:j30)'),style2B)
w_sheet.write(30,10,Formula('sum(k4:k30)'),style4B)

#Sum pole count columns
w_sheet.write(30,11,Formula('sum(L4:L30)'),styleB)
w_sheet.write(30,12,Formula('sum(m4:m30)'),styleB)

#---------------------------------------------------------------------------
```

```
#Populate Watercourse & 25-ft Riparian Buffer Impacts table - Temporary Matting
#columns
w_sheet = wb.get_sheet(2)

fields = ["Num", "SUM_Shape_Area"]
with arcpy.da.SearchCursor(wcrpbufImpactsSums, fields) as cursor:
    for row in cursor:
        wcNum = row[0]
        xlRow = wcNum + 2
        w_sheet.write(xlRow,3,row[1],style2)
        acres = row[1]/43560
        w_sheet.write(xlRow,4,acres,style4)
#change row number to
w_sheet.write(17,3,Formula('sum(d4:d17)'),style2B)
w_sheet.write(17,4,Formula('sum(e4:e17)'),style4B)

# Populate Watercourse & 25-ft Riparian Buffer Impacts table - Number of Installed
# Poles/Pole Impacts columns
# Add wetland IDs into a Python list
wcRipBufIDList = []

fields = ["Num"]
with arcpy.da.SearchCursor(poleWCripBufCounts, fields) as cursor:
    for row in cursor:
        if row[0] not in wcRipBufIDList:
            wcRipBufIDList.append(row[0])
        else:
            pass

fields = ["Num", "Radius"]
for ID in wcRipBufIDList:
    numPolesSmall = 0
    numPolesLarge = 0
    with arcpy.da.SearchCursor(poleWCripBufCounts, fields) as cursor:
        for row in cursor:
            if ID == row[0]:
                if row[1] == 3.5:
                    numPolesSmall += 1
                elif row[1] == 8.5:
                    numPolesLarge += 1
            else:
                pass
    xlRow = ID + 2
    if numPolesSmall > 0:
        w_sheet.write(xlRow,11,numPolesSmall,style)
```

```
        poleImpacts = numPolesSmall * math.pi * 1.75**2
        w_sheet.write(xlRow,5,poleImpacts,style2)
        acres = poleImpacts / 43560
        w_sheet.write(xlRow,6,acres,style4)
    else:
        w_sheet.write(xlRow,5,0,style2)
        w_sheet.write(xlRow,6,0,style4)

    if numPolesLarge > 0:
        w_sheet.write(xlRow,12,numPolesLarge,style)
        poleImpacts = numPolesLarge * math.pi * 4.25**2
        w_sheet.write(xlRow,7,poleImpacts,style2)
        acres = poleImpacts / 43560
        w_sheet.write(xlRow,8,acres,style4)
    else:
        w_sheet.write(xlRow,7,0,style2)
        w_sheet.write(xlRow,8,0,style4)

for i in range(3,17):
    w_sheet.write(i,9,Formula("$F$%d+$H$%d" % (i+1, i+1)),style2)
    w_sheet.write(i,10,Formula("$G$%d+$I$%d" % (i+1, i+1)),style4)

#Sum total impacts from poles and foundations
w_sheet.write(17,5,Formula('sum(f4:f17)'),style2B)
w_sheet.write(17,6,Formula('sum(g4:g17)'),style4B)
w_sheet.write(17,7,Formula('sum(h4:h17)'),style2B)
w_sheet.write(17,8,Formula('sum(i4:i17)'),style4B)
w_sheet.write(17,9,Formula('sum(j4:j17)'),style2B)
w_sheet.write(17,10,Formula('sum(k4:k17)'),style4B)

#Sum pole count columns
w_sheet.write(17,11,Formula('sum(L4:L17)'),styleB)
w_sheet.write(17,12,Formula('sum(m4:m17)'),styleB)


# VEGETATION IMPACT CALCULATIONS
# Output each vegetation class and its respective aggregate area in acres to table
w_sheet = wb.get_sheet(4)

arcpy.Statistics_analysis(veg, vegAreaSums, "Shape_Area SUM", "Veg_Class")
vegTypeList = []
xlRowType = 1
xlColType = 0
fields = ["Veg_Class", "SUM_Shape_Area"]
with arcpy.da.SearchCursor(vegAreaSums, fields) as cursor:
    for row in cursor:
```

```
        vegTypeList.append(row[0])
        w_sheet.write(xlRowType,xlColType,row[0],styleText)
        xlColArea = xlColType + 1
        areaAcres = row[1] / 43560
        w_sheet.write(xlRowType,xlColArea,areaAcres,style4)
        xlRowType += 1
w_sheet.write(13,1,Formula('sum(B2:B13)'),style4B)

# Output Temporary Impacts of each vegetation class in acres to table
arcpy.Intersect_analysis([matting_mrg_Dissolve,veg], vegMatInt)
arcpy.Statistics_analysis(vegMatInt, vegAreaImpactSums, "Shape_Area SUM", \
                          "Veg_Class")
vegTypeTempImpactsList = []
vegTypeTempImpactsAreaList = []
xlRowTempImpacts = 1
xlColTempImpacts = 2
fields = ["Veg_Class", "SUM_Shape_Area"]
with arcpy.da.SearchCursor(vegAreaImpactSums, fields) as cursor:
    for row in cursor:
        vegTypeTempImpactsList.append(row[0])
        vegTypeTempImpactsAreaList.append(row[1])
for i in range(1,4):
    vegTypeTempImpactsList.append(i)
    vegTypeTempImpactsAreaList.append(i)

for i in range(0,12):
    if vegTypeList[i] == vegTypeTempImpactsList[i]:
        areaTempImpactsAcres = vegTypeTempImpactsAreaList[i] / 43560
        w_sheet.write(xlRowTempImpacts,xlColTempImpacts,areaTempImpactsAcres,\
                      style4)
        xlRowTempImpacts += 1
    elif vegTypeList[i] == vegTypeTempImpactsList[i-1]:
        areaTempImpactsAcres = vegTypeTempImpactsAreaList[i-1] / 43560
        w_sheet.write(xlRowTempImpacts,xlColTempImpacts,areaTempImpactsAcres,\
                      style4)
        xlRowTempImpacts += 1
    elif vegTypeList[i] == vegTypeTempImpactsList[i-2]:
        areaTempImpactsAcres = vegTypeTempImpactsAreaList[i-2] / 43560
        w_sheet.write(xlRowTempImpacts,xlColTempImpacts,areaTempImpactsAcres,\
                      style4)
        xlRowTempImpacts += 1
    elif vegTypeList[i] == vegTypeTempImpactsList[i-3]:
        areaTempImpactsAcres = vegTypeTempImpactsAreaList[i-3] / 43560
        w_sheet.write(xlRowTempImpacts,xlColTempImpacts,areaTempImpactsAcres,\
                      style4)
        xlRowTempImpacts += 1
```

```
    else:
        w_sheet.write(xlRowTempImpacts,xlColTempImpacts,0,style4)
        xlRowTempImpacts += 1

w_sheet.write(13,2,Formula('sum(C2:C13)'),style4B)

# Output Number of Poles to be Removed and Installed for each vegetation class
# Poles Removed
arcpy.SpatialJoin_analysis(polesRemove, veg, prVegSpJoin, "JOIN_ONE_TO_ONE", \
                           "KEEP_ALL", "Veg_Class \"Veg_Class\" +\
                           true true false 20 Text 0 0 ,First,#," + veg + ", +\
                           Veg_Class,-1,-1", "INTERSECT", "", "")

arcpy.Statistics_analysis(prVegSpJoin, prVegTypeCounts, "Veg_Class COUNT", \
                          "Veg_Class")
prVegTypeList = []
prVegTypeCountList = []
xlRowPRcount = 1
xlColPRcount = 3
fields = ["Veg_Class", "COUNT_Veg_Class"]
with arcpy.da.SearchCursor(prVegTypeCounts, fields) as cursor:
    for row in cursor:
        prVegTypeList.append(row[0])
        prVegTypeCountList.append(row[1])
for i in range(1,4):
    prVegTypeList.append(i)
    prVegTypeCountList.append(i)

for i in range(0,12):
    if vegTypeList[i] == prVegTypeList[i]:
        prCount = prVegTypeCountList[i]
        w_sheet.write(xlRowPRcount,xlColPRcount,prCount,style)
        xlRowPRcount += 1
    elif vegTypeList[i] == prVegTypeList[i-1]:
        prCount = prVegTypeCountList[i-1]
        w_sheet.write(xlRowPRcount,xlColPRcount,prCount,style)
        xlRowPRcount += 1
    elif vegTypeList[i] == prVegTypeList[i-2]:
        prCount = prVegTypeCountList[i-2]
        w_sheet.write(xlRowPRcount,xlColPRcount,prCount,style)
        xlRowPRcount += 1
    elif vegTypeList[i] == prVegTypeList[i-3]:
        prCount = prVegTypeCountList[i-3]
        w_sheet.write(xlRowPRcount,xlColPRcount,prCount,style)
        xlRowPRcount += 1
    else:
```

```
        w_sheet.write(xlRowPRcount,xlColPRcount,0,style)
        xlRowPRcount += 1

w_sheet.write(13,3,Formula('sum(D2:D13)'),styleB)

# Poles Installed
arcpy.SpatialJoin_analysis(polesInstall, veg, piVegSpJoin, "JOIN_ONE_TO_ONE", \
                           "KEEP_ALL", "Veg_Class \"Veg_Class\" true true false +\
                           20 Text 0 0 ,First,#," + veg + ",Veg_Class,-1,-1", \
                           "INTERSECT", "", "")

arcpy.Statistics_analysis(piVegSpJoin, piVegTypeCounts, "Veg_Class COUNT", \
                          "Veg_Class")
piVegTypeList = []
piVegTypeCountList = []
xlRowPIcount = 1
xlColPIcount = 4
fields = ["Veg_Class", "COUNT_Veg_Class"]
with arcpy.da.SearchCursor(piVegTypeCounts, fields) as cursor:
    for row in cursor:
        piVegTypeList.append(row[0])
        piVegTypeCountList.append(row[1])
for i in range(1,6):
    piVegTypeList.append(i)
    piVegTypeCountList.append(i)

for i in range(0,12):
    if vegTypeList[i] == piVegTypeList[i]:
        piCount = piVegTypeCountList[i]
        w_sheet.write(xlRowPIcount,xlColPIcount,piCount,style)
        xlRowPIcount += 1
    elif vegTypeList[i] == piVegTypeList[i-1]:
        piCount = piVegTypeCountList[i-1]
        w_sheet.write(xlRowPIcount,xlColPIcount,piCount,style)
        xlRowPIcount += 1
    elif vegTypeList[i] == piVegTypeList[i-2]:
        piCount = piVegTypeCountList[i-2]
        w_sheet.write(xlRowPIcount,xlColPIcount,piCount,style)
        xlRowPIcount += 1
    elif vegTypeList[i] == piVegTypeList[i-3]:
        piCount = piVegTypeCountList[i-3]
        w_sheet.write(xlRowPIcount,xlColPIcount,piCount,style)
        xlRowPIcount += 1
    elif vegTypeList[i] == piVegTypeList[i-4]:
        piCount = piVegTypeCountList[i-4]
        w_sheet.write(xlRowPIcount,xlColPIcount,piCount,style)
```

```
        xlRowPIcount += 1
    elif vegTypeList[i] == piVegTypeList[i-4]:
        piCount = piVegTypeCountList[i-4]
        w_sheet.write(xlRowPIcount,xlColPIcount,piCount,style)
        xlRowPIcount += 1
    elif vegTypeList[i] == piVegTypeList[i-5]:
        piCount = piVegTypeCountList[i-5]
        w_sheet.write(xlRowPIcount,xlColPIcount,piCount,style)
        xlRowPIcount += 1
    else:
        w_sheet.write(xlRowPIcount,xlColPIcount,0,style)
        xlRowPIcount += 1

w_sheet.write(13,4,Formula('sum(E2:E13)'),styleB)

#Populate Miscellaneous Impact Calculations worksheet
w_sheet = wb.get_sheet(3)

# Field Wetlands: Total Area w/in ROW, % ROW classified as Field Wetlands
arcpy.Statistics_analysis(wls, wlsAreaSum, "Shape_Area SUM")
fields = ["SUM_Shape_Area"]
with arcpy.da.SearchCursor(wlsAreaSum, fields) as cursor:
    for row in cursor:
        wlsAreaROW = row[0] / 43560
w_sheet.write(1,1,wlsAreaROW,style4)

fields = ["Shape_Area"]
with arcpy.da.SearchCursor(pa, fields) as cursor:
    for row in cursor:
        rowArea = row[0]
percWlsROW = wlsAreaROW / (rowArea / 43560)
w_sheet.write(2,1,percWlsROW,style2perc)

# Critical Areas: Total Area w/in ROW, % ROW classified as Critical Areas
arcpy.Clip_analysis(cas, pa, casPAclip, "")
arcpy.Statistics_analysis(casPAclip, casPAareaSum, "Shape_Area SUM")
fields = ["SUM_Shape_Area"]
with arcpy.da.SearchCursor(casPAareaSum, fields) as cursor:
    for row in cursor:
        casAreaRow = row[0] / 43560
w_sheet.write(5,1,casAreaRow,style4)
percCAsROW = casAreaRow / (rowArea / 43560)
w_sheet.write(6,1,percCAsROW,style2perc)

# MD DNR State Wetlands and 25-ft buffers: Total Area w/in 1/4-mile ROW buffer,
# % of 1/4-mile ROW buffer
```

```
arcpy.Statistics_analysis(mdWLS_clip, mdWLSAreaSum, "Shape_Area SUM")
fields = ["SUM_Shape_Area"]
with arcpy.da.SearchCursor(mdWLSAreaSum, fields) as cursor:
    for row in cursor:
        mdWLSAreaROWbuf = row[0] / 43560
arcpy.Statistics_analysis(mdWLS_buf, mdWLSbufAreaSum, "Shape_Area SUM")
fields = ["SUM_Shape_Area"]
with arcpy.da.SearchCursor(mdWLSbufAreaSum, fields) as cursor:
    for row in cursor:
        mdWLSbufAreaROWbuf = row[0] / 43560
mdWLSArea = mdWLSAreaROWbuf + mdWLSbufAreaROWbuf
w_sheet.write(9,1,mdWLSArea,style4)

fields = ["Shape_Area"]
with arcpy.da.SearchCursor(pa_buffer, fields) as cursor:
    for row in cursor:
        rowBufArea = row[0]
percMDwlsROWbuf = mdWLSArea / (rowBufArea / 43560)
w_sheet.write(10,1,percMDwlsROWbuf,style2perc)

# MD Wetland of Special State Concern (WSSC) and 100-ft buffers: Total Area w/in
# 1/4-mile ROW buffer, % of 1/4-mile ROW buffer
arcpy.Statistics_analysis(mdWSSC_clip, mdWSSCareaSum, "Shape_Area SUM")
fields = ["SUM_Shape_Area"]
with arcpy.da.SearchCursor(mdWSSCareaSum, fields) as cursor:
    for row in cursor:
        mdWSSCAreaROWbuf = row[0] / 43560
arcpy.Statistics_analysis(mdWSSC_buf, mdWSSCbuf_areaSum, "Shape_Area SUM")
fields = ["SUM_Shape_Area"]
with arcpy.da.SearchCursor(mdWSSCbuf_areaSum, fields) as cursor:
    for row in cursor:
        mdWSSCbufAreaROWbuf = row[0] / 43560
mdWSSCArea = mdWSSCAreaROWbuf + mdWSSCbufAreaROWbuf
w_sheet.write(12,1,mdWSSCArea,style4)

percMDWSSCROWbuf = mdWSSCArea / (rowBufArea / 43560)
w_sheet.write(13,1,percMDWSSCROWbuf,style2perc)

totalArea = mdWLSArea + mdWSSCArea
totalPerc = percMDwlsROWbuf + percMDWSSCROWbuf

w_sheet.write(15,1,totalArea,style4B)
w_sheet.write(16,1,totalPerc,style2percB)

wb.save(filePath)
```

REFERENCES

Afanasiev, A., 2013. *Always Show 4 decimal places* [online]. Available from: http://stackoverflow.com/questions/16176211/always-show-4-decimal-places. [Accessed 4 August 2013].

Andrews, A., 1990. Fragmentation of habitat by roads and utility corridors: a review. *Australian Zoologist*, 26(3&4), 130-141.

Atkinson, S.F. and Canter, L.W., 2011. Assessing the cumulative effects of projects using geographic information systems. *Environmental Impact Assessment Review*, 31, 457-464.

Avraam, M., 2009. *Manipulating Excel files using Python part 2: writing Excel files* [online]. Available at: http://michalisavraam.org/2009/06/manipulating-excel-files-using-python-part-2-writing-files/ [Accessed 4 December 2011].

Bernier, A., 2009. *Calculating formulae in Excel with Python* [online]. Available from: http://stackoverflow.com/questions/1116725/calculating-formulae-in-excel-with-python. [Accessed 4 August 2013].

Black & Veatch, 2011. *Townsend to Church environmental review document*. Overland Park: Black & Veatch.

Black & Veatch, 2013. *Environmental monitoring manual for construction activities – Church to Townsend 138 kV transmission line rebuild project.* Wilmington: Black & Veatch.

Chappell, D., 2007. Remote sensing for vegetation management. *Esri Energy Currents*, Summer 2007 [online]. Available at: http://www.Esri.com/library/newsletters/energycurrents/energysummer2007.pdf#page=8 [Accessed 7 November 2011].

Davidovic, N., Markovic, V., Vasiljevic, D., and Stankov, U., 2010. GIS approach in tourism management in National Park Berchtesgaden (Bayern, Germany). *Geographia Technica*, 2, 18-24.

Earthsafe, 2011. *Tufftrak heavy duty access mat* [online]. Available at: http://earthsafeaccess.com/tufftrak-heavy-equipment-mat.php [Accessed 3 December 2011].

Electric Light & Power, 2010. *Transmission and distribution construction project on heavily regulated wetlands* [online]. Available at: http://www.elp.com/index/display/article-display/4526695758/articles/utility-products/volume-7/issue-8/product-focus/line-construction/transmission-and-distribution-construction-project-on-heavily-regulated-wetlands.html [Accessed 9 November 2011].

Environmental Protection Agency (EPA), 2001. *Wetland monitoring & assessment* [online]. Available at: http://www.epa.gov/owow/wetlands/pdf/monitor.pdf [Accessed 19 November 2011].

Environmental Protection Agency (EPA), 2009. *Wetlands (Section 404) compliance monitoring* [online]. Available at: http://www.epa.gov/compliance/monitoring/programs/cwa/wetlands.html [Accessed 19 November 2011].

Environmental Systems Research Institute (Esri), 2010. *ArcGIS resource center – additional conversion – generic tools* [online]. Available at: http://resources.arcgis.com/gallery/file/geoprocessing/details?entryID=95009B25-1422-2418-7FB5-B8638ECB2FA9 [Accessed 4 December 2011].

Environmental Systems Research Institute (Esri), 2011a. *Desktop help 10.0 – writing geometries* [online]. Available at: http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//002z0000001v000000 [Accessed 4 December 2011].

Environmental Systems Research Institute (Esri), 2011b. *Desktop help 10.0 – what is ModelBuilder?* [online] Available at: http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//002w00000001000000 [Accessed 2 December 2011].

Environmental Systems Research Institute (Esri), 2012. *ArcGIS help 10.1 – extend line (editing)* [online]. Available at: http://resources.arcgis.com/en/help/main/10.1/index.html#//001v00000004000000 [Accessed 6 June 2014].

Environmental Systems Research Institute (Esri), 2013. *ArcGIS help 10.1 – writing geometries* [online]. Available at: http://resources.arcgis.com/en/help/main/10.1/index.html#//002z0000001v000000 [Accessed 6 June 2014].

Examples Generating Excel Documents Using Python's xlwt, 2011 [online]. Available from: http://www.youlikeprogramming.com/2011/04/examples-generating-excel-documents-using-pythons-xlwt/. [Accessed 4 August 2013].

Gunasekera, R., 2004. Use of GIS for environmental impact assessment: an interdisciplinary approach. *Interdisciplinary Science Reviews*, 29(1), 37-48.

Herrero-Jimenez, C.M., 2012. An expert system for the identification of environmental impact based on a geographic information system. *Expert Systems with Applications*, 39(8), 6672-6682.

Kauffman, J. L., 2007. *Development and application of a GIS based evaluation for prioritization of wetland restoration opportunities*. Master's thesis. Humboldt State University [online]. Available at: http://humboldt-dspace.calstate.edu/xmlui/bitstream/handle/2148/278/Jennifer%20Kauffman%20Masters%20Thesis.pdf?sequence=1 [Accessed 19 November 2011].

Kelly, C., 2008. GIS clips hassles in vegetation management for power lines. *Esri Energy Currents*, Summer 2008 [online]. Available at: http://www.Esri.com/library/newsletters/energycurrents/energysummer2008.pdf#page=19 [Accessed 7 November 2011].

Maryland Department of Natural Resources (MD DNR), nd. *Critical Areas Commission for the Chesapeake and Atlantic coastal bays – Maryland* [online]. Available at: http://www.dnr.state.md.us/criticalarea/ [Accessed 2 December 2011].

Maryland Department of Natural Resources (MD DNR), 1997. *Wetlands of Special State Concern metadata* [online]. Available at: http://www.dnr.state.md.us/watersheds/surf/indic/metadata/wssc_met.pdf [Accessed 2 December 2011].

McBurney, S., 2014. *Matting used for construction*. [email] (Personal communication, 6 June 2014).

Meehan, B., 2007. *Empowering electric and gas utilities with GIS*. Redlands: Esri Press. [online] Available at: http://books.google.com/books?id=td7_6X3SK78C&printsec=frontcover&dq=empowering+electric+and+gas+utilities+with+GIS&hl=en#v=onepage&q=empowering%20electric%20and%20gas%20utilities%20with%20GIS&f=false [Accessed 19 November 2011].

Muller, B., Johnson, L., Wycoff, J., Nuszdorfer, F., and Beckham, B., 2007. Area wide cumulative effects analysis using GIS. *In: 86th Annual Meeting of the Transportation Research Board*, January 2007 Washington D.C.

Nkeki, F.N.,Henah, P.J. and Ojeh, V.N., 2013. Geospatial techniques for the assessment and analysis of flood risk along the Niger-Benue basin in Nigeria. *Journal of Geographic Information System*, [online] 5, 123-135. Available at: http://www.scirp.org/journal/PaperDownload.aspx?paperID=29778 [Accessed 8 February 2014].

Pepco Holdings, Inc. (PHI), 2010. Powering a sustainable future – 2010 annual environmental sustainability report. Washington D.C.: Pepco Holdings, Inc.

[online]. Available at: http://www.middournolandesign.com/PepcoSustainRep10Flip/files/inc/503956097.pdf [Accessed 2 December 2011].

Pinero, G., 2010. *Writing to existing workbook using xlwt* [online]. Available at: http://stackoverflow.com/questions/2725852/writing-to-existing-workbook-using-xlwt. [Accessed 28 July 2013].

Python, 2012. [online] Available at: http://wiki.gis.com/wiki/index.php/Python#ArcGIS_10. [Access 2 September 2013].

Savage, M., 2013. *Matting*. [email] (Personal communication, 10 December 2013).

Small, D., 2013. *Matting*. [email] (Personal communication, 23 September 2013).

State of Delaware, 2009. *7502 Wetlands regulations* [online]. Available at: <http://regulations.delaware.gov/AdminCode/title7/7000/7500/7502.shtml#TopOfPage> [Accessed 19 November 2011].

Sutter, L.A, Stanfill, J., Haupt, D., Bruce, C., and Wuenscher, J., 1999. NC-CREWS: North Carolina coastal region evaluation of wetland significance [online]. Available at: http://dcm2.enr.state.nc.us/wetlands/NCCREWSDOC.pdf [Accessed 19 November 2011].

Walbeck, D., Clearwater, D., and Neff, K., 2011. *Maryland nontidal wetland mitigation guidance*. Maryland Department of Environment – Nontidal Wetlands and Waterways Division. [online] Available at: http://www.mde.state.md.us/programs/Water/WetlandsandWaterways/AboutWetlands/Documents/www.mde.state.md.us/assets/document/wetlandswaterways/MITGUIDEfeb72011.pdf [Accessed 19 November 2011].