ERDC/CHL CHETN-X-4
December 2017

US Army Corps
of Engineers®

Development of an Automatic Soil Texture Extraction Routine Using Combined SSURGO and STATSGO2 Dataset

by Kayla A. Cotterman and Nawa Raj Pradhan

PURPOSE: The purpose of this Coastal and Hydraulics Engineering Technical Note (CHETN) is to document an approach that can be utilized for user-specified project sites to create a complete set of soil texture classifications for locations of interest within the United States. This is accomplished by using the Soil Survey Geographic (SSURGO) dataset in combination with the U.S. General Soil Map (STATSGO2) dataset to replace the missing values in the SSURGO dataset. The datasets are separated into tabular and spatial datasets; then, by using Python and Geographic Information Systems (GIS) (analyses as described herein), the tabular data can be added to the spatial data creating a complete soils dataset.

BACKGROUND: When simulating hydrology and geomorphology of past, current, and future climate scenarios, soil data are very important as the data affect the properties of variables and parameters describing physical processes like infiltration, surface runoff, and evaporation. These variables correspond to different soil properties depending upon the soil texture class. Therefore, determining the correct soil coverage and associating it with the appropriate parameter value is critical to ensure physics-based soil properties are taken into account properly in the simulations of hydrological and geomorphological models.

Problem Identification. When running simulations, it is imperative to use the data that most accurately represent the region of interest, which is typically the finest resolution of data available. In regards to soil data within the United States, SSURGO has the finest resolution; however, certain regions of the SSURGO dataset have missing attributes leading to null textural values and unidentified soil types. This causes a problem when using the soil data for hydrologic simulations as there are no data available for use in the simulation, which can lead to an additional layer of parametric uncertainty in the results. Therefore, to increase accuracy, soil data from another source are needed to replace the areas where the current soil data are unknown or null. The amount of missing data can be very large, and interpolation, such as nearest neighbor, would be inefficient in filling in the missing areas; at times, the missing data are larger than other more coarse soil datasets. Therefore, using another dataset to fill in the missing areas will provide a more complete dataset.

To replace the null textural values with actual textural values, other datasets such as Food and Agriculture Organization (FAO) of the United Nations and STATSGO2 can be combined with the SSURGO dataset, therefore creating a complete dataset (LaHatte and Pradhan 2016). With FAO soil dataset having a resolution at 1 kilometer and STATSGO2 ranging in scale from 1:250,000 to 1:1,000,000, both are coarser than the SSURGO dataset, which has a scale between 1:12,000 and 1:63,360. Therefore, starting with the highest resolution SSURGO dataset and replacing the null values with STATSGO2 first, then FAO, this will allow for the most complete and highest resolution soil texture dataset; this is critical for hydrological modeling as soils affect infiltration, evaporation, erosion, and runoff. The process of replacing the null values can be done through ArcGIS; however, it can be cumbersome and time consuming when repeating this process multiple times. Therefore, developing an automated process in Python to automate the process will save user time and reduce human error.

Soil Resolution Analysis from Different Data Sources. As the resolution of each dataset increases, the number of different soils also increases. STATSGO2 and SSURGO are more similar in comparison to the FAO dataset (Figure 1). Both the STATSGO2 (Figure 1B) and SSURGO (Figure 1C) identify the eastern part of the county as being silt loam with silt loam being the dominant soil type in the study region for SSURGO and STATSGO2. The percentages of silt loam for STATSGO2 and SSURGO are 61.0% and 57.2%, respectively, as shown in Table 1. While the FAO dataset does not recognize silt loam as the major soil texture, it recognizes sandy loam in the same region as shown in Figure 1A. Similarly in Figure 1A, where the FAO shows clay loam, STATSGO2 and SSURGO show clay. In Table 1, the percentages of clay for STATSGO2 and SSURGO are 26.3% and 19.3%, respectively, whereas the percentage of clay loam for FAO is 73.2%. Although there are other non-dominant soil types in SSURGO and STATSGO2, these are less than 12.7% and 9.8%, respectively. From this analysis, STATSGO2 presents a soil type distribution that is consistent with SSURGO data. Therefore, it is adequate to use the STATSGO2 dataset to fill the data gap in the SSURGO dataset.

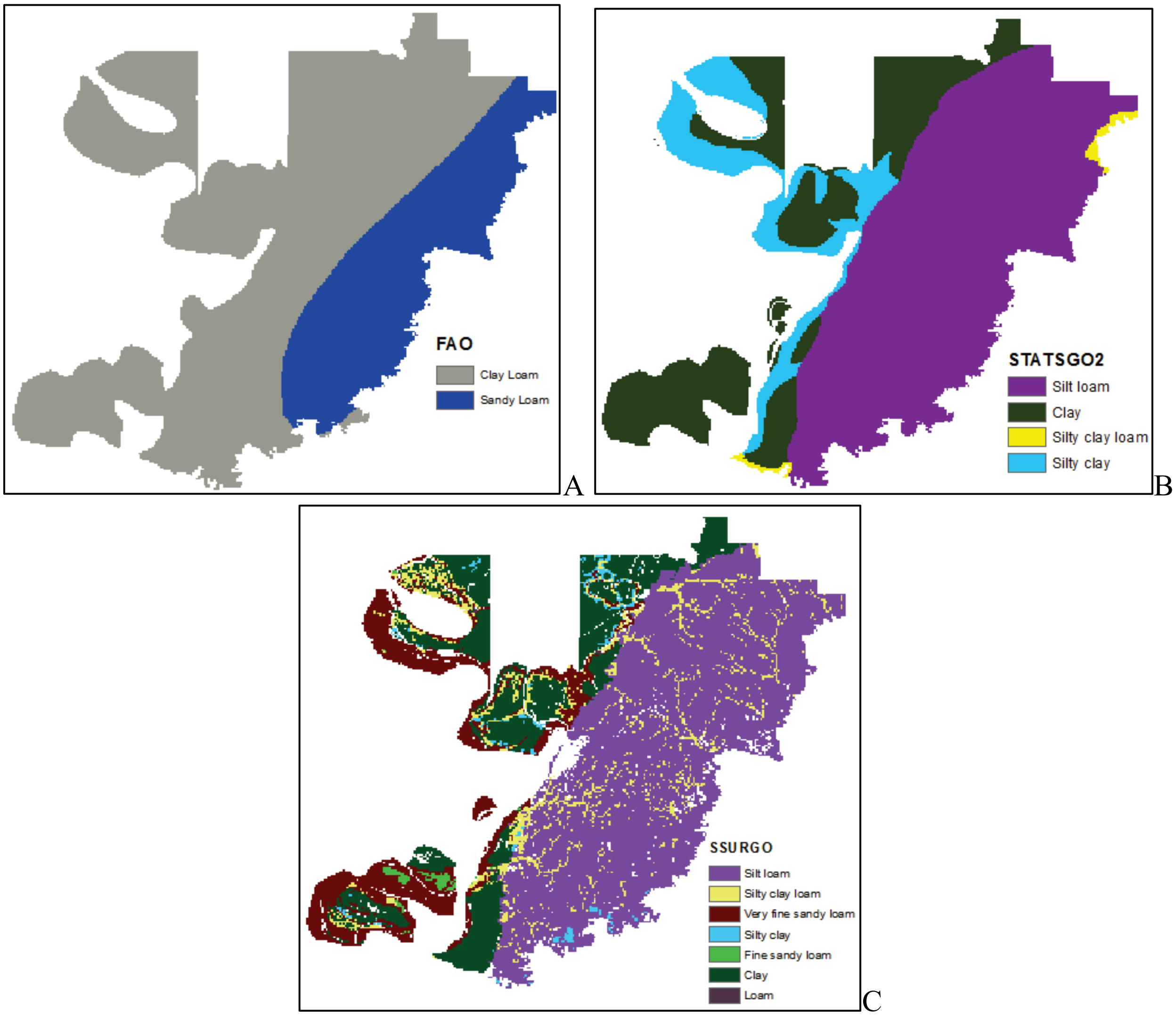


Figure 1. Soil texture classification for Warren County, Mississippi, using (A) FAO, (B) STATSGO2, and (C) SSURGO datasets.

METHODOLOGY: By using a combination of Python and ArcGIS, a holistic dataset of soils can be created. The SSURGO spatial data do not include attributes of the soil's texture classification; therefore, the spatial dataset requires combining with the tabular dataset to create a complete soil textural classification dataset. Once the spatial and tabular datasets have been combined in the SSURGO dataset, an additional dataset can be merged to replace the missing values and can create a complete soils dataset. STATSGO2 and SSURGO are related products therefore making it easier to combine them.

To successfully complete the soils extraction for the United States using SSURGO and STATSGO2 soil data, the following software tools and data should be downloaded and readily available:

Software Requirements

- Environmental Systems Research Institute (ESRI) ArcMap and ArcCatalog (ESRI 2011)
- Microsoft Access (or similar software)
- Python version 2.7 (Python Software Foundation 2010)

SSURGO DATA: The National Cooperative Soil Survey collected soil samples for over a century to obtain information to create the SSURGO database. The information collected can be viewed as a table or map for most of the United States, U.S. Territories, Island Nations, and Commonwealths operated under the United States Department of Agriculture-Natural Resources Conservation Service. The soils information ranges in scale from 1:12,000 to 1: 63,360. The soil data were obtained by walking over land, observing the soils, and in many cases, analyzing it in a laboratory (Soil Survey Staff 2017a). Figure 2 shows the different soil types for Warren County, Mississippi, as described by the SSURGO dataset (Soil Survey Staff 2017b). Each soil is represented by a different color; the zoomed-in portion of the map highlights the high resolution of the soil texture data, including null areas, where no soil information is available.

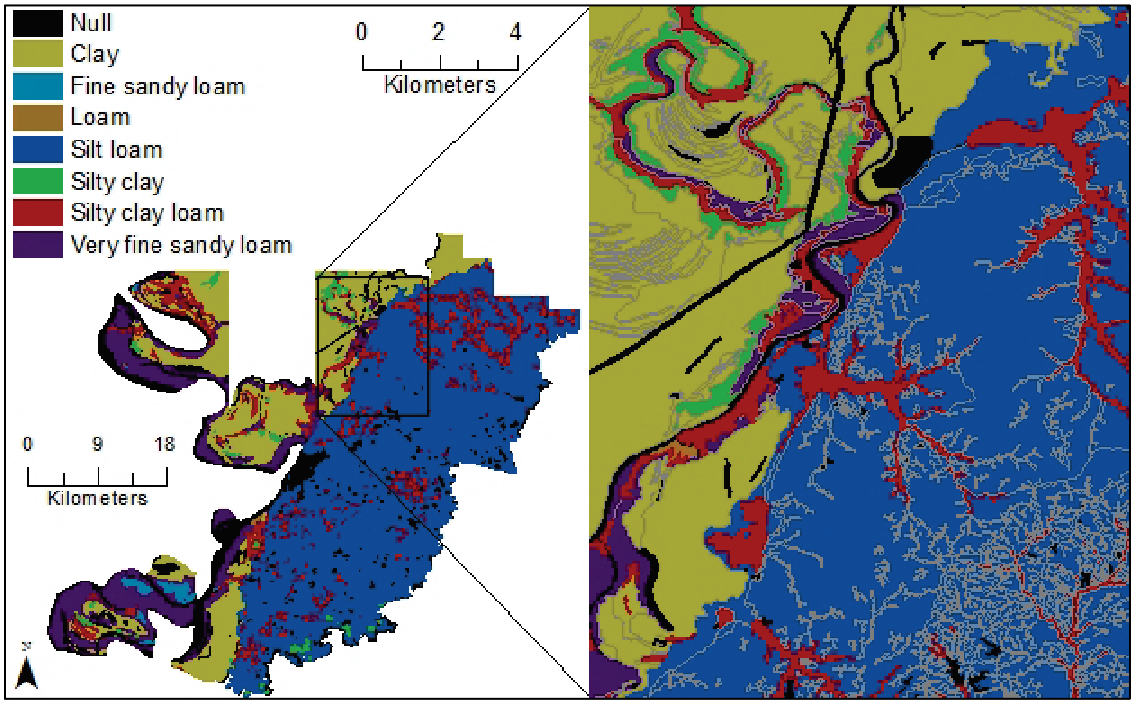


Figure 2. Soils classified by SSURGO for Warren County, Mississippi.

STATSGO2 DATA: Similarly to the SSURGO data, the STATSGO2 dataset was developed by the National Cooperative Soil Survey and is a follow-up product of the STATSGO dataset. The data are available for the continental United States, Hawaii, Virgin Islands, and Puerto Rico at a scale of 1:250,000 and for Alaska at a scale of 1:1,000,000. The map is maintained and disseminated as both a tabular and spatial dataset. The soil data are composed of both soils and non-soil areas that follow a repeatable pattern throughout the landscape (Soil Survey Staff 2017a). Figure 3 shows the different soils textures that occur in Warren County, Mississippi, as described by the STATSGO2 dataset (Soil Survey Staff 2017b).

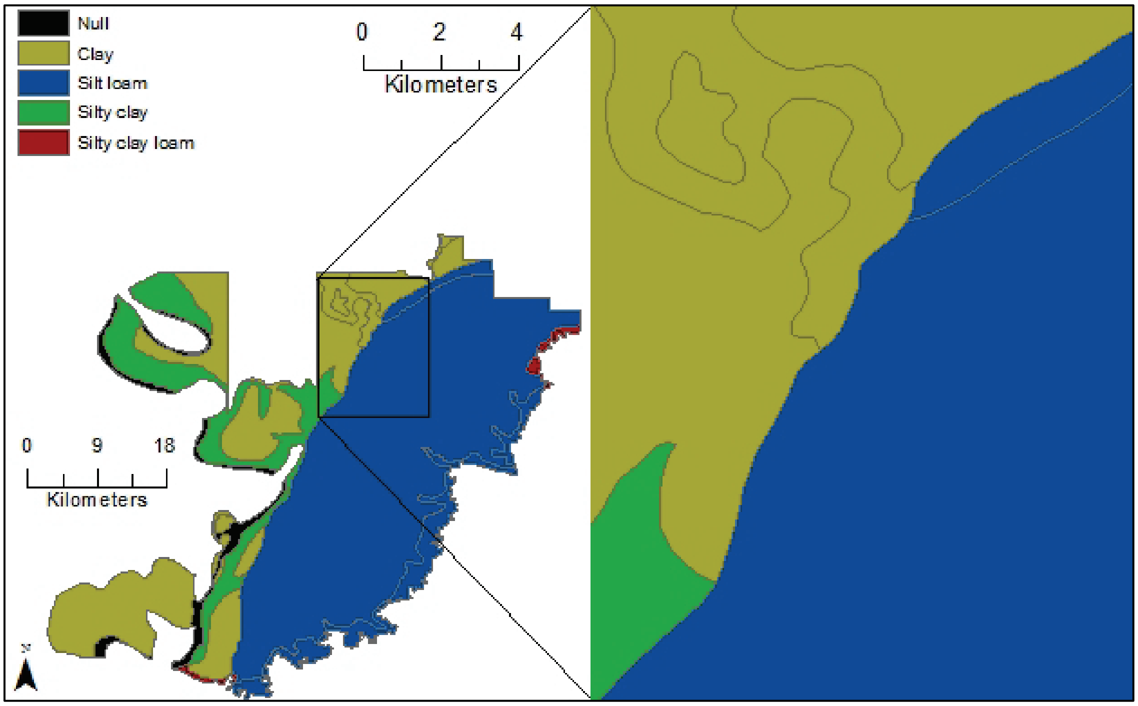


Figure 3. Soils classified by STATSGO2 for Warren County, Mississippi.

Pre-processing for STATSGO2 and SSURGO Data Correlation. These steps prepare the data for usage in Python and ArcMap.

1. Download SSURGO data for the area of interest (county level). This download will include spatial data (shapefiles), tabular data (text files), and a soils template database (Microsoft Access database file) (LaHatte and Pradhan 2016).
   a. Data can be obtained at http://websoilsurvey.sc.egov.usda.gov/App/WebSoilSurvey.aspx by choosing “Download Soils Data” from the top menu.
   b. Select “Soil Survey Area (SSURGO)” from the menu on the left-hand side of the screen and choose your area of interest.
2. Using Microsoft Access, open the soils template database. There is a macro within the template that will be used to import the tabular data. (You may need to click “Enable Content” located on the security bar, if the security settings are active). Enter the entire directory path to the folder that has the tabular data, which will allow the macro to import the tabular data into the database (LaHatte and Pradhan 2016).
3. Download STATSGO2 for the entire United States (downloads are also available separately for each state, the District of Columbia, Puerto Rico, and the Virgin Islands).

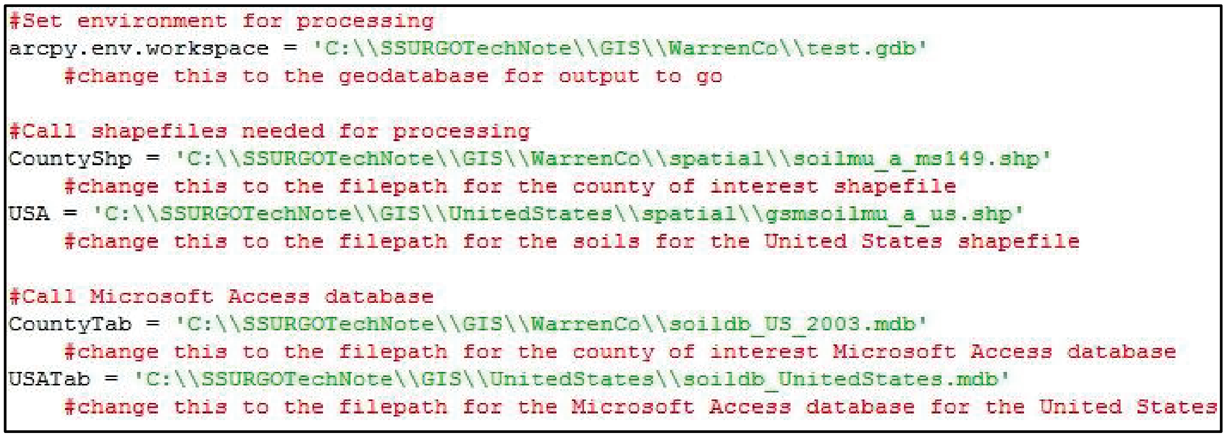
Similarly to SSURGO data, the STATSGO2 dataset includes a spatial component and tabular component.

   a. Data can be obtained by following Step 1a.
   b. Select "STATSGO2 Download Links" from the menu on the left-hand side of the screen and choose "(entire data set)."

4. Repeat Step 2 for the STATSGO2 dataset.
5. In ArcCatalog, create a new file geodatabase by going to the folder where you want to save the newly created files, right clicking and selecting "New" -> "File Geodatabase."
6. Be sure to note where these files are saved as they are needed for future processing.

Python Script Customizing. The following steps outline how to change the corresponding Python script (located in the Appendix) to customize it for your area of interest. (Note: make sure the Python version you are using is compatible with your ArcGIS version.)

1. Create a file geodatabase in ArcCatalog where the output from processing the script will go.
2. Copy and then paste the Python script found in the appendix to an editor for you to customize the code.
3. Refer to the green text in Figure 4–Figure 6 that shows what parts of the script need edited to suit the user's needs.
4. Within the Python script, perform the following:
   a. Change the variable "arcpy.env.workspace" to the file path where the output from the processing will be saved (line 8).
   b. Change the variable "CountyShp" to the file path of where the spatial data (county level shapefile of soil) for your county of interest are located (line 12).
   c. Change the variable "USA" to the file path of where your spatial USA data are located (line 14).
   d. Change the variable "CountyTab" to file path of where your Microsoft Access database for your county of interest is located (line 18).
   e. Change the variable "USATab" to the file path of where your Microsoft Access database for the county of interest is located (line 20).
   f. Change the variable "Join," located in the "for" loop, to the file path of where the file geodatabase you want to save files to is located (line 47).
   g. Change the variable "CountyJoin" to the file path where the newly joined county data is located; this is the same location as Part F, but add "\\TempCountyShp" to end of Part F file path (line 50).
   h. Change the variable "USAJoin" to the file path where the newly joined USA data are located; this is the same location as Part F, but add "\\TempUSAShp" to end of Part F file path (line 52).
   i. Change the first four comparisons in the "for" loop that deletes unnecessary attribute table fields by replacing the "ms149" component of the comparison with the two character state and three character location designation of your county of interest (lines 64, 66, 68, and 70).
5. Save and run the Python script.
6. Check the output to make sure everything ran correctly.

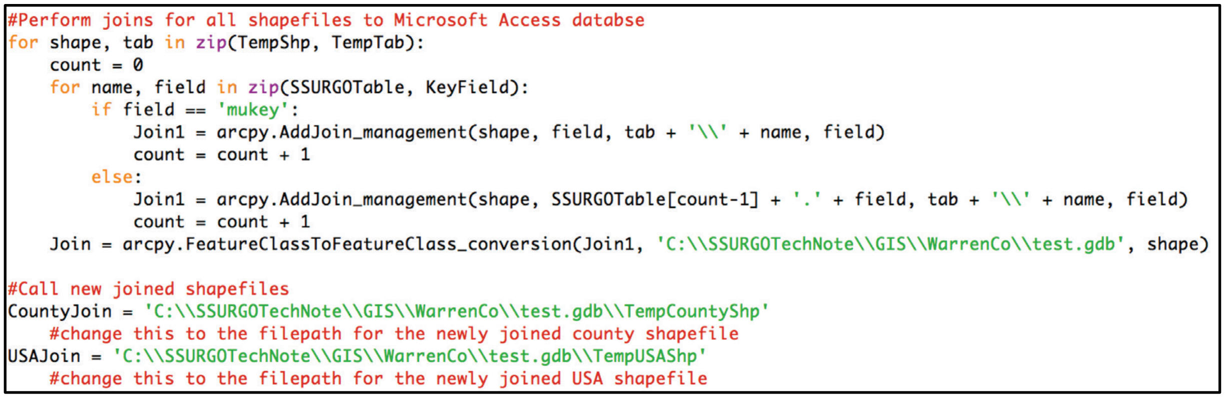


```
#Set environment for processing
arcpy.env.workspace = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\test.gdb'
    #change this to the geodatabase for output to go

#Call shapefiles needed for processing
CountyShp = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\spatial\\soilmu_a_ms149.shp'
    #change this to the filepath for the county of interest shapefile
USA = 'C:\\SSURGOTechNote\\GIS\\UnitedStates\\spatial\\gsmsoilmu_a_us.shp'
    #change this to the filepath for the soils for the United States shapefile

#Call Microsoft Access database
CountyTab = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\soildb_US_2003.mdb'
    #change this to the filepath for the county of interest Microsoft Access database
USATab = 'C:\\SSURGOTechNote\\GIS\\UnitedStates\\soildb_UnitedStates.mdb'
    #change this to the filepath for the Microsoft Access database for the United States
```

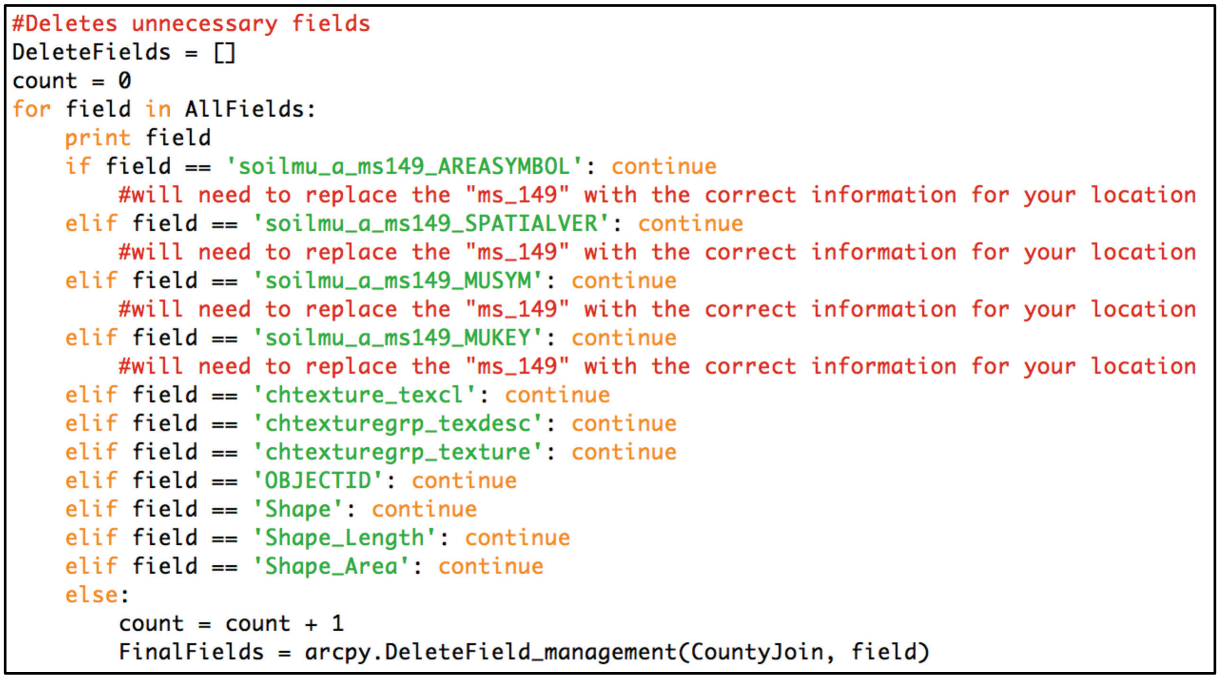
Figure 4. The file paths in green will need to be changed to the correct file paths for the user's data.



```
#Perform joins for all shapefiles to Microsoft Access databse
for shape, tab in zip(TempShp, TempTab):
    count = 0
    for name, field in zip(SSURGOTable, KeyField):
        if field == 'mukey':
            Join1 = arcpy.AddJoin_management(shape, field, tab + '\\' + name, field)
            count = count + 1
        else:
            Join1 = arcpy.AddJoin_management(shape, SSURGOTable[count-1] + '.' + field, tab + '\\' + name, field)
            count = count + 1
    Join = arcpy.FeatureClassToFeatureClass_conversion(Join1, 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\test.gdb', shape)

#Call new joined shapefiles
CountyJoin = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\test.gdb\\TempCountyShp'
    #change this to the filepath for the newly joined county shapefile
USAJoin = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\test.gdb\\TempUSAShp'
    #change this to the filepath for the newly joined USA shapefile
```

Figure 5. Similar to Figure 3, the file paths that need to be changed for the user's data are in green text.



```
#Deletes unnecessary fields
DeleteFields = []
count = 0
for field in AllFields:
    print field
    if field == 'soilmu_a_ms149_AREASYMBOL': continue
        #will need to replace the "ms_149" with the correct information for your location
    elif field == 'soilmu_a_ms149_SPATIALVER': continue
        #will need to replace the "ms_149" with the correct information for your location
    elif field == 'soilmu_a_ms149_MUSYM': continue
        #will need to replace the "ms_149" with the correct information for your location
    elif field == 'soilmu_a_ms149_MUKEY': continue
        #will need to replace the "ms_149" with the correct information for your location
    elif field == 'chtexture_texcl': continue
    elif field == 'chtexturegrp_texdesc': continue
    elif field == 'chtexturegrp_texture': continue
    elif field == 'OBJECTID': continue
    elif field == 'Shape': continue
    elif field == 'Shape_Length': continue
    elif field == 'Shape_Area': continue
    else:
        count = count + 1
        FinalFields = arcpy.DeleteField_management(CountyJoin, field)
```

Figure 6. The green text of "ms149" needs to be changed to match the user's data. The first two characters represent the state (e.g., ms = Mississippi), and the last three characters represent the area of the state (e.g., 149 = Warren County).

RESULTS AND DISCUSSION: As shown by Figure 7, SSURGO and STATSGO2 soil texture data in Warren County were combined with the null SSURGO soil texture values replaced by STATSGO2 soil textures. This allows for a holistic soil texture map, with the sharpest resolution possible by using SSURGO soil texture first and replacing the missing values with STATSGO2 soil texture. This is also shown when comparing Figure 2 and Figure 3. Figure 2 has more areas of different color compared to Figure 3, indicating that it has a higher resolution of soil texture characteristics.

Figure 8 compares the difference between (A) the SSURGO soil texture dataset with null values and (B) the SSURGO soil texture dataset with the null values replaced with the STATSGO2 dataset. The null area noted with black coloring is apparent in Figure 8A but is no longer there in Figure 8B due to being replaced with the STATSGO2 dataset. While the majority of the United States has SSURGO soil texture data available, it is important to replace the missing data with data from another source allowing for a more complete dataset.

A combined map of SSURGO and STATSGO2 soil texture is shown in Figure 9. The null values from the SSURGO dataset have been replaced with soil texture data from the STATSGO2 dataset. This allows for the finest resolution and most complete dataset available.

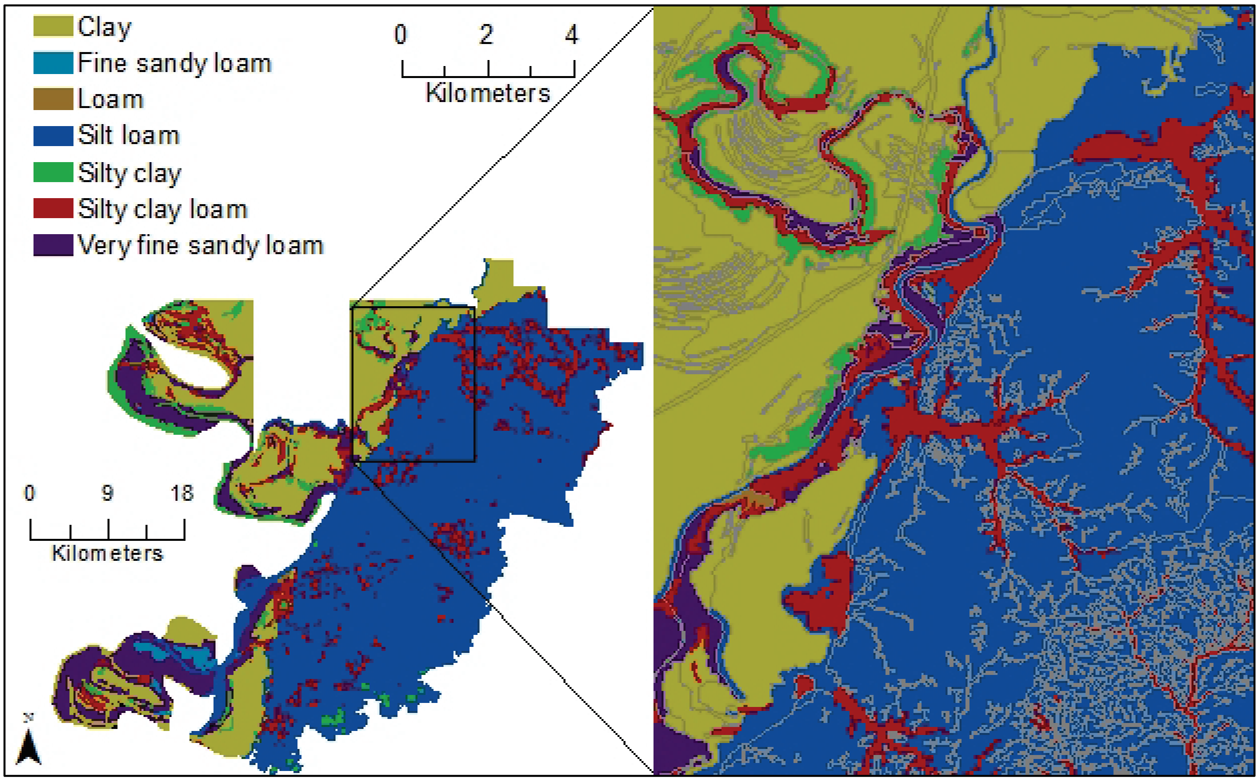


Figure 7. The STATSGO2 soil textures have replaced the null values from the SSURGO soil textures creating a complete soil texture dataset for Warren County, Mississippi.

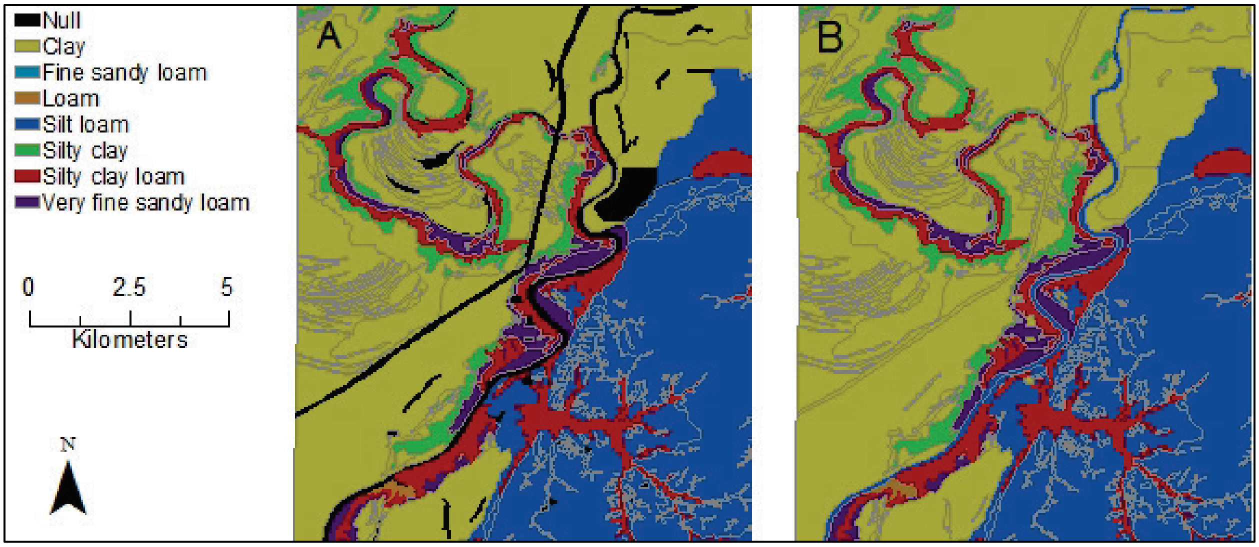


Figure 8. (A) SSURGO soil textures with the null values shown in black; (B) A combination of SSURGO and STATSGO2 soil textures with the null values from the SSURGO soil texture dataset replaced with STATSGO2 soil textures.

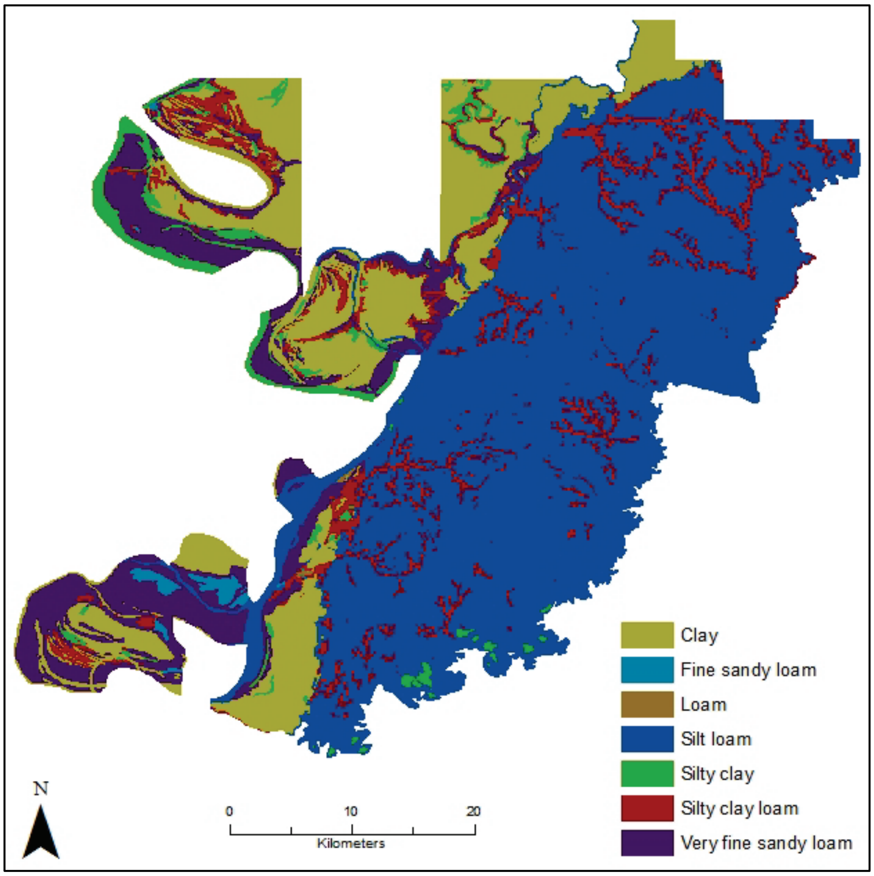


Figure 9. A soil texture map for Warren County, Mississippi, using SSURGO soil textures and replacing the null values with STATSGO2 soil textures.

CONCLUSIONS: The methods outlined in this CHETN can be utilized to better populate and refine soil information for the United States. While a similar approach may be completed in ArcGIS (LaHatte and Pradhan 2016), utilizing the Python code discussed herein (and presented in the appendix) saves time, especially when performed multiple times, and reduces human error. By replacing the null values in the SSURGO dataset with data from STATSGO2, this creates a holistic dataset of soil textures, vastly reducing the amount of null values and unknown soil textures. Additionally, starting with the finest soil textures (SSURGO) and replacing the null values with the second finest soils textures (STATSGO2) creates a soil dataset of the higher precision, therefore yielding more accurate spatially distributed simulation results when employed in a physical process-based hydrological or geomorphological model. The illustrated Python code in this CHETN is in the simplest form that can be easily understood and deployed/enhanced.

ADDITIONAL INFORMATION: For additional information, contact Ms. Kayla Cotterman, Coastal and Hydraulics Laboratory, U.S. Army Research and Development Center, 3909 Halls Ferry Road, Vicksburg, MS 39180, at 601-634-5507 or e-mail: *Kayla.A.Cotterman@erdc.dren.mil*.

This work was funded by Austere A/SPOD Entry Assessment Military Engineering work package under work item number 5F09BK. This CHETN should be cited as follows:

> Cotterman, K. A., and N. R. Pradhan. 2017. *Development of an Automatic Soil Texture Extraction Routine Using Combined SSURGO and STATSGO2 Dataset.* ERDC/CHL CHETN-X-4. Vicksburg, MS: U.S. Army Engineer Research and Development Center. *http://dx.doi.org/10.21079/11681/25794*

REFERENCES

ESRI. 2011. ArcGIS Desktop: Release 10. Redlands, CA: Environmental Systems Research Institute.

LaHatte, W. C., and N. R. Pradhan. 2016. *Analysis of SURRGO Data and Obtaining Soil Texture Classifications for Simulating Hydrologic Processes*. ERDC/CHL CHETN-X-3. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

Python Software Foundation. 2010. *The Python Language Reference*, v2.7. Accessed 7 September. *https://docs.python.org/release/2.7/reference/index.html*.

Soil Survey Staff, Natural Resources Conservation Service, United States Department of Agriculture. 2017a. Official Soil Series Descriptions. Accessed 7 September. *https://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/home/?cid=nrcs142p2_053587*.

Soil Survey Staff, Natural Resources Conservation Service, United States Department of Agriculture. 2017b. Web Soil Survey. Accessed 7 September. *https://websoilsurvey.nrcs.usda.gov/*.

APPENDIX

```
#This script is for auto soil processing for the continental US

#import tools for processing
import arcpy
from arcpy import env

#Set environment for processing
arcpy.env.workspace = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\test.gdb'
   #change this to the geodatabase for output to go

#Call shapefiles needed for processing
CountyShp = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\spatial\\soilmu_a_ms149.shp'
```

```
    #change this to the filepath for the county of interest shapefile
USA = 'C:\\SSURGOTechNote\\GIS\\UnitedStates\\spatial\\gsmsoilmu_a_us.shp'
    #change this to the filepath for the soils for the United States shapefile

#Call Microsoft Access database
CountyTab = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\soildb_US_2003.mdb'
    #change this to the filepath for the county of interest Microsoft Access database
USATab = 'C:\\SSURGOTechNote\\GIS\\UnitedStates\\soildb_UnitedStates.mdb'
    #change this to the filepath for the Microsoft Access database for the United States

#List of SSURGO table
SSURGOTable = ['component', 'chorizon', 'chtexturegrp', 'chtexture']

#List of key fields
KeyField = ['mukey', 'cokey', 'chkey', 'chtgkey']

#Make temporary feature layer to allow joining of tables to each other
TempCountyShp = arcpy.MakeFeatureLayer_management(CountyShp, "TempCountyShp")
TempUSAShp = arcpy.MakeFeatureLayer_management(USA, "TempUSAShp")

#Make a list of shapefiles and Microsoft Access database that need joins
TempShp = [TempCountyShp, TempUSAShp]
TempTab = [CountyTab, USATab]

#Perform joins for all shapefiles to Microsoft Access databse
for shape, tab in zip(TempShp, TempTab):
    count = 0
    for name, field in zip(SSURGOTable, KeyField):
        if field == 'mukey':
            Join1 = arcpy.AddJoin_management(shape, field, tab + '\\' + name, field)
            count = count + 1
        else:
            Join1 = arcpy.AddJoin_management(shape, SSURGOTable[count-1] + '.' + field, tab + '\\' + name, field)
            count = count + 1
    Join  =  arcpy.FeatureClassToFeatureClass_conversion(Join1,  'C:\\SSURGOTechNote\\GIS\\WarrenCo\\test.gdb',
                 shape)

#Call new joined shapefiles
CountyJoin = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\test.gdb\\TempCountyShp'
    #change this to the filepath for the newly joined county shapefile
USAJoin = 'C:\\SSURGOTechNote\\GIS\\WarrenCo\\test.gdb\\TempUSAShp'
    #change this to the filepath for the newly joined USA shapefile

#Preps and deletes fields for counties
#Creates a list for all fields in the shapefile
AllFields = [f.name for f in arcpy.ListFields(CountyJoin)]

#Deletes unnecessary fields
DeleteFields = []
count = 0
for field in AllFields:
    print field
    if field == 'soilmu_a_ms149_AREASYMBOL': continue
        #will need to replace the "ms_149" with the correct information for your location
    elif field == 'soilmu_a_ms149_SPATIALVER': continue
        #will need to replace the "ms_149" with the correct information for your location
    elif field == 'soilmu_a_ms149_MUSYM': continue
```

```
        #will need to replace the "ms_149" with the correct information for your location
    elif field == 'soilmu_a_ms149_MUKEY': continue
        #will need to replace the "ms_149" with the correct information for your location
    elif field == 'chtexture_texcl': continue
    elif field == 'chtexturegrp_texdesc': continue
    elif field == 'chtexturegrp_texture': continue
    elif field == 'OBJECTID': continue
    elif field == 'Shape': continue
    elif field == 'Shape_Length': continue
    elif field == 'Shape_Area': continue
    else:
        count = count + 1
        FinalFields = arcpy.DeleteField_management(CountyJoin, field)

#Preps and deletes fields for United States
#Creates a list for all fields in the shapefile
AllFields = [f.name for f in arcpy.ListFields(USAJoin)]

#Deletes unnecessary fields
DeleteFields = []
count = 0
for field in AllFields:
    print field
    if field == 'soilmu_a_us_AREASYMBOL': continue
    elif field == 'soilmu_a_us_SPATIALVER': continue
    elif field == 'soilmu_a_us_MUSYM': continue
    elif field == 'soilmu_a_us_MUKEY': continue
    elif field == 'chtexture_texcl': continue
    elif field == 'chtexturegrp_texdesc': continue
    elif field == 'chtexturegrp_texture': continue
    elif field == 'OBJECTID': continue
    elif field == 'Shape': continue
    elif field == 'Shape_Length': continue
    elif field == 'Shape_Area': continue
    else:
        count = count + 1
        FinalFields = arcpy.DeleteField_management(USAJoin, field)

#Join USA and county together
CoUSAJoin = arcpy.SpatialJoin_analysis(CountyJoin, USAJoin, "CoUSAJoin")

#Add column to attribute table for a combination of USA and Co level data
arcpy.AddField_management(CoUSAJoin, "ComboCoUSA", "TEXT")

#Combine replace Null county data with USA data
arcpy.CalculateField_management(CoUSAJoin, "ComboCoUSA", expression="replace(!chtexturegrp_texdesc!,
                !chtexturegrp_texdesc_1!)", expression_type="PYTHON_9.3",
                        code_block="def replace(Co, USA):\n  if Co is None:\n    return USA\n  else:\n    return Co")
```

***NOTE:** The* contents *of this technical note are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such products.*