

APPENDIX A: DETAILED INSTRUCTIONS FOR DATA PRE-PROCESSING AND PARAMETER ESTIMATION

This appendix provides detailed instructions for developing the forcing data, model structure, and parameters necessary to implement the modeling methods as described in the main body of this report. The modeling methods were developed for basins in the forested and mountainous Colorado Front Range in this study. The methods incorporate principles that apply to a wide range of land covers, soil types, and geologic conditions, but their applicability to other regions needs to be assessed empirically. Caution should especially be used when applying the methods to regions where the hydrologic response may be affected by snowmelt, frozen ground, wildfire, and urban development as these factors were not considered in this study.

When applying the methods to historical storms, rainfall forcing data must be processed to adapt it to the format required for input to HEC-HMS. Significant pre-processing is required to convert rainfall data from hourly depth grids as output from SPAS to the hyetograph format that can be directly input into HEC-HMS.

The hydrologic model discussed in this study is implemented with a semi-distributed sub-basin structure. Disaggregation of the basin into sub-basins is completed to manage spatial variability of rainfall data and watershed properties. The main body of this document discusses the selection of the methods used to adequately represent the hydrologic processes occurring within each sub-basin. Section 3.2.2 Process Representation of the main document outlines the methods used for each hydrologic process. These methods can be selected in HEC-HMS under the “Subbasin” tab. Parameters must be estimated for Canopy Interception, Loss, Transform, Baseflow, and Routing processes. Section 3.3 Model Parameters outlines the parameters required for the selected methods for each process developed in this study.

Some parameters that are required within HEC-HMS for the methods used in this study are excluded in order to simplify the modeling process. Parameters could be excluded because they are unnecessary for single event simulations (i.e. Loss: GW 2 Max Storage, GW 2 Max Percolation, GW 2 Storage Coefficient) or they have minimal impact during extreme events such as those that this study focuses on (i.e. Canopy: Initial Storage; Loss: Initial GW 1 and Initial GW 2 Storage; and Baseflow: GW 1 Initial Storage).

Forcing Data

1. Rainfall:
 - Instructions in this section are for historical storms in which hourly rainfall total grids were in ASCII format were developed using SPAS
 - The CO-NM Regional Extreme Precipitation Study tool should be used in developing rainfall data for design storms instead of the procedure shown here for historical storms
 - Use ModelBuilder model to project ASCII rasters to UTM Zone 13N and resize to 5-10 m resolution
 - Direct path of “ASCII” bubble to folder which contains ASCII rasters
 - Direct path of %Name%_proj.tif bubble to desired location of output rasters
 - ModelBuilder model: in data repository
 - Use Python code provided in Appendix B to calculate hourly spatial average and create hyetographs for each sub-basin: in data repository
 - The Python code for this process uses both Pandas and ArcPy libraries. Installing the Pandas package with the IDLE Python GUI can be difficult. For this reason, it can be simplest to split the code into two separate model runs if unfamiliar with Python. The first run uses ArcPy portion of the code and is run in IDLE; the second uses Pandas to

- plot the hyetographs and can be executed in a distribution such as Enthought's Canopy. It is possible; however, to use Pandas with the IDLE GUI to run the entire code at once.
 - Required data and variables are described in the Python file
- The Python script will output hyetographs to Excel and save the file in the location that the user specifies
 - Excel hyetographs are in incremental cm and need to be converted to mm or in for HEC-HMS input

Model Structure

2. Sub-basin Disaggregation

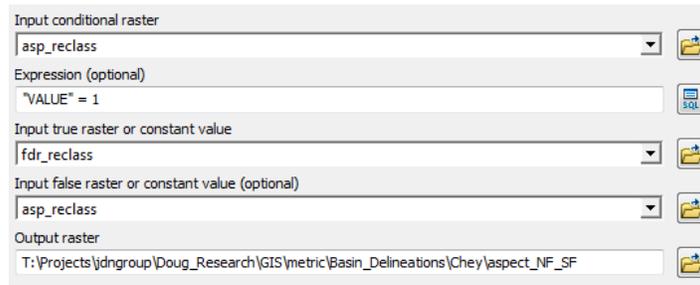
- Digital Elevation Model (DEM)
 - When developing the modeling methods, DEMs for all basins were downloaded from the National Elevation Dataset (NED) to ensure consistent data quality for all hydrologic models. NED data covers the entire Front Range, and the tiles are 1/3 arc-second resolution.
 - Reproject the DEM into the NAD83 UTM projection using the Project Raster tool
 - The entirety of the eastern plains and front range as well as much of the western slope lies within Zone 13N; only a portion of far western Colorado lies in zone 14
 - Do not change the Output Cell Size and choose Bilinear for the Resampling Technique
 - Convert the DEM elevation values from feet to meters with the Raster Calculator tool if it is not already in meters.
- Use ArcHydro tools in ArcGIS to develop sub-basins (refer to Djokic et al (2011) for overview)
 - Do not use DEM reconditioning to burn stream lines into DEM prior to disaggregation
 - Fill sinks/pits in DEM to avoid undefined flow directions
 - Run Flow Direction tool to determine flow direction of each cell based on neighboring elevations
 - Run Flow Accumulation tool to determine the number of cells flowing into each cell
 - Run Stream Definition tool
 - Creates stream grid that indicates the presence/absence of a stream in each cell based on a contributing area threshold
 - Threshold of 15 km² was selected as consistent threshold for all basins to adequately characterize spatial variation of precipitation
 - Run Stream Segmentation tool to separate stream grid into stream segments where confluences distinguish the ends of each segment
 - Run Catchment Grid Delineation tool to delineate a separate sub-basin for each stream segment
 - Run Catchment Polygon Processing tool to create sub-basin shapefiles from grid
 - Because stream gauge location likely will not coincide with outlet of any sub-basin, use Watershed tool to delineate basin at gauge location and incorporate the basin into the sub-basin structure
- Separate sub-basins into north- and south-facing aspects
 - Run Aspect tool with the DEM as input to calculate slope direction in degrees
 - Run Reclassify tool on Aspect raster to specify that North includes aspects from 0°-90° and 270°-360° while South includes aspects from 90°-270° (In example below, North=2 and South=3)

Old values	New values
-1 - 0	1
0 - 90	2
90 - 270	3
270 - 360	2
NoData	NoData

- Because some cells will have no aspect (i.e. flat), use flow direction raster to assign aspect to these cells
 - Use Reclassify tool on flow direction raster to assign north-facing values to flow direction cells equal to 32, 64, and 128 and south-facing values to flow directions of 2, 4, and 8 (refer to ESRI's online help for an overview of the flow direction raster)
 - For flow direction cells exactly east or west (value=16 or 1), assign one direction to be north and the other to be south (in example below, east is assigned to be north and west is assigned to be south)
 - These can be assigned arbitrarily because they comprise a small portion of the basin and will have minimal impact on the delineation

Old values	New values
1	2
32	2
64	2
128	2
NoData	NoData
2 - 16	3

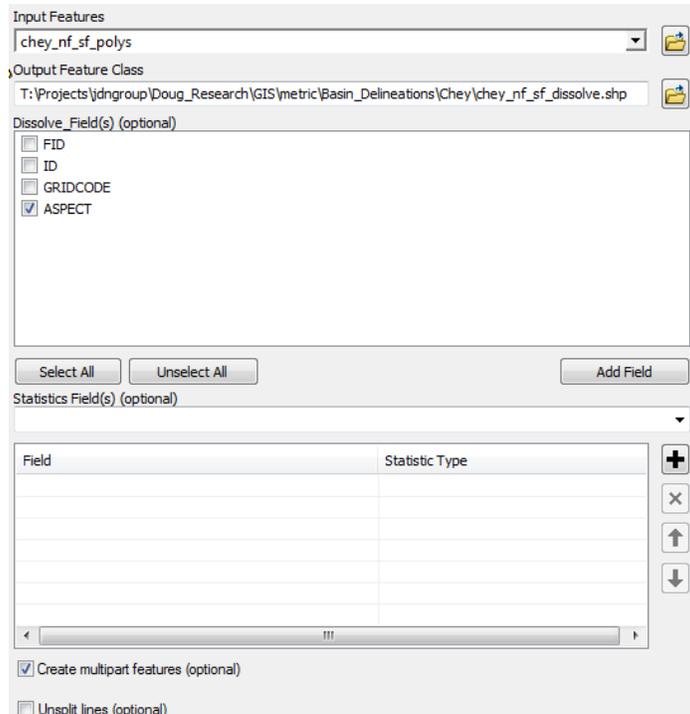
- Assign North and South from flow direction raster to reclassified aspect raster using the Con tool



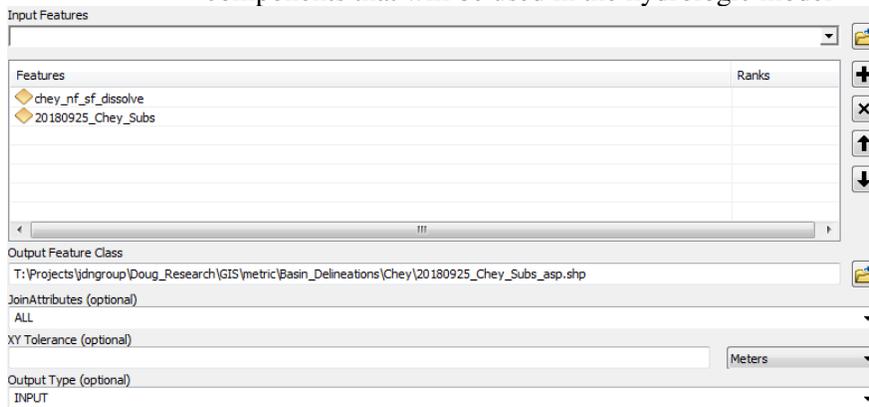
- Add 'Aspect' field raster to define North and South facing cells
 - Consistent with the naming convention from the previous example, in the image below "Value=2" is assigned to be North and "Value=3" is South

Rowid	VALUE	COUNT	ASPECT
0	2	367794	North
1	3	304079	South

- Convert the raster to a shapefile using the Raster to Polygon tool with 'Aspect' as the new value field (do not check the simplify polygons box).
- Use NF/SF shapefile to separate the sub-basin shapefile by aspect
 - Using Dissolve tool, dissolve discontinuous features of NFS/SFS aspect shapefile into two single features (one for NFS, one for SFS)
 - Specify newly created "Aspect" field as the Dissolve Field as shown in image below



- Use Intersect tool with dissolved NFS/SFS shapefile to find north- and south-facing component of each sub-basin
 - The output of this tool is the final sub-basin shapefile with NFS and SFS components that will be used in the hydrologic model



- Create basin model in HEC-HMS with NFS and SFS sub-basins corresponding to the shapefile produced in the previous step
 - Sub-basin SFS and NFS elements will be comprised of noncontiguous areas, but they can be lumped together and treated as a single sub-basin because unit hydrograph theory assumes additivity
 - Area can be obtained from shapefile and input into HEC-HMS

Parameters

3. Canopy Interception

The instructions provided in this section assist with estimation of the *maximum canopy storage* and *potential evapotranspiration rate* parameters under the Simple Canopy. These parameters can be input into HEC-HMS under the Canopy tab.

- Based on rainfall data measured at a research catchment in the Cache la Poudre River basin (Traff et al., 2015) during September 2013 storms, the Simple Canopy model was implemented in a spreadsheet to calibrate *maximum canopy storage* and *potential evapotranspiration rate* while optimizing the NSCE in comparison with the observed throughfall data. The analysis was completed for both NFS and SFS.
- The optimized canopy capacity is 7.2 mm on NFS and 1.5 on SFS. These values can be entered as the *maximum canopy storage* under the Simple Canopy method
- The optimized depletion rate is 2.3 mm/day on NFS and 5.6 on SFS. These are applied by selecting annual evapotranspiration in the meteorologic model. Enter the values in the “Rate” column.

4. Loss and Baseflow

The instructions provided in this section assist with estimation of the *initial soil storage*, *maximum infiltration rate*, *maximum soil storage*, *tension storage*, *maximum soil percolation rate*, *maximum GW 1 storage*, *maximum GW 1 percolation*, and *GW 1 storage coefficient* parameters under the Soil Moisture Accounting method and the *linear reservoir GW 1 storage coefficient* under the Linear Reservoir method. These parameters can be input into HEC-HMS under the Loss and Baseflow tabs.

The soil layer parameters were estimated primarily using the Natural Resources Conservation Service’s (NRCS) Soil Survey Geographic (SSURGO) database. The soil texture data is used as an input in the Python script `pedotransfer_fn.py` (Appendix C) to perform a number of calculations to obtain SMA parameters. The script first uses pedotransfer functions (Saxton and Rawls, 2006; Rawls et al., 1983) to calculate the bare ground saturated hydraulic conductivity K_{sat} , porosity ϕ , field capacity θ_{fld} , wilting point θ_{wfp} , and wetting front suction head ψ_f . K_{sat} is then adjusted for vegetation cover based on existing hydrology guidelines for Colorado (Sabol, 2008). The script then uses the Green-Ampt equation to calculate the maximum infiltration rate at a representative depth. Finally, the maximum and initial soil storage are calculated.

Use this portion of the document in conjunction with the Python script ‘`pedotransfer_fn.py`’ in Appendix C. Variable names below in **bold** denote files that must be created or updated by the user. The remaining variables in *italics* are created internally by the model. The user must update all file paths in the script for input, internal, and output variables. ‘`pedotransfer_fn.py`’ is used to calculate *maximum infiltration rate*, *maximum soil storage*, and *tension storage*. The Python script outputs these parameters as spatial average values for each sub-basin in table format.

- Sub-basin shapefile
 - Variable name in ‘`pedotransfer_fn.py`’: ***subs***
 - Sub-basin shapefile (without NFS and SFS elements) as developed previously in Section 1. Sub-basin Disaggregation
- NFS and SFS polygons
 - Variable name in ‘`pedotransfer_fn.py`’: ***aspect***
 - NFS and SFS polygons as developed previously in Section 1. Sub-basin Disaggregation
- Sub-basin shapefile with NFS and SFS elements
 - Variable name in ‘`pedotransfer_fn.py`’: ***asp_subs***
 - Sub-basin shapefile (separated into NFS and SFS elements) as developed previously in Section 1. Sub-basin Disaggregation
- Soil textures and organic matter
 - Variable names in ‘`pedotransfer_fn.py`’: ***sand***, ***clay***, and ***OM***

- Rasters representing the percent sand, percent clay, and percent organic matter of the soil. The Python script uses the soil texture data as an input to pedotransfer functions (Saxton and Rawls, 2006; Rawls et al., 1983) to calculate the bare ground saturated hydraulic conductivity K_{sat} , porosity ϕ , field capacity θ_{fld} , wilting point θ_{wp} , and wetting front suction head ψ_f .
- Steps to download rasters:
 - Extract dominant percent sand, clay, and organic matter from SSURGO database
 - Download soils data from NRCS (<https://gdg.sc.egov.usda.gov/>)
 - Import map unit raster or polygon from soils data into ArcGIS
 - Run Create Soil Map tool in Soil Data Development Toolbox for percent sand, clay, and organic matter
 - Use Weighted Average Aggregation Method and top 45.7 cm of soil—or 18 inches per Colorado’s existing guidelines for storms less frequent than the 100-yr event (Sabol, 2008)
 - Convert shapefiles to rasters, set cell size between 5-10 m (make sure cell size is consistent for all rasters) and project to UTM Zone 13N
- Theta parameters
 - Variable names in ‘pedotransfer_fn.py’: θ_{33t} , θ_{33} , θ_{s33t} , θ_{s33} , θ_s
 - This series of calculations is done to determine the θ_s parameter (porosity) using Equations 2-3 and 5 from Saxton and Rawls (2006). Additionally, θ_{33} represents the field capacity soil moisture
- Bare ground hydraulic conductivity
 - Variable names in ‘pedotransfer_fn.py’: B , θ_{1500t} , θ_{1500} , λ , $K_{satbare}$
 - This series of calculations uses Equations 1, 15, 16, and 18 from Saxton and Rawls (2006) to calculate the bare ground hydraulic conductivity ($K_{satbare}$). Additionally, θ_{1500} represents the wilting point soil moisture
- Wetting front suction head
 - Variable names in ‘pedotransfer_fn.py’: A , ψ_{iet} , ψ_{ie} , ψ_{iehead} , ψ_{if}
 - This series of calculations uses Equations 4 and 14 from Saxton and Rawls (2006) and Equation 4 from Rawls et. al (1983) to calculate wetting front suction head (ψ_{if})
- Vegetation cover (or fractional vegetation)
 - Variable name in ‘pedotransfer_fn.py’: V_c
 - Fractional vegetation cover is used to adjust the bare ground K_{sat} for vegetation cover based on existing hydrology guidelines for Colorado (Sabol, 2008)
 - Request NDVI data for area of interest from Earth Science Processing Architecture (ESPA) (<https://espa.cr.usgs.gov/>)
 - NDVI data has scale factor of 0.0001; use Raster Calculator to convert to unscaled NDVI
 - Use Raster Calculator to calculate fractional vegetation per Montandon and Small (2007):

$$F_g = \left(\frac{NDVI_{pixel} - NDVI_0}{NDVI_{\infty} - NDVI_0} \right)^2$$
 - $NDVI_{pixel}$ is the raster that was obtained from ESPA; $NDVI_0$ is the bare soil NDVI which can be obtained by determining value of the ESPA NDVI raster in bare soil locations; $NDVI_{\infty}$ is the live vegetation NDVI which can be obtained by determining value of the ESPA NDVI raster in densely forested locations
- Saturated hydraulic conductivity (adjusted for vegetation)
 - Variable names in ‘pedotransfer_fn.py’: ck , $K_{sathalf}$, K_{sat}

- The equation from Figure 6.8 of Sabol (2008) is used to adjust the saturated hydraulic conductivity based on vegetation cover. Per Bouwer (1964), hydraulic conductivity should be adjusted to ½ of the saturated hydraulic conductivity value to account for flow in the unsaturated zone.
- Maximum infiltration
 - Variable name in 'pedotransfer_fn.py': *delta, f, f_halfks*
 - Green-Ampt equation is used to calculate the maximum infiltration rate (*f*) at a representative depth, *δ* (*delta*)
 - *δ* assumed to be 3 inches

$$f = K_{sat} \left[1 + \frac{|\psi_f|}{\delta} \right]$$

K_{sat}: saturated hydraulic conductivity (*Ksathalf*)

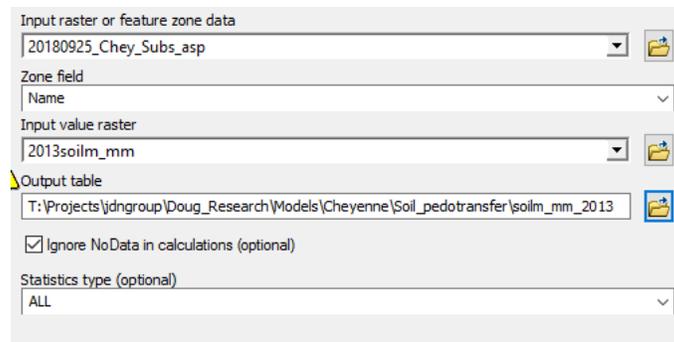
ψ_f: wetting front suction head (*psi_f*)

φ: porosity (*theta_s*)

- The full saturated hydraulic conductivity (*K_{sat}*) and half saturated hydraulic conductivity (*Ksathalf*) were both used in the equation to compare results, and it was determined that using *Ksathalf* produces more realistic results.
- *f* corresponds to *maximum infiltration rate* and can be input in the HEC-HMS model under the Loss tab
- Depth to restrictive layer
 - Variable name in 'pedotransfer_fn.py': ***d_restr***
 - Raster of soil depth to a restrictive layer
 - Steps to develop raster:
 - Use Soil Data Viewer to extract depth to restrictive layer
 - Due to an error in NRCS's aggregation method with GSSURGO data, Soil Data Development Toolbox should not currently be used for depth to restrictive layer
 - Download soils data from Web Soil Survey
 - Add soil layer shapefile which defines soil polygon boundaries into ArcMap (shapefile name usually starts with soilmu_a)
 - Import tabular data into database
 - Open .mdb database file that was downloaded from Web Soil Survey
 - Follow "SSURGO Import" instructions that prompt user to enter path of tabular data folder
 - Run Soil Data Viewer by clicking on its icon on the toolbar
 - Select soil layer that was previously added to map
 - Select database that contains tabular soil data (this is the .mdb file downloaded from Web Soil Survey)
 - Select the Depth to Any Soil Restrictive Layer tool in Soil Data Viewer and select the "Map" option
 - Convert output shapefile to raster, set cell size consistent with soil textures and organic matter rasters, and project to UTM Zone 13N
- Maximum Soil Storage and Wilting Point Storage
 - Variable names in 'pedotransfer_fn.py': *S_max, Si_fld, Si_wp*
 - Maximum soil storage (*S_max*), field capacity soil storage (*Si_fld*), and wilting point soil storage (*Si_wp*) are calculated by multiplying soil depth by the corresponding soil moisture content.
 - *S_max* corresponds to *maximum soil storage* and *Si_wp* corresponds to *tension storage*, and both parameters are input into HEC-HMS under the Loss tab

The remaining parameters are not calculated using the pedotransfer_fn.py file. The procedures used to estimate *initial soil storage*, *maximum soil percolation rate*, *maximum GW 1 storage*, *maximum GW 1 percolation*, *GW 1 storage coefficient* and *linear reservoir GW 1 storage coefficient* are described below.

- *Initial soil storage*
 - Download NLDAS soil moisture data for immediately prior to storm (https://disc.gsfc.nasa.gov/datasets/NLDAS_MOS0125_H_V002/summary?keywords=NLDAS)
 - Select Subset / Get Data option and enter date and location
 - File Format: NetCDF
 - Variables: SOILM = Soil moisture content (kg/m²)
 - Note: units convert to 1 mm of water ($\frac{kg}{m^2} \times \frac{1000\ cm^3}{m^2} \times \frac{1\ m^3}{10^6\ cm^3} = 1\ mm$)
 - Import soil moisture NetCDF data into ArcGIS
 - Use “Make NetCDF Raster Layer” tool in ArcGIS to import soil moisture data
 - Export resulting layer to a raster that covers basin area
 - Calculate spatial average of soil moisture data by sub-basin (separated by aspect) using the Zonal Statistics as Table tool:



- Output is the soil moisture volume in mm
 - *Initial soil storage* can be calculated by dividing the soil moisture volume in mm by *maximum soil storage* for each sub-basin
- *Maximum GW 1 storage*, *maximum GW 1 percolation*, *GW 1 storage coefficient* and *linear reservoir GW 1 storage coefficient*
 - Process for estimating subsurface flow parameters is described in detail in Fleming and Neary (2004) and Linsley et al. (1958).
 - Plot daily average flow versus time for large storm on semi-log graph
 - Take the natural log of linear reservoir equation and fit it to the recession of the hydrograph; this line represents the baseflow
$$Q(t) = Q_0 e^{-t/k} \rightarrow \ln Q(t) = -\frac{t}{k} + \ln Q_0$$
 - Subtract the baseflow from the total storm hydrograph; the resulting hydrograph represents the interflow + surface flow
 - SMA requires the use of a “Baseflow” linear reservoir to route interflow to the basin outlet in addition to the SMA Groundwater 1 reservoir; therefore, interflow is modeled using two linear reservoirs.
 - Take the natural log of equation for 2 linear reservoirs in series and fit to the recession of the interflow + surface flow hydrograph; this line represents interflow

$$Q(t) = Q_0 \frac{e^{-t/k}}{k} t \rightarrow \ln Q(t) = -\frac{t}{k} + \ln t + \ln Q_0 - \ln k$$

- This equation is used to estimate the SMA groundwater and baseflow parameters
 - *GW 1 storage coefficient* and *linear reservoir GW 1 storage coefficient* = k
 - *Maximum GW 1 storage* = $Q_0 k / A$
 - *Maximum GW 1 percolation* initially estimated to be between 1-3 mm/hr then calibrated
 - *Maximum soil percolation rate* was estimated based on infiltrometer tests of weathered bedrock data in the Front Range (Ebel, 2016)
 - The average hydraulic conductivity of the infiltrometer tests is 82.9 mm/hr
 - *Maximum soil percolation rate* = 82.9 mm/hr; input under Loss tab in HEC-HMS for all basins
5. Clark Unit Hydrograph:

Use this portion of the document in conjunction with the Python script ‘Tc_model.py’ in Appendix D. Variable names in **bold** denote files that must be created or updated by the user. The remaining variables are created internally by the model. The user must update all file paths in the script for input, internal, and output variables.

The instructions provided in this section assist with estimation of the Time Area Curve which describes the distribution of travel times through the sub-basin. The *time of concentration* T_c and *storage coefficient* R parameters can then be calculated from the Time Area Curve. These parameters can be input into HEC-HMS under the Transform method tab.

Time Area Curve:

- Manning’s roughness:
 - Variable name in ‘Tc_model.py’: ***mannings_n***
 - Download NLCD 2011 Land Cover data from USGS National Map download (<https://apps.nationalmap.gov/download/>)
 - Add Manning’s n field to Land Cover raster and populate field using values in Table 6.1 from Follum et al. (2016). These values can be adjusted within the Low to High range later to ensure realistic implied channel velocities (see *chantime* variable below)
 - Use Lookup tool and set the Lookup field as the new manning’s n field that was just created
 - Input raster: land cover raster from NLCD
 - Lookup field: field which contains manning’s n values
- DEM:
 - Variable name in ‘Tc_model.py’: ***DEM***
 - See Section 1. Digital Elevation Model above
 - The elevations should be in metric
- Stream extent:
 - Variable name in ‘Tc_model.py’: ***stream_extent***
 - This variable defines the locations where the model uses channel flow equations rather than overland flow equations. The raster is created using the ArcHydro tools and varying the contributing area threshold until the extent of the streams reasonably matches what can be observed from aerial imagery. The threshold used to create this raster will be significantly smaller than the threshold in developing the sub-basins. ArcHydro tools and the contributing area threshold are explained in Section 2. Subbasin Disaggregation.
- NF and SF polygons:
 - Variable name in ‘Tc_model.py’: ***asp_reclass***
 - This shapefile contains north- and south-facing polygons and was developed by reclassifying the aspect raster. It was previously developed for Section 2. Subbasin Disaggregation.

- Flow Accumulation Raster:
 - Variable name in ‘Tc_model.py’: *fac*
 - Flow accumulation raster as developed previously in Section 2. Subbasin Disaggregation.
- Flow Direction Raster:
 - Variable name in ‘Tc_model.py’: *fdr*
 - Flow direction raster as developed previously in Section 2. Subbasin Disaggregation.
- Cell size
 - Variable name in ‘Tc_model.py’: *cellsize*
 - A constant which represents the DEM cell size. Update this value to correspond to DEM’s resolution.
- Sub-basin shapefiles:
 - Variable name in ‘Tc_model.py’: *sub*
 - The *sub* variable includes separate shapefiles for each sub-basin in the watershed. A for loop is used to iterate through each sub-basin and read the appropriate sub-basin’s boundary to the variable. The shapefile is used to set the extent of the analysis. The Accum_Time function that is called in Tc_model.py iterates through each cell within the shapefile’s extent, so it is important to minimize the extent to increase the function’s efficiency.
 - These shapefiles can be developed simply by exporting the individual features in the sub-basin shapefile created in Section 2. Sub-basin Disaggregation. Be sure to name the shapefiles consistent with the Tc_model.py script (e.g. extent_1.shp for sub-basin 1)
- DEM clipped to each subbasin
 - Variable name in ‘Tc_model.py’: *DEM_sub*
 - Similar to the *sub* variable, the portion of the DEM within each subbasin is extracted with each for loop. Each sub-basin’s DEM is extracted using the Extract by Mask tool in ArcPy. The resulting DEM is the initial input to the processes within each for loop.
- Slope
 - Variable name in ‘Tc_model.py’: *slope*
 - The slope of each subbasin is calculated using the ArcPy Slope tool. The resulting slope grid is in percent slope.
- Slope grid with reservoir slopes filtered out
 - Variable name in ‘Tc_model.py’: *slpprcnt*
 - Low or zero slope values within reservoirs/lakes produce unrealistically low velocity values, causing portions of the subbasins to drain too slowly. A slope threshold should be established such that slopes below the threshold are located primarily within reservoirs with minimal occurrences outside of the reservoirs. Slopes below the threshold are replaced using the ArcPy Con tool so that the new values provide a reasonable minimum flow velocity. The threshold can also be adjusted to obtain a reasonable average flow velocity for each sub-basin
 - The Con tool is already incorporated into the Python code, but the user must adjust the value to obtain reasonable minimum and average velocities. The threshold is currently set at 0.4
- Decimal slope grid
 - Variable name in ‘Tc_model.py’: *slpdec*
 - Converts the percent slope grid to decimal slope using the ArcPy Divide tool
- Channel slope grid
 - Variable name in ‘Tc_model.py’: *chanslp*
 - Extracts the channel slope from the decimal slope grid using the Extract by Mask tool and *stream_extent* as the mask.
- Flow time per channel cell grid

- Variable name in ‘Tc_model.py’: *chantime*
- The flow time per channel cell is calculated using a series of ArcPy processes. The processes include several intermediate variables that must be calculated to complete the travel time calculation. Manning’s equation is used to calculate the travel time per cell:

$$t = \frac{d \cdot n}{Rh^{2/3} \cdot \sqrt{s}}$$
- t: time per cell, d: distance (cell size), n: manning’s roughness coefficient, Rh: hydraulic radius, s: slope
- *Intermediate variables:* The hydraulic radius raster, Rh, is calculated using downstream hydraulic geometric relationships. Livers and Wohl (2014) collected bankfull depth and width measurements at 111 sites throughout the Colorado Front Range. The data were used to establish relationships for depth vs drainage area ($d = c2 \cdot DA^f$) and width vs. drainage area ($w = c1 \cdot DA^b$). The relationships are used to calculate depth and width for the entire sub-basin where DA is determined from the flow accumulation raster. The wetted perimeter, P, and flow area, A, can then be calculated. A and P are then used to calculate Rh.
- Flow velocity per overland cell grid
 - Variable name in ‘Tc_model.py’: *Vel*
 - The overland flow velocity is calculated using an approximation of manning’s equation:

$$V = k \cdot \sqrt{s}$$
 - k: roughness k-factor, s: slope
 - ‘Tc_model.py’ includes two options: (1) cell-specific k-factor based on the landcover dataset, and (2) a constant k-factor that can be adjusted to achieve realistic overland velocities. Because the overland flow velocity equation is based on manning’s equation, k represents the variables n and Rh. Because we have no method to estimate or measure Rh for every overland cell in the subbasin, using a cell-specific k value implies that we know more information than we actually do. For this reason, we used the constant k-factor method for this study. Therefore, we initially set a constant k of 0.4 and adjust the value to achieve a realistic average mean velocity.
- Flow time per overland cell grid
 - Variable name in ‘Tc_model.py’: *overlandtime*
 - The overland travel time per cell is calculated as:

$$t = \frac{d}{v}$$
 - t: travel time per overland cell, d: cell size, v: velocity grid (*Vel*)
- Combined travel time per cell raster
 - Variable name in ‘Tc_model.py’: *travelttime*
 - Uses the ArcPy tool Mosaic to New Raster to combine the overland and channel travel time rasters to create a travel time per cell raster for the entire basin.
- Accumulated travel time raster
 - Variable name in ‘Tc_model.py’: *totaltime*
 - Calls the user-defined function Accum_Time to calculate the accumulated travel time to each point in the subbasin.
 - Intermediate variable within Accum_Time that user must provide:
 - Variable name in ‘Tc_model.py’: *outs_layer*
 - Shapefile of sub-basin outlet points, which must have a field named “Id” with a value of 0. The points must be located on the cell corresponding to the highest flow accumulation within each sub-basin. If the points are placed improperly, the code could get caught in an infinite loop.
 - The Accum_Time function calculates the accumulated travel time raster as follows:

- Extracts *traveltime* raster to sub-basin boundary and convert raster to an array for processing
 - Clips flow direction raster to sub-basin boundary using Extract by Mask tool
 - Replaces the outlet cell value of the flow direction raster of each sub-basin with a value of zero using the Point to Raster and Mosaic to New Raster tools. This step is completed because the algorithm used by this function iterates through the sub-basin until it reaches a flow direction value of zero. The new flow direction raster is named *fdROUT*.
 - Creates array of *fdROUT* for processing using the Raster to NumPy Array tool
 - Creates empty array with the same dimensions as the flow direction array
 - Iterates through the time array and change any negative values to zero
 - Iterates through each cell in the subbasin's array and calculates the travel time to the outlet for each cell. The code first checks to ensure the current cell is located within the subbasin. If it is, the code then follows the flow direction raster to the basin's outlet, accumulating the individual cell travel times along the way. Once it reaches the basin outlet, the total accumulated travel time is assigned to that cell.
 - Converts the accumulated travel time array to a georeferenced raster so that it can be viewed in ArcMap.
- Cumulative Density Function of travel times (CDF for entire sub-basin is calculated first, then separate CDFs for NFS and SFS are completed in a for loop)
 - Variable name in 'Tc_model.py': *Perc_Time* and *Perc_Area*
 - *Perc_Area* and *Perc_Time* are plotted as the x and y ordinates, respectively, of the cumulative distribution function. This plot is used as an input to HEC-HMS as the Time-Area Method Percentage Curve under the Transform tab.

Time of Concentration and Storage Coefficient:

- Since the *time of concentration* T_c is the time of travel of the hydraulically most distant point in a basin, it is determined to be equal to the longest time from the time-area curve. The *storage coefficient* R is then calculated using methods developed by Sabol (2008):

$$R = 0.37T_c^{1.11}L^{0.8}A^{-0.57}$$

- R : storage coefficient (hr), T_c : time of concentration (hr), L : length of watercourse to hydraulically most distant point (miles), A : area (mi²)
- These values can be input into HEC-HMS under the Transform tab

6. Routing

The instructions provided in this section assist with estimation of the *Length (m)*, *Slope (m/m)*, *Manning's n*, *Cross Section*, *Left Manning's n*, and *Right Manning's n* parameters under the Muskingum-Cunge with Eight Point Cross Section method. These parameters can be input into HEC-HMS under the Routing tab.

- *Length (m)*, *Slope (m/m)*, and *Cross Section*
 - *Length (m)* and *Slope (m/m)* are measured from aerial imagery and the DEM
 - *Cross Section*
 - Assume rectangular channel
 - Channel width data is obtained from aerial imagery and the DEM
 - Flow depth is calculated iteratively using Manning's equation and assuming a flow that is half of the maximum flow during the storm of interest
 - Floodplain width is measured from the DEM
- *Manning's n*, *Left Manning's n*, and *Right Manning's n*

- Estimated based on channel type and floodplain vegetation based on aerial imagery and representative roughness values (Chow, 1959)

