Soil Infiltration Calculations Using ArcGIS Pro: Creating an SCS* Curve Number Grid

CPLN 675, Spring, 2022

Instructor: Michael Fichman - mfichman@design.upenn.edu

Methodology adapted from Venkatesh Merwade, Perdue University

Adapted for an ArcGIS Pro workflow by Jenna Epstein, University of Pennsylvania

* - “SCS” stands for Soil Conservation Service – a division of the US Department of Agriculture which devised the metric.

The data and some complimentary R and Python code are located in the Week 3 folder in the class Github at:

https://github.com/mafichman/CPLN_675

You will want to have this repo open so you can copy and paste a code block in one of the later steps of the analysis.

Overview

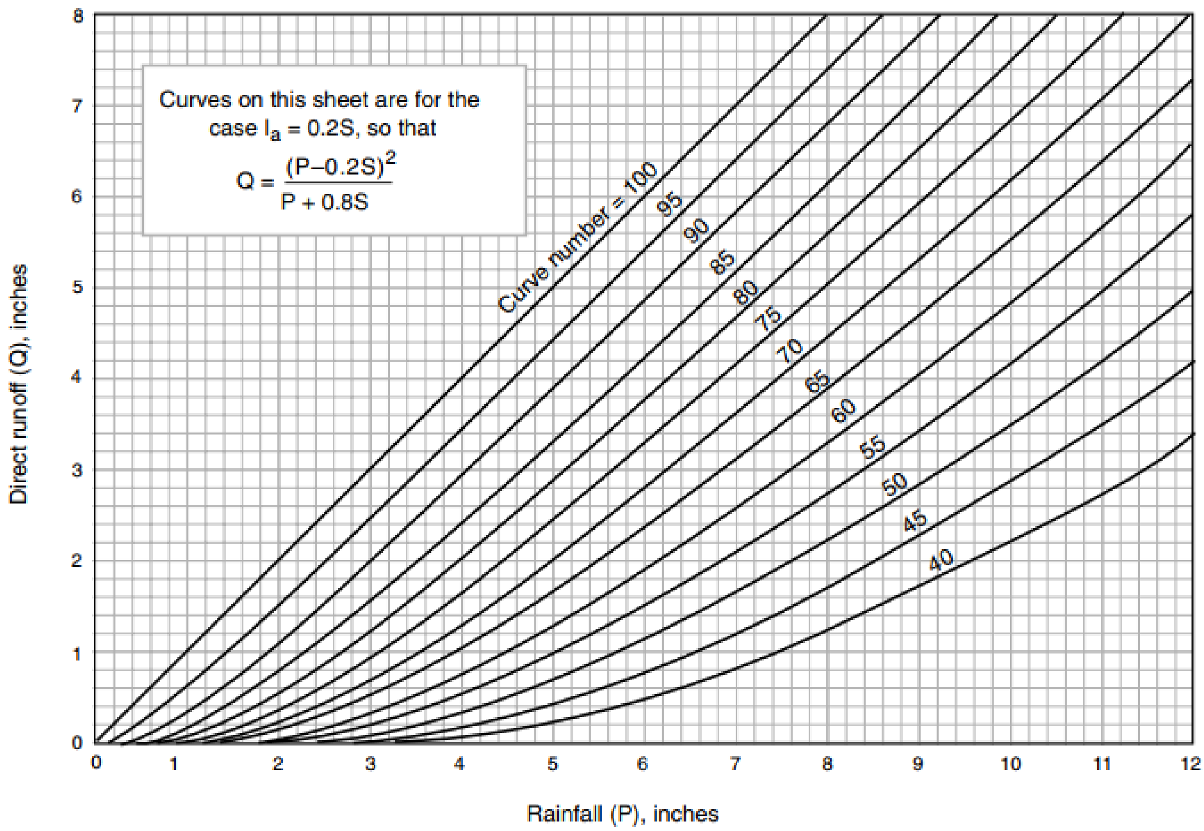
The objective of this tutorial is to calculate runoff potential using soil and land use data. Run off curve number or simply “curve number” is a parameter used in hydrology for predicting direct runoff or infiltration from rainfall.

We will use to use ArcGIS to wrangle land use data and relate it to soil data from a national soil survey. The calculations we will perform take these land cover and soil data inputs to create a curve number (CN) based on a set of assumptions about how the two inputs interact. Simply, the type of land (e.g. a farm) and the type of soil (e.g. a sandy soil) are assumed to “on average” produce a certain amount of runoff given a certain volume of rainfall.

This curve number can be used as a parameter in other models to understand how much of a landscape in a given watershed will drain overland into a water body or into a sewer system. This is important for understanding the degree to which runoff may cause risk of flooding. It is also useful in order to understand the potential for nutrients like N and P or trace minerals or sediments to be carried into rivers, lakes, and estuaries. This transport has the potential to cause eutrophication and other types of pollution.

Later this semester you will be building more sophisticated flood models, and you can build CN information into statistical models to attempt to predict inundation.

The chart at the top of the following page describes the relationship between runoff and rainfall associated with the CN.



Data:

The data are available on the class Piazza page, and on the class Github.

Our demo area is the watershed of Cedar Creek, in northeastern Indiana. Files include:

- ***cedar_lu* –** Despite its name, don't confuse this as "land use". It is land cover, similar to what we used last week in the fragstats dataset.
- ***cedar_dem* –** This is a digital elevation model. What is the grid cell size?
- ***cedar_clip*** – This is a shapefile clipped to the Cedar Creek area exported from an access database of soil data. It's the highest resolution open soil data available.

Steps:

Preparation: Create a new project in ArcGIS Pro, and set its location to wherever your week 3 data folder sits on your computer. Save the new project as “cnGrid”.

1. Add ***cedar_lu*** to the map. Add the **DEM** *(**cedar_dem**)* as well.

2. **Set/confirm the environments and extent** to align with the ***cedar_lu*** layer.

3. **Reclassify *cedar_lu*** into higher level classes as below. We assume these four categories are similar in terms of their infiltration potential. Let’s call this ***cedar_lu_rc*** *(recommend saving this and your other new files into a new folder called “new” within your week 3 data folder).*

Original NLCD classification		Revised classification (re-classification)	
Number	*Description*	*Number*	*Description*
11	Open water	1	Water
90	Woody wetlands		
95	Emergent herbaceous wetlands		
21	Developed, open space	2	Medium Residential
22	Developed, low intensity		
23	Developed, medium intensity		
24	Developed, high intensity		
41	Deciduous forest	3	Forest
42	Evergreen forest		
43	Mixed forest		
31	Barren land	4	Agricultural
52	Shrub/scub		
71	Grassland/herbaceous		
81	Pasture/hay		
82	Cultivated crops		

4. **Convert these raster areas to polygons** using “VALUE” as the field. Let’s call this ***landusepolygons.*** Then, symbolize by “gridcode”. What do we see here about the geography and hydrography of the study area?

5. Now we’ll work with the soil data. Add **cedar_clip_soil** to the project. What kind of data is this? What do you think the geometry represents?

6. Open the attribute table. You’ll notice there are four distant soil hydrologic groups in the “SoilCode” column - A,B,C, and D. These are based on grouping of soil types that have the same run off potential.
   - **A -** soils having high infiltration rates; chiefly sands or gravel.
   - **B -** moderate infiltration

- **C -** slow infiltration
- **D** - very low infiltration like clay

More information is available from the USDA.

7. **Create four percentage fields** named **PctA, PctB, PctC**, and **PctD** all of type long in **cedar_clip_soil**. There may be instances where polygons have a mix of soil types, so we want to assign relative percentages over four fields A-D.

8. We are going to make some assumptions about the relative percentages of soil types in our geometries. If you have an observation that's just a single soil type, **input** a value of **100** for each **PctA, PctB, PctC**, and **PctD** using **select by attributes** and **field calculator.** This means that if a geometry is soil type A, it gets a 100 for **PctA** and a zero for **PctB** etc. If it's split, say, "B/C" give it a 50/50 split between each type.

   There are lots of ways to do this – and sometimes ArcGIS gets really cranky if you do field calculations in a way it doesn't like. You can try to do a python expression if you want, there is also a script in the course folder to process these data in R. If you are having really big trouble and getting lots of errors, there are backup data in the Week 3 Github.

9. Output a shapefile called ***soil_clip_joined.***

10. **Union** the ***soil_clip_joined*** data to ***landusepolygons*****.** It is okay if ArcGIS Pro sends this to a temporary geodatabase (not saving to our "new" folder) because in the next step, we will be editing it and resaving it.

11. We have lots of geographic artifacts in the new unioned data – places where FID = -1 and gridcode = 0. Highlight these and take a look at what they are. Use **Select by Attribute** to get rid of 'slivers' by selecting all the features that have "FID_..." = -1 and deleting them. Make sure to save your edits. You will have to do this twice – once for the "FID_landus" column and once for the "FID_soil_c" column (there are two fields beginning with FID_). Make sure to save. Now we can export, and call the resulting feature class ***union_lu_soils***.

12. Create a new categorical column called "LandUse" in ***union_lu_soils***. We already have this information in the "gridcode" field. Click "Add Field" and name it "LandUse" – then make sure to Save. Use "Calculate Field" to set this new field equal to "gridcode."

13. Now we are done with the geoprocessing. The next step is to calculate curve numbers for different combinations of land uses and soil groups. We are going to apply the CN values from the following table showing relationship between Land Use (LUValue and corresponding Descrip) and A, B, C, and D soil types.

   Pause for a moment and look at these values in the table below and think about what CN values really mean – does this seem reasonable to you?

LUVALUE	DESCRIP	A	B	C	D
1	Water	100	100	100	100
2	Medium Residential	57	72	81	86
3	Forest	30	58	71	78
4	Agricultural	67	77	83	87

HEC-RAS, the USGS ArcGIS extension we used to use in this analysis, has not been updated to work with ArcGIS Pro yet, so we are going to re-invent it. The way HEC-RAS worked is that it would take a table like the one above, that you would create, and then “look” at each row in the data and apply a CN value based on LU and Soil Type.

We are going to use the Python language to define a function to return the appropriate CN value based on the LandUse and SoilCode values. (For more information about using Python expressions in field calculations, check out the examples in ESRI’s documentation.)

OK, so **add a new field to your shapefile** and call it “CN”. Then **open the Field Calculator**. We are going to paste some code into the window, but first, let’s look at what this code below does – it’s a bunch of “if else” statements – so if LUVALUE is 1 (Water), and SoilCode is A – return a value of 100 for the CN.

Choose “Python 3” for the Expression Type, then enter the following in the Expression box and the Code Block box (you can copy and paste this from a .txt file in the class github repo called “code/python/python_expressions.txt”):

In the Expression box:

```
calcCN(!LandUse!, !SoilCode!)
```

In the Code Block box:

```
def calcCN(LandUse, SoilCode):
  if (LandUse == 1 and SoilCode == "A"):
    return 100
  elif (LandUse == 1 and SoilCode == "A/D"):
    return 50
  elif (LandUse == 1 and SoilCode == "B"):
    return 100
  elif (LandUse == 1 and SoilCode == "B/D"):
    return 50
  elif (LandUse == 1 and SoilCode == "C"):
    return 100
  elif (LandUse == 1 and SoilCode == "C/D"):
    return 50
  elif (LandUse == 1 and SoilCode == "D"):
    return 100
  elif (LandUse == 2 and SoilCode == "A"):
    return 57
  elif (LandUse == 2 and SoilCode == "A/D"):
    return 71.5
  elif (LandUse == 2 and SoilCode == "B"):
    return 72
  elif (LandUse == 2 and SoilCode == "B/D"):
    return 79
  elif (LandUse == 2 and SoilCode == "C"):
    return 81
  elif (LandUse == 2 and SoilCode == "C/D"):
    return 83.5
  elif (LandUse == 2 and SoilCode == "D"):
```

```
    return 86
  elif (LandUse == 3 and SoilCode == "A"):
    return 30
  elif (LandUse == 3 and SoilCode == "A/D"):
    return 54
  elif (LandUse == 3 and SoilCode == "B"):
    return 58
  elif (LandUse == 3 and SoilCode == "B/D"):
    return 68
  elif (LandUse == 3 and SoilCode == "C"):
    return 71
  elif (LandUse == 3 and SoilCode == "C/D"):
    return 74.5
  elif (LandUse == 3 and SoilCode == "D"):
    return 78
  elif (LandUse == 4 and SoilCode == "A"):
    return 67
  elif (LandUse == 4 and SoilCode == "A/D"):
    return 77
  elif (LandUse == 4 and SoilCode == "B"):
    return 77
  elif (LandUse == 4 and SoilCode == "B/D"):
    return 82
  elif (LandUse == 4 and SoilCode == "C"):
    return 83
  elif (LandUse == 4 and SoilCode == "C/D"):
    return 85
  elif (LandUse == 4 and SoilCode == "D"):
    return 87
```

If you open the attribute table for ***union_lu_soils***, you'll now see the new column "CN" populated with the returned values.

14. Adjust the symbology of ***union_lu_soils*** to reflect the unique values in the "CN" column. Feel free to create a custom color ramp that stretches from lower CN to higher CN values for ease of visualization.

Let's revisit the chart from the first page of this tutorial and our soil descriptions. Consider the values in our table (which come from a USGS model from the 1980s). How do these numbers translate to ABCD above?

Questions to ponder:

Let's consider a situation in which we'd want to model the probability of a flood in a certain spot, given a certain level of rainfall. How does knowing these curve numbers help you do that? What else might you need to know?

What kinds of assumptions did we make in this analysis? Can you imagine how these assumptions might cause error in our outputs? How would you know these are errors?