

Homework 2 – Problem 4

The Height Above Nearest Drainage (HAND) methodology has recently been adopted by Xing Zheng and Dr. David Maidment at the University of Texas at Austin to improve flood inundation mapping efforts; this is done by generalizing National Elevation Dataset (NED) grid-cells (10m x 10m) as the height above their nearest drainage location (found by following a cell down the path of steepest descent until intersecting with the bottom of a stream reach). The HAND method is relevant to this project because of the data outputs it provides: stream variables (particularly wetted area, hydraulic radius, and slope), and the resulting rating curves derived from these hydraulic properties, can be used to compare the computed HAND rating curves with the “correct” United States Geological Survey (USGS) rating curves. Of note, the current HAND rating curves are determined assuming a manning’s n of 0.05 for all stream reaches (see equation 1); this is the motivation for this study – I would like to programmatically determine what the “correct” manning’s n should be on a large scale, and perhaps statistically compare these corrected n values to the 0.05 assumption currently used.

$$Q = \frac{k}{n} A R^{\frac{2}{3}} S^{\frac{1}{2}} \quad (1)$$

where:

Q is the discharge (L^3/T ; ft/s, m/s);

A is the cross-sectional area (L^2 ; ft², m²);

R is the hydraulic radius (L; ft, m);

S is the channel bed slope at constant water depth (L/L , ft/ft, m/m).

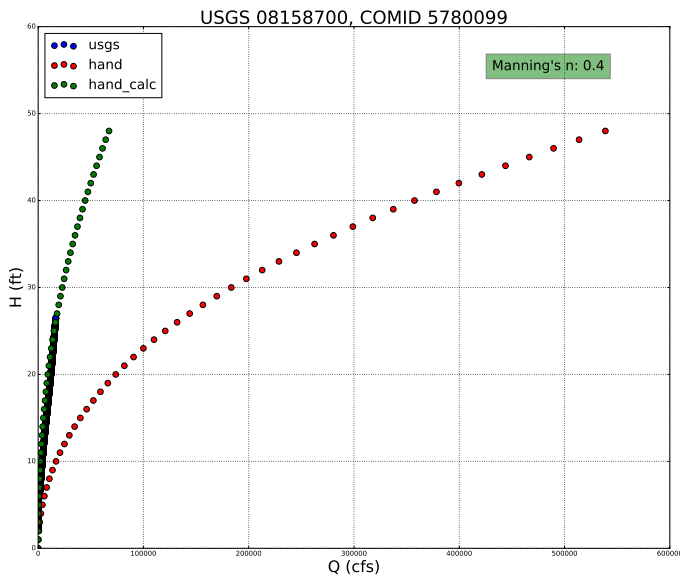
k is a conversion factor, 1.0 for SI units and 1.49 for English units.

In short, this project has changed to become more of a statistical curve-fitting of HAND rating curves to USGS rating curves. The intent is to programmatically snap a HAND rating curve to the USGS rating curve; this involves a statistical optimization of sorts (I’m still figuring this part out) including an adjustment of the HAND rating curve to the USGS stream bottom depth (which itself requires adjustment), and a subsequent fitting of the HAND rating curve to the USGS rating curve by adjusting manning’s n (roughness). The first step, then, requires collection of the actual data: USGS rating curves were collected online (ex: http://waterdata.usgs.gov/nwisweb/get_ratings?file_type=exsa&site_no=08158810, where the last eight digits are the USGS identifier), and HAND rating curves and hydrologic information were received from Xing Zheng for the Onion Creek watershed in Austin, Texas. Onion Creek will be used as an initial test case, with the intent of expanding the final rating curve optimization scheme to all of Texas (and, eventually, all of the United States).

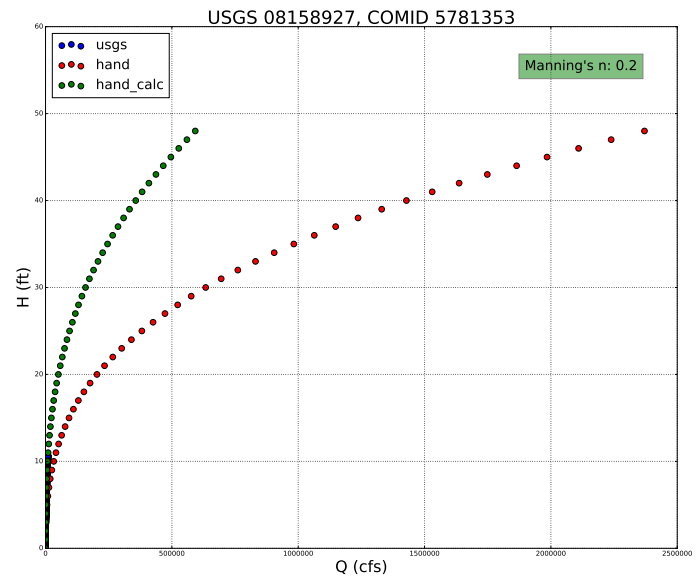
After the data was collected, a preliminary assessment was conducted in which the “correct” manning’s n for each of eight stream reaches within the Onion Creek watershed were manually determined. Figures 1a through 1h (see pages 3 and 4) show both the HAND rating curves (red), the USGS rating curves (blue), and the HAND rating curve after it has been manually fitted to the USGS rating curve (green); the manning’s n value displayed in the upper right-hand corner of each image is the roughness that was used for the fit.

The first thing that is evident is that the “correct” manning’s n values are quite high. As seen in the following url: http://www.fsl.orst.edu/geowater/FX3/help/8_Hydraulic_Reference/Mannings_n_Tables.htm, manning’s n is generally between 0.02 and 0.07, with values rarely getting

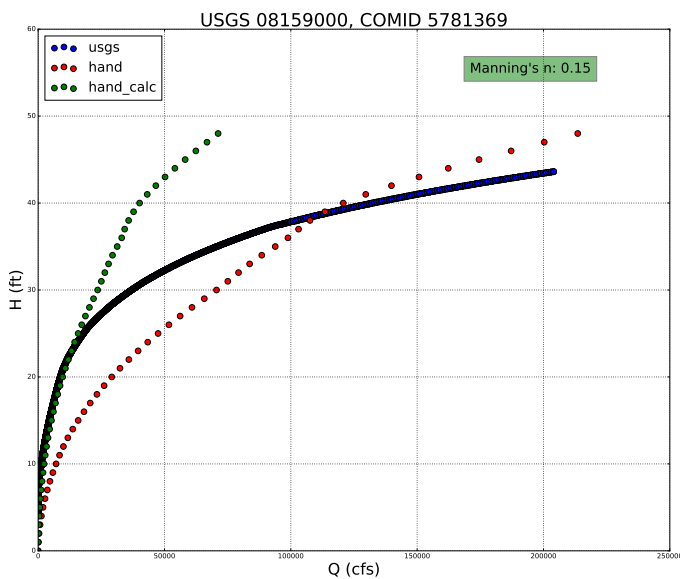
above 0.2. As such, I will need to spend some time reviewing the Python script I am using (Listing 1) to be certain that my mathematical approach and units are correct. Once I am certain that the mathematics is correct, I will proceed to automate the curve fitting.



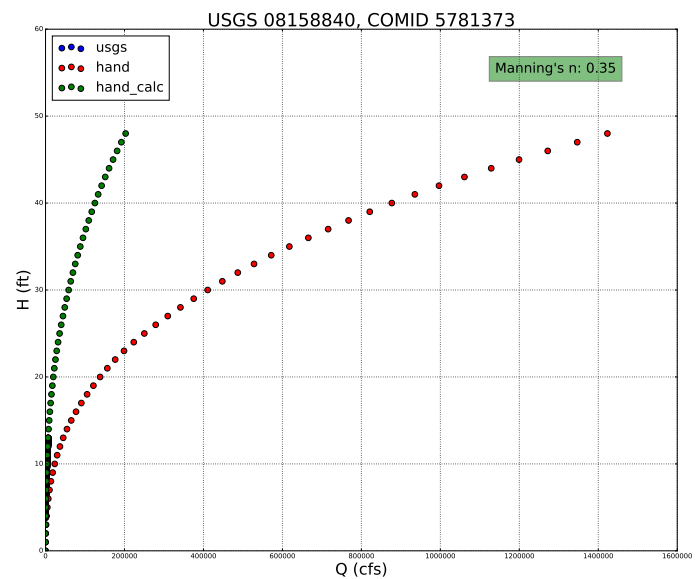
(a) Rating Curve for COMID 5780099



(b) Rating Curve for COMID 5781353

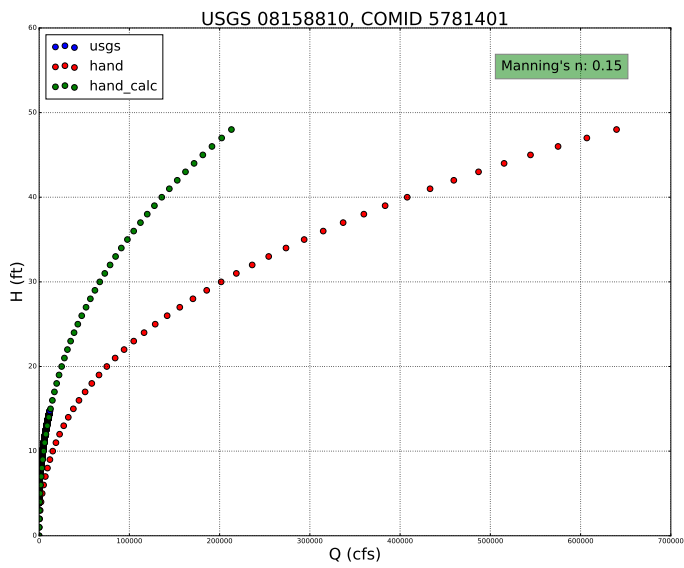


(c) Rating Curve for COMID 5781369

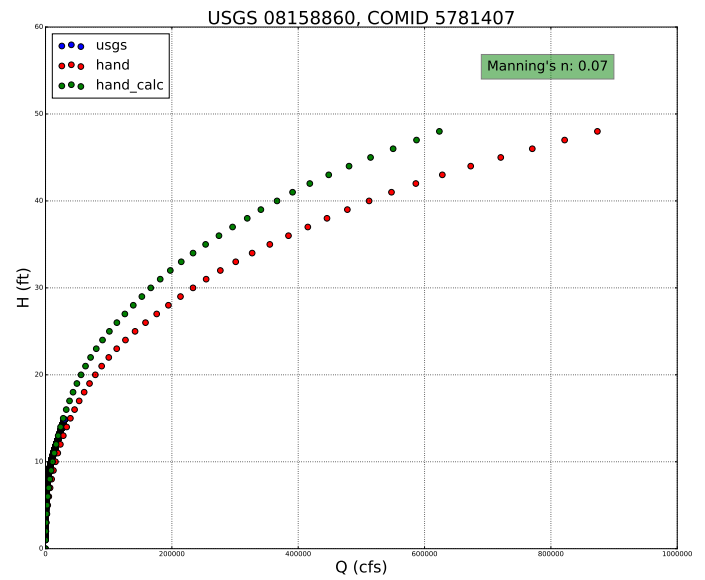


(d) Rating Curve for COMID 5781373

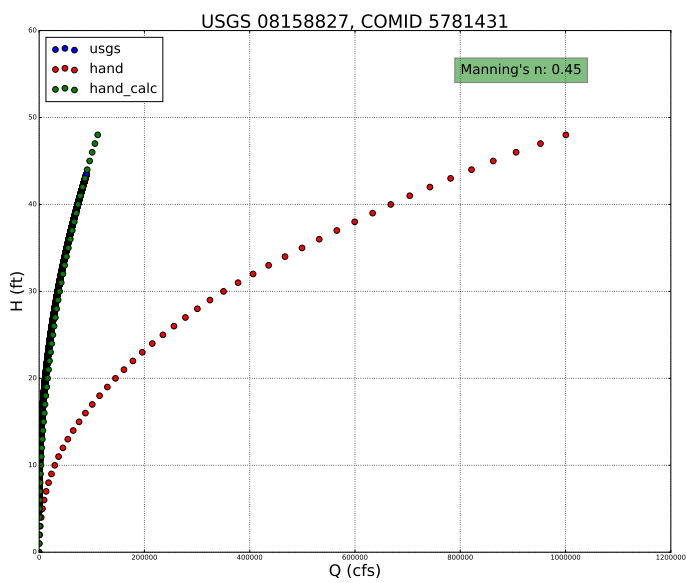
Figure 1: Rating Curves for the Onion Creek watershed in Austin, Texas



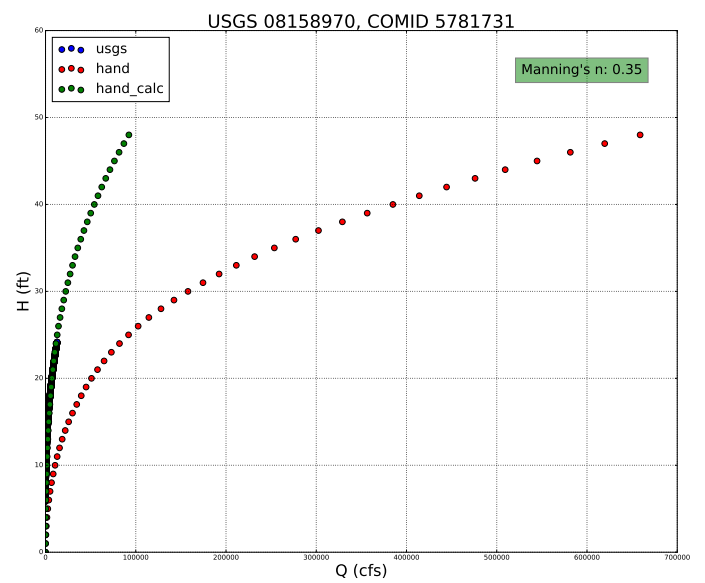
(e) Rating Curve for COMID 5781401



(f) Rating Curve for COMID 5781407



(g) Rating Curve for COMID 5781431



(h) Rating Curve for COMID 5781731

Figure 1: Rating Curves for the Onion Creek watershed in Austin, Texas

Regarding the automation of the curve fitting, there are two main steps:

1. Adjust HAND rating curve to fit USGS rating curve along the y-axis (Height, ft) to correct for the USGS accepted bank bottom. I am still devising the solution to this problem, but – after taking into account the vertical “shift” – I believe I will need to find the highest depth associated with a discharge of zero (USGS rating curves often have many depths with $Q = 0$, including depths up to multiple feet high) and then set that height as a “true” zero. Once this zero is obtained, I will need to correct the HAND rating curve heights to shift them appropriately.
2. Fit each vertically-adjusted HAND rating curve to the USGS rating curve by taking each HAND height and its associated hydraulic parameters, and looking up the discharge associated with that height in the USGS rating curve table; with all these data points, manning’s n can easily be computed and averaged over all heights for each stream reach. This idea should work fine, but I am skeptical that the results will be beneficial: by relying on the USGS rating curves to compute the HAND rating curves, I can not see a clear path forward for generalizing the HAND rating curve computation for reaches without USGS rating curves. As such, I think a statistical approach would be best for the long-term. I’m not entirely sure how this approach may work, but I would suspect there are ways to statistically fit the HAND rating curves to the USGS rating curves (by minimizing the difference between discharge values at the same height, perhaps using an ANOVA or Kruskal-Wallis test and accepting the fit once the null hypothesis is no longer rejected). I am also anticipating that we will learn ways for statistically correlating datasets, such that I may correlate manning’s n with the National Land Cover Dataset I mentioned in the first portion of this assignment.

Both these steps are starting to come together, but this is still a work in progress and as such any advice would be very welcome. The intent at this point is to proceed with steps 1 and 2 until Onion Creek is working appropriately. Once this is done, I will try to expand my code to function on all of the stream reaches in Texas that have USGS gage locations. To do this, I will test over some fraction of the stream reaches (ie. 90%), use the results to inform some sort of machine-learning algorithm that I hope to implement, and finally verify and/or validate the functionality on the remaining smaller fraction (ie. 10%) of the stream reaches in Texas. This would make room for a further statistical analysis of the correctness of my algorithm, and once that analysis passes in a statistically-significant manner, I can safely proceed to automate my problem for all of Texas (including all stream reaches that do not contain USGS rating curves for potential validation).

Listing 1: Class CompareRC defined in ratingcurves-onionck-clean.py module to quickly collect, extract, calculate, and display the USGS/HAND rating curves shown in Figure 1.

```

1 class CompareRC:
2
3     def __init__(self, idlookup, handrc, handrcidx, handnc, handncidx):
4         self.idlookup = idlookup
5         self.handnc = handnc
6         self.handrc = handrc
7         self.handrcidx = handrcidx
8         self.handncidx = handncidx
9         self.usgsrsrc = 'http://waterdata.usgs.gov/nwisweb/get_ratings?file_type=
        exsa&site_no={0}'
10
11     def get_usgsrsrc(self, usgsid):
12         urlfile = urllib.urlopen(self.usgsrsrc.format(str(usgsid)))
13         urllines = urlfile.readlines()
14         check = False
15         usgsq = []
16         usgsh = []
17         for j in range(len(urllines)):
18             line = urllines[j]
19             if not re.search('[a-zA-Z]', line): # No letters
20                 check = True
21             if check:
22                 usgsq.append(float(line.split('\t')[2]))
23                 # apply shift in [1] to stage height
24                 usgsh.append(float(line.split('\t')[0]) - float(line.split('\t')[1]))
25         return (usgsq, usgsh)
26
27     def get_handrc(self, comid):
28         handq = self.handrc.variables['Q_cfs']
29         handh = self.handrc.variables['H_ft']
30         handc = self.handrc.variables['COMID']
31         idx = self.handrcidx.loc[self.handrcidx['comid'] == comid]['index'].values
32         [0]
33         if handc[idx] == comid:
34             handq = handq[idx]
35             return (handq, handh)
36
37     def mannings(self, wetarea, hydraulicradius, slope, en):
38         return wetarea*scipy.power(hydraulicradius, (2/3.0))*sqrt(slope)/en
39
40     def get_handnc(self, comid, en=0.05):
41         idx = self.handncidx.loc[self.handncidx['comid'] == comid]['index'].values
42         [0]
43         handstage = self.handnc.variables['StageHeight']
44         handc = self.handnc.variables['COMID']
45         handslope = self.handnc.variables['Slope']
46         handarea = self.handnc.variables['WetArea']
47         handrad = self.handnc.variables['HydraulicRadius']

```

```

46     if handc[idx] == comid:
47         handarea = handarea[idx]
48         handrad = handrad[idx]
49         handslope = handslope[idx]
50         handdisch = self.mannings(wetarea=handarea,hydraulicradius=handrad,slope=
handslope,en=en)
51         handstage = handstage[:49]*3.28084 # Convert m to ft
52         handdisch = handdisch[:49]*35.3147 # Convert cms to cfs
53     return (handstage,handdisch)
54
55 def get_usgs_n(self,usgsid):
56     comid = self.idlookup.loc[self.idlookup['SOURCE_FEA'] == usgsid]['FLComID'
].values[0]
57     stage = self.handrc.variables['H_ft']
58     usgsq,usgsh = get_usgsrc(usgsid)
59
60 def plot_rcs(self,comid):
61     usgsid = self.idlookup.loc[self.idlookup['FLComID'] == comid]['SOURCE_FEA'
].values[0]
62     usgsq,usgsh = self.get_usgsrc(usgsid)
63     if usgsq == usgsh == []:
64         print 'USGS Rating Curve does not exist for COMID {0}'.format(str(comid)
)
65         return
66     handq,handh = self.get_handrc(comid)
67     en = 0.15
68     stage,disch = self.get_handnc(comid,en)
69     fig, ax = plt.subplots()
70     fig.set_size_inches(20,16, forward=True)
71     ax.scatter(usgsq,usgsh,c='b',s=100,label='usgs')
72     ax.scatter(handq,handh,c='r',s=100,label='hand')
73     ax.scatter(disch,stage,c='g',s=100,label='hand_calc')
74     plt.gca().set_xlim(left=0)
75     plt.gca().set_ylim(bottom=0)
76     ax.set_title('USGS {0}, COMID {1}'.format(str(usgsid),str(comid)),fontsize
=32)
77     plt.xlabel('Q (cfs)',fontsize=28)
78     plt.ylabel('H (ft)',fontsize=28)
79     ax.text((handq[-1]*0.8),55,"Manning's n: {0}".format(str(en)),
horizontalalignment='left',
80             fontsize=24,bbox={'facecolor':'green', 'alpha':0.5, 'pad':10})
81     plt.legend(loc='upper left',fontsize=24)
82     plt.grid()
83     plt.show()
84     fig.savefig('manualresults/rc_comid_{0}.pdf'.format(comid))

```