Optimizing Social Climate Action: AI Assisted Reconstruction of Riverbeds for Global Climate Risk Management

Ryan Chen
Geothara Team 1A
Fall 2023, AI Studio Project Write-Up

Business Context

Climate change remains one of the most pressing issues facing our society in the 21st century, and if left undealt with, it can lead to many consequences on our planet, including hotter temperatures, more severe storms, the loss of species, and increased health risks. The impact of climate change is also critical when it comes to our river systems, yet our understanding of this is rather limited. In the United States alone, there are approximately 2.7 million 2-km river segments, but only a small fraction — approximately 0.3% of such river segments — have gauges on them to measure their flow. While satellite technology has long been used to measure river widths and approximate flow rates, its use is limited when presented with riverbeds of complex shapes.

To address various climate risks associated with water and energy, it is necessary to generate more sophisticated computer models, and key to this is data on river cross-section measurements, which enable us to better understand climate risks associated with water flow and hydropower generation. The focus of this project is to curate and build on a large dataset of river measurements gathered by several US state and federal initiatives in recent years, generating one of the largest public datasets for river cross-sections yet. From this, with the help of machine learning techniques, the goal is to be able to better understand riverbeds across the globe, even in regions where data is scarce. By doing so, we can address various climate risks, such as floods, droughts, energy reliability, and environmental conservation, more effectively.

*** Note that the business context described is an ongoing project (as of January 2024) by Solomon Vimal, the CEO of the Cornell Tech-based startup Geothara, whose mission is to utilize data science to solve challenges in water, energy, and climate security. The Breakthrough Tech group Geothara 1A was among several undergraduate student groups assigned to assist in this project throughout August - December 2023. For the Geothara 1A team in particular, our efforts were focused largely on the data generation portion of the project along with some machine learning experimentation. Our journey and methodologies are detailed below.*

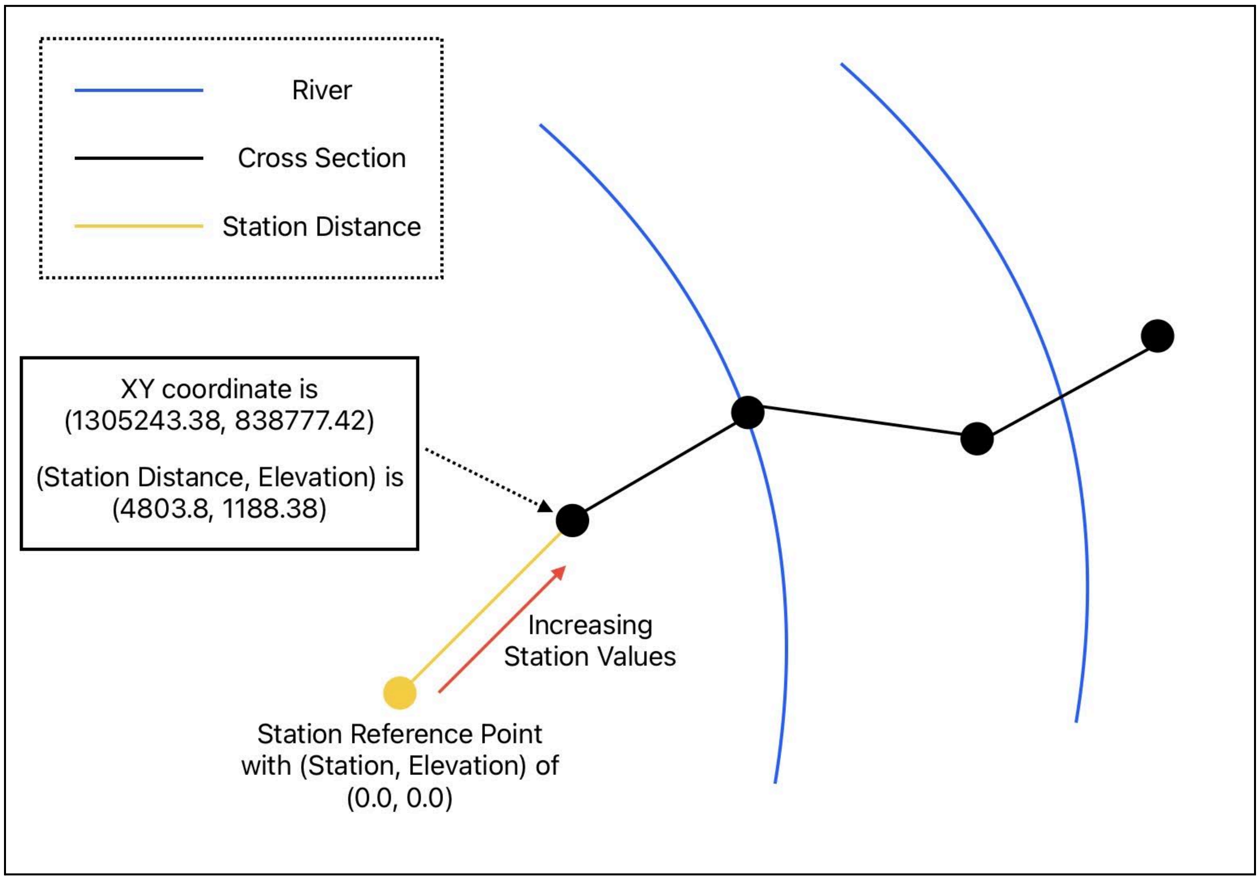
Data

In Computational Fluid Dynamics, HEC-RAS is a popular simulation software that is used to specifically model the hydraulics of water flow through rivers and channels. One of the various file types available are what are known as G01 or geometry files, which contain inside it the geometric data of the river system being analyzed, such as cross-sections, bridges, culverts, and other relevant features. It also contains inside it the spatial layout and characteristics of the river elements required for hydraulic computations, making it an excellent choice for engineers and hydrologists to model and analyze river behavior. Let us take a look at an example below.

```
Geom Title=Existing Conditions
Viewing Rectangle=1303004.70,1306211.02,839516.02,838059.17

River Reach=Beaver Creek Tri,Reach-1
Reach XY=204
      1303004.70       838147.75      1303025.53       838151.92
      ... [coordinates continue]
      1305243.38       838777.42      1305422.44       838433.55

Type RM Length L Ch R = 1,1299,461,461,461
BEGIN DESCRIPTION:
Beaver Creek Tributary 3 4.0
END DESCRIPTION:
XS GIS Cut Line=2
      1305243.38       838777.42      1305422.44       838433.55
#Sta/Elev=33
  4803.8 1188.38  4809.2 1188.11  4826.8 1186.50  ... [coordinates
continue]
  5087.1 1168.90  5104.0 1172.80  5136.4 1182.09  5151.6 1183.02
```

```
#Mann=3,-1,0
  4718.1    0.15       0  4995.4    0.05       0  5005.2    0.15
0
Bank Sta=4995.4,5005.2
XS Rating Curve= 0
Exp/Cntr=0.3,0.1

```

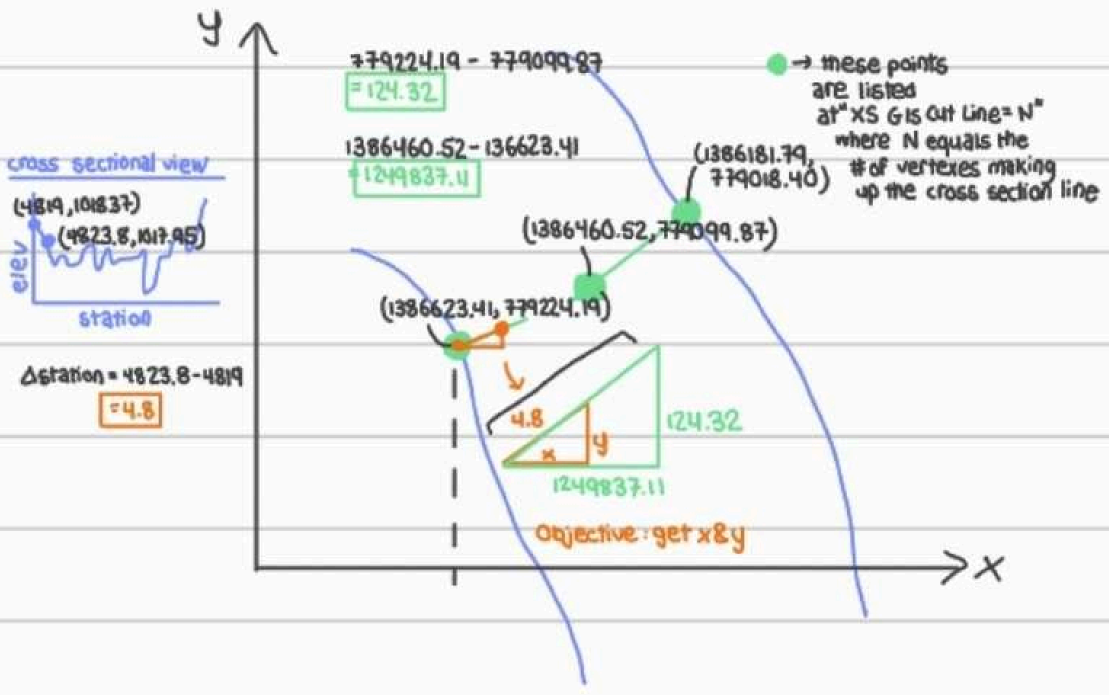
In the above sample G01 file, we have geometric data for a particular river, which contains information about various cross sections that were observed. Notably, for each provided river cross section (denoted ‘Cut Line’ in the file), we have measurements for points belonging to a Cartesian coordinate plane, each of the form (x,y). Taking Cut-Line 2 in the above example, we have two distinct points (1305243.38, 838777.42) and (1305422.44, 838433.55), which denote the endpoints of the cross section. Although there are only two points here, it is worth noting that cuts may consist of any arbitrary number of points (at least 2). Aside from these coordinate pairs, the other notable geometric representation we have available is what is known as Station/Elevation points, but this is best understood with the help of an illustration.


River
Cross Section
Station Distance
XY coordinate is (1305243.38, 838777.42)
(Station Distance, Elevation) is (4803.8, 1188.38)
Increasing Station Values
Station Reference Point with (Station, Elevation) of (0.0, 0.0)

As seen in the figure here, the points in black joined together by straight lines represent a particular cross section for the given river (sketch shown in blue), consisting of 4 points total. Each one of these points can be described by their xy plane coordinates, as shown earlier, but they also may take on an alternate coordinate system denoted by what is described as (station, elevation) pairs. Elevation here is what one might expect, that being the height above a defined level. Station, on the other hand, can be understood as the distance along the cut as measured from some reference point (denoted in the diagram as the yellow dot). Traversing the cross section from the origin then gives larger and larger values for the 'station', each consisting of a measured elevation that has been made available as shown in the sample G01 file. So for example, the provided (4803.8, 1188.38) can be understood as 4803.8 units from the origin, which happens to have an elevation of 1188.38 units. The provided (station, elevation) pairs, however, do not necessarily correspond to the exact points that define the cut, and in most, if not all, our files, for each cut, there is a significantly larger amount of (station, elevation) pairs when compared to the points that define a cut. The challenge then was to determine a way to merge these two coordinate systems to generate a list of (x,y,z) pairs for every cut in a given river, but first let’s consider some data manipulation.

```
from google.colab import files
files.upload()
f = open("Horsepen.g01")
line = ""
f.seek(0,2) #Jumps to the end
endLocation = f.tell()    #Give you the end location (characters from
start)
f.seek(0)
river = []
cross_sections = []
while (f.tell() < endLocation ):
    line = f.readline()
    if line.startswith("Reach XY"):
        while not(line.startswith("Rch Text X Y")):
            line = f.readline()
            if not(line.startswith("Rch Text X Y")):
                vals = [float(x) for x in line.split()]
                #print(vals)
                for i in range(len(vals)//2):
                    river.append([vals[2*i], vals[2*i + 1]])
    if line.startswith("XS GIS Cut Line"):
        cross_section = [[],[]]
        while not(line.startswith("Node") or
line.startswith("#Sta/Elev")):
            line = f.readline()
            if not(line.startswith("Node") or
line.startswith("#Sta/Elev")):
                #print(line)
                vals = [float(x) for x in line.split()]
                #print(vals)
                for i in range(len(vals)//2):
                    cross_section[0].append([vals[2*i], vals[2*i +
1]])
        while not(line.startswith("#Sta/Elev")):
            line = f.readline()
        while not(line.startswith("#Mann")):
            line = f.readline()
            if not(line.startswith("#Mann")):
                vals = [float(x) for x in line.split()]
                #print(vals)
                for i in range(len(vals)//2):
                    cross_section[1].append([vals[2*i], vals[2*i +
1]])
        cross_sections.append(cross_section)
    #print(f.tell())
print(cross_sections)
f.close()
```

Rather than dealing with the G01 file as a whole, it was much more convenient for data analysis to fetch the more relevant parts of the file, and store them in a matrix representation. As such, the above code was designed to take in a particular G01 file, generating a cross_sections list, where cross_sections[i][j] corresponds to the ith cross-section, and for j = 0, it gives a list of (x, y) coordinates, while for j = 1, it provides a list of (station, elevation) pairs. The next step then was to utilize this representation to merge, for every ith cross-section, their j components into a three dimensional representation. The basis for this transformation was made possible through collaboration with team members Vladislav Vostrikov (Hunter College) and Michelle Weng (Cooper Union), the latter of whom was kind enough to make this illustration for understanding.



Step 1: Get hypotenuse of larger triangle

$$\sqrt{1249837.11^2 + 124.32^2}$$

$$= 1249837.11618$$

Step 2: Using triangle ratios, get x & y

y: $$\frac{1249837.11618}{4.8} = \frac{124.32}{y}$$

$$y = 4.7745e\text{-}4$$

x: $$\frac{1249837.11618}{4.8} = \frac{1249837.11}{x}$$

$$x = 4.7999999\ldots \approx 4.8$$

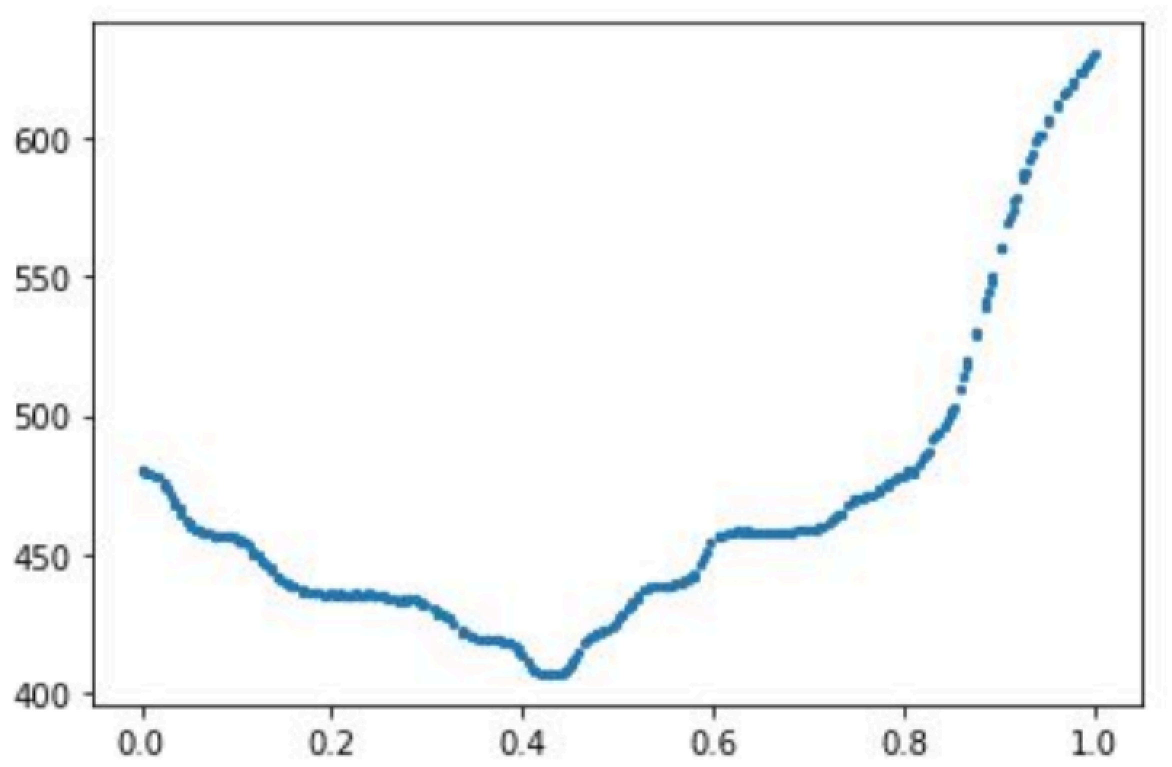
As illustrated in the diagram, for any particular cross-section edge, a triangle can be drawn such that the horizontal and vertical component lengths take on values equivalent to the difference between the y and x components of the points that define that edge. In doing so, a ratio relationship may be established between the larger triangle and smaller triangles formed using station values from the dataset. In general, with consideration of the disjoint edges making up the cut and partitioning the (station, elevation) data for that cut accordingly, the code below utilizes the notion that (x,y,z) coordinates can be computed by evaluating for the x and y values using basic geometry, and the z coordinate is essentially the elevation associated with the station value in consideration.

```
from os import path
import numpy as np
def Task_2(cross_section):
  #input of a list of all cross sections in a river file
  #intput of a index to get a specific cross section
  #resulting list of xyz tuples representing (x,y,elevation)
  xyz = []
  i = 0 #i is used to index the reference station at the beginning of
a line segment
  distance_leftover = 0 #useful when distance travelled does not end
at line segment

  for pt in range(len(cross_section[0])-1):
    st0 = cross_section[1][i][0]
    x2_y2 = cross_section[0][pt+1]
    x1_y1 = cross_section[0][pt]
    hypotenuse = np.sqrt((x2_y2[1]-x1_y1[1])**2 +
(x2_y2[0]-x1_y1[0])**2)
    hypotenuse = round(hypotenuse, 2)
    # print (f"Commence New Line Segment: {x2_y2}, {x1_y1}" )
    # print (f"Length of Line Segment: {hypotenuse}\n")
    for st in range(i, len(cross_section[1])-1):
      # print (f"STATION {st+1}") #Station 1, ..., Station N
      st_elev = cross_section[1][st]
      # print (f"Station, Elev: {st_elev}")
      dist = st_elev[0]-st0 + distance_leftover #distance travelled
along cross section
      # print (f"Distance Travelled Along Cross Section: {dist}")
      if dist <= hypotenuse:
        x0 = x1_y1[0]
        y0 = x1_y1[1]
        x = x0 + dist/hypotenuse*(x2_y2[0]-x1_y1[0])
        x = round(x, 2)
        y = y0 + dist/hypotenuse*(x2_y2[1]-x1_y1[1])
        y = round(y, 2)
        #if x2_y2[0] > x1_y1[0]:
          #x = x0 + dist/hypotenuse*(x2_y2[0]-x1_y1[0])
        #if x2_y2[0] < x1_y1[0]:
          #x = x0 - dist/hypotenuse*(x2_y2[0]-x1_y1[0])
        #if x2_y2[1] > x1_y1[1]:
          #y = y0 + dist/hypotenuse*(x2_y2[1]-x1_y1[1])
        #if x2_y2[1] < x1_y1[1]:
          #y = y0 - dist/hypotenuse*(x2_y2[1]-x1_y1[1])

z = st_elev[1]
        # print(f"xyz = ({x}, {y}, {z})")
        # print("\n")
        xyz.append((x,y,z))
      else:
        i = st
        distance_leftover = dist-hypotenuse
        # print(f"Distance Left Over (carry over to next line segment
if it exists) = {distance_leftover}\n")
        break
  xF_yF = cross_section[0][-1]
  xyz.append((round(xF_yF[0], 2), round(xF_yF[1], 2),
round(cross_section[1][-1][1], 2)))
  return xyz
```

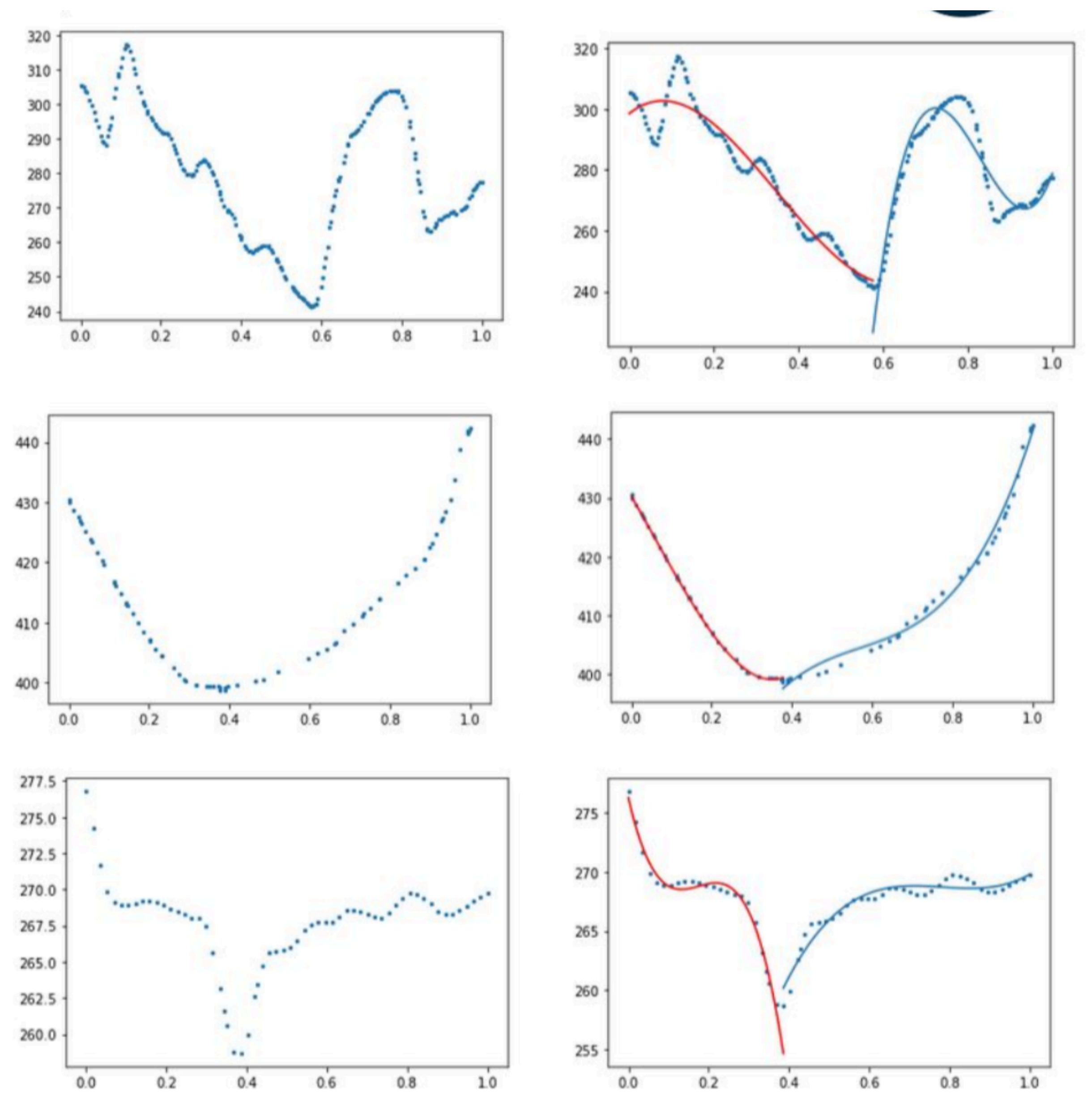
With consideration for the analysis above, the code takes in as input a particular river cross section (defined as a list with index 0 giving the cut (x,y) coordinates and index 1 giving the associated (station, elevation) pairs as described earlier). The function returns as output a list of (x,y,z) coordinates associated with that particular cut line. The function code can then be run intuitively for each cross section provided for a given river.

Approach

Having generated a uniform data representation for all the G01 files, the next phase of the project was led largely by Carla Sanchez (Hunter College) in collaboration with the full Geothara 1A team. To simplify our data generation problem, instead of dealing with a third-dimensional model, the data generated was modified to extract (from the (x,y,z) coordinates list for each cut) solely the x and z coordinates, resulting in plots looking like the one below, with the horizontal axis depicting the x coordinates after normalization and the vertical axis the respective elevation.



One of the motivations for the team early on was to create a prototype for a more sophisticated river cross-section model. Given the above sample scatterplot, with the horizontal representing position and the vertical the respective elevation, a regression approach to the data modeling problem seemed most appropriate. After some experimentation with different-order polynomials, higher-degree polynomials consistently performed poorly when it came to generalization, an indication of overfitting. On the other hand, second-degree polynomials seemed to underfit the data. Provided these constraints, a solution was proposed, where for each cross section, a combination of two third-order polynomials (examples shown below) was selected for the modeling, which seemed to perform reasonably well.



Lessons Learned

By the end of the four months, the team was successful in generating a dataset that the Geothara team can build off of for enhanced geospatial analyses and modeling. Throughout the project, however, many of our team members were faced with the reality of how meticulous and labor-intensive the river data cleaning process was, provided that there were many different ways of doing the job. Furthermore, although our analyses are justified in many ways, one of the important things we have realized is that it was only through the various assumptions and simplifications we took on during the process that we were able to produce such a model. Even so, we cannot ignore the fact that our Challenge Advisor

(Geothara CEO Solomon Vimal) has a vast team of talented PhD students working very hard on this project, which illustrates how complex problems in water, energy, and climate security can be. So although we didn't care for many of the nuances brought up in fields like hydrology, geomorphology, and civil engineering, the data we generated is a good starting point for further analysis by the team.

Acknowledgements

This project's success was made possible through the invaluable collaboration and support of key stakeholders. We extend our deepest appreciation to Break Through Tech for fostering a partnership with the dynamic Cornell Tech startup, Geothara. Our gratitude is extended to Erika Bramwell, whose exemplary guidance and leadership in the AI Studio program were instrumental in shaping our journey. A sincere thank you to Break Through Tech staff for their unwavering support throughout the project.

Special acknowledgment goes to Geothara CEO Solomon Vimal, whose role as a challenge advisor and mentor significantly contributed to the project's success. We are indebted to Teaching Assistant Jackie Kim for her steadfast support. This project owes its accomplishments to the exceptional dedication and collaboration of the entire Geothara 1A team—Vladislav Vostrikov, Michelle Weng, Carla Sanchez, Sabreen Shigri, and Ryan Chen. Their hard work and commitment were integral to realizing the project's objectives. This collaborative effort exemplifies how the convergence of talent and expertise can bring about transformative outcomes.