Jake Berberian
Numerical Analysis: Basic Prob
18 September 2020

Project One Writeup

Plots

In our plots, we see that my speed was relatively constant, while my elevation changed a little, but nothing out of the ordinary in the duration (over 12 feet). During this period of time, I was walking back from my friend's house. He lives about a mile away. I would say that the plots do a decent job of showing this— my speed definitely changed a little because it was later at night and I wanted to get home quicker, but for the most part, my velocity was constant.

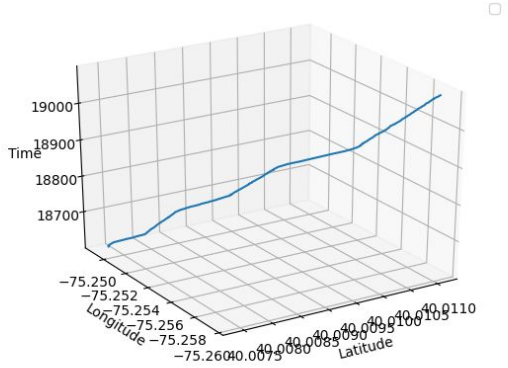

Figure 1: Plot of coordinates over time.

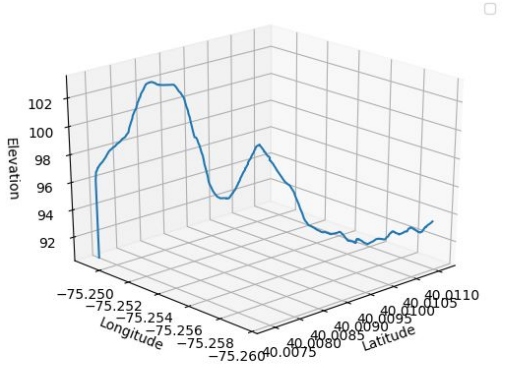

Figure 2: Plot of coordinates by elevation.

Overview

After tracking my coordinates and speed, we calculated both the distance between each latitude and longitude coordinate pairs. This results in a delta between every time the cell phone was tracked. However, since the Earth is an oblate spheroid, we need to use the haversine formula[1] to calculate this. We'll calculate this change in both miles and feet. Using this step, we'll calculate our speed, in miles per hour, between two track points. Finally, we take these outputs and write this to a text file. This will simulate what a GPX converter does, reading a GPX-formatted file and outputting them to another file type with the necessary components (distance traveled, speed, etc.) that's more widely used for data (.csv or .txt).

[1] https://www.movable-type.co.uk/scripts/latlong.html

Outputs

Below are some of the outputs from the project.

Figure 3: GPX file visualized on a map[2].

Figure 4: GPX file visualized on Google Earth[3].

Figure 5[4]: GPS file visualized on Google Earth with destinations marked.

	A	B	C	D	E	F	G
1	0	40.007404576834	-75.2490820742918	296.911797300201	-4.10179711314684	0.550428130329784	2906.1960139823
2	1	40.007404868815	-75.2490575327522	315.504576799402	4.10557063940662	0.011404362887 24	6.02136993707951
3	2	40.007411660775 8	-75.2490707813225	317.102983514581	2.78380231674908	0.000773278421319	4.08281943074005
4	3	40.007407372268 2	-75.2490894996995	317.40365635 57	3.0345010158227	0.000918779474235	4.85104793636123
5	4	40.007404377723	-75.2491062008961	317.632699907588	2.89170455221227	0.000803251264503	4.24107252971867
6	5	40.007412083054 5	-75.2491277971798	317.833134121618	4.09037084720417	0.001136214124223	5.99907740338414
7	6	40.007412552316 9	-75.2491528292525	318.082746482368	4.18855505569801	0.001163487515472	6.14307771253634
8	7	40.007415420645 2	-75.2491753743689	318.373794125104	3.83786077034499	0.001066072436207	5.62873751177061
9	8	40.007419833045 4	-75.2491776941579	318.642548992589	1.16410450115224	0.000323362361431	1.70731536794849
10	9	40.007422913465 1	-75.2491992655677	318.694081091291	3.68858490985148	0.001024606919403	5.40980444310534
11	10	40.007426585265 9	-75.2492256721264	318.928544171492	4.51032799518388	0.001252868887551	6.61500088097217
12	11	40.007437020081 2	-75.2492401407912	319.232686125874	3.54877613727846	0.000985771149244	5.20475612849813
13	12	40.007449019765 3	-75.2492490082446	319.416897205559	3.33301050896321	0.00092583625249	4.88830689843928
14	13	40.007459279810 7	-75.2492649435467	319.52958564356	3.69025284798187	0.001025070235551	5.41225069805924
15	14	40.007458818995 9	-75.249285333312	319.710240568018	3.40913599651263	0.000947929203475	5.0049550901873
16	15	40.007474096443 7	-75.2493202239002	319.945045451608	6.97133526472094	0.00193454553596	10.2141736871347
17	16	40.007490707584 9	-75.2493502214396	320.339411395335	6.49357516573661	0.001805574650251	9.53322252691392
18	17	40.007503879855 8	-75.2493876802151	320.697716892311	7.07095204872625	0.001964153346868	10.3704994584832
19	18	40.00751816086 4	-75.2494219100746	321.140161893775	6.73835546947872	0.001871765408189	9.88270197078 83
20	19	40.007523747810 4	-75.2494603580492	321.555505862824	6.58009764474737	0.001827804901319	9.65059564700509
21	20	40.007534384624 3	-75.2495012247701	321.9756076746	7.33047599182721	0.002036243331063	10.7511261255636
22	21	40.007545947319 9	-75.2495316286985	322.391792361256	5.8430180893053	0.001623060580363	8.56056962987236
23	22	40.007557486934 5	-75.2495611096699	322.678017036037	5.70628980206605	0.001585080500574	8.36903926012825
24	23	40.007571672342 1	-75.2495941661675	322.916665627996	6.55983102554878	0.001822175284875	9.62087193201227
25	24	40.007581139649 6	-75.249626240157	323.137918549265	5.85975431640706	0.001627709532335	8.59411555139747
26	25	40.007592559446	-75.2496583931927	323.347159253806	6.08292489747736	0.00168970136041	8.92142513774318
27	26	40.007602871272 2	-75.2496905687674	323.516518606274	5.96279019615799	0.001656330610044	8.74523148710148
28	27	40.007612021495 5	-75.2497269623847	323.664385769181	6.50009634454061	0.001805582317928	9.53326301134939

Figure 6: File output from Python3 code.

Conclusions

Overall, we were able to map our GPS coordinates over Google Earth and GPSVisualizer (above). Additionally, we were able to create somewhat of "converter" code, which takes a GPX-formatted file and converts it into a more accessible .csv-formatted file. This creates a program similar to GPSVisualizer that allows you to do likewise.

[2] Source: https://www.gpsvisualizer.com/
[3] Source: https://www.earth.google.com/
[4] Source: https://www.earth.google.com/