Notebook for shallow water bathymetry with laser satellite

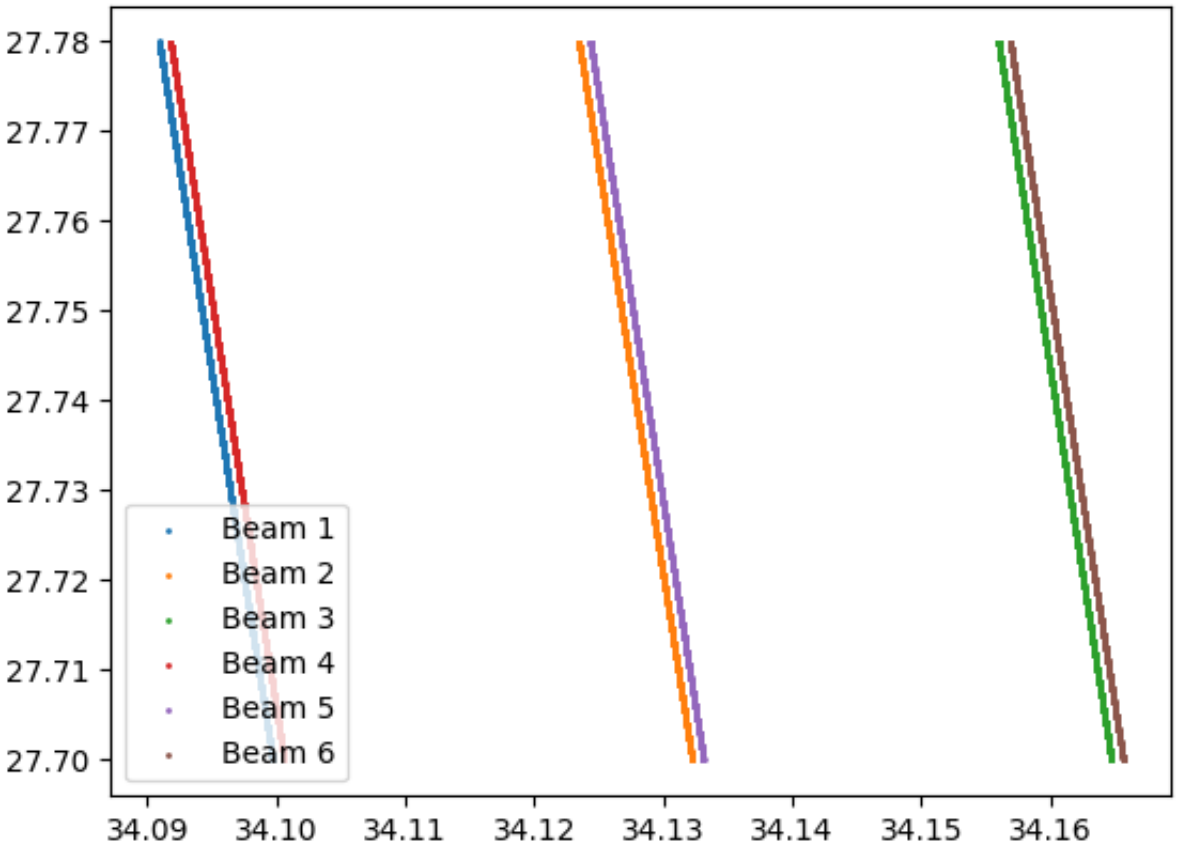
First we load in the data we need, and plot it to inspect the values

In [49]:
```
# Import the nessesary python libraries
import numpy as np
import matplotlib.pyplot as plt
```

In [68]:
```
# Load in the .txt file. We tell python its a comma separated file
data = np.loadtxt("icesat.txt", delimiter=",")

# Extract the values from the columns
lat = data[:,0]
lon = data[:,1]
depth = data[:,2]
beam_id = data[:,3]
```

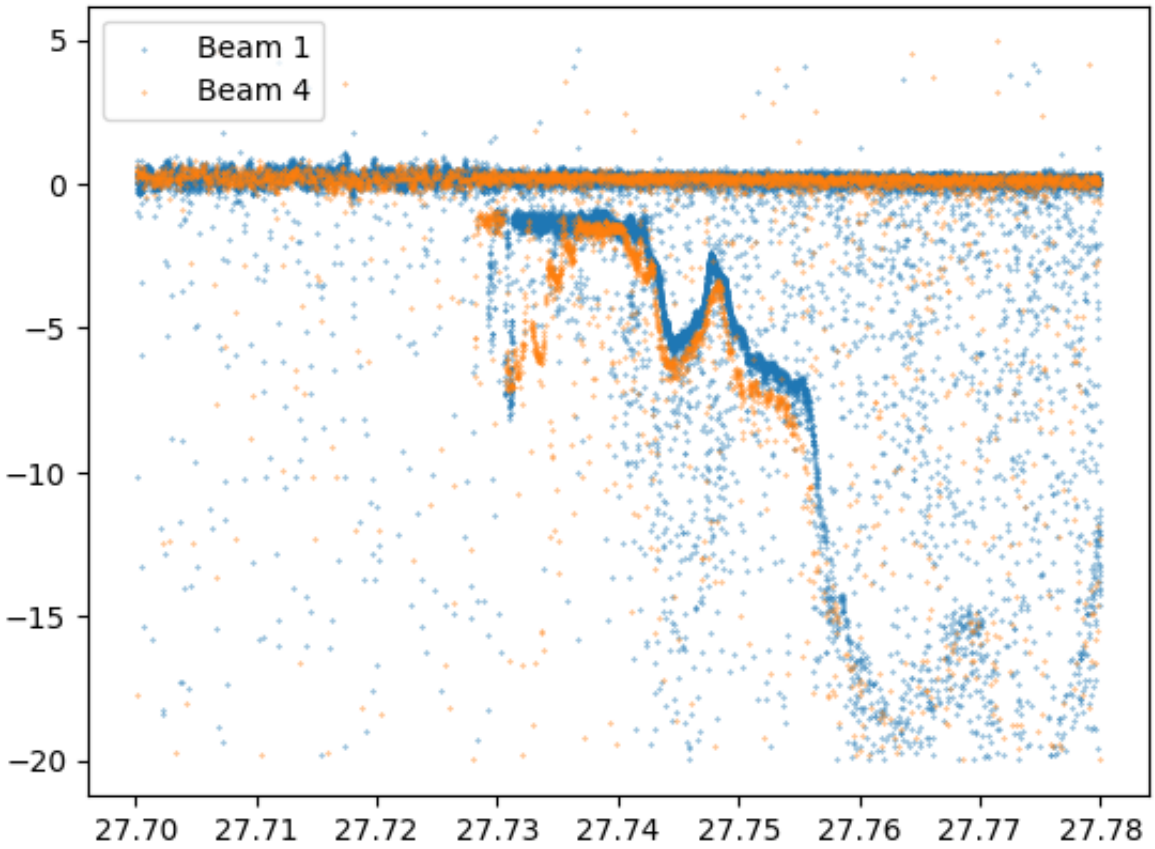
In [69]:
```
# Plot each beam id number to see the spatial location
for i in range(1, 7):
    mask = beam_id == i
    plt.scatter(lon[mask], lat[mask], 1, label="Beam "+str(i))
plt.legend()
plt.show()
```



In [70]:
```
# Inspect the beams furthest to the west
mask = beam_id == 1
```

```
plt.scatter(data[mask,0], data[mask,2], 0.1, label="Beam 1")
mask = beam_id == 4
plt.scatter(data[mask,0], data[mask,2], 0.1, label="Beam 4")
plt.legend()
```

Out[70]: <matplotlib.legend.Legend at 0x11aa125d0>



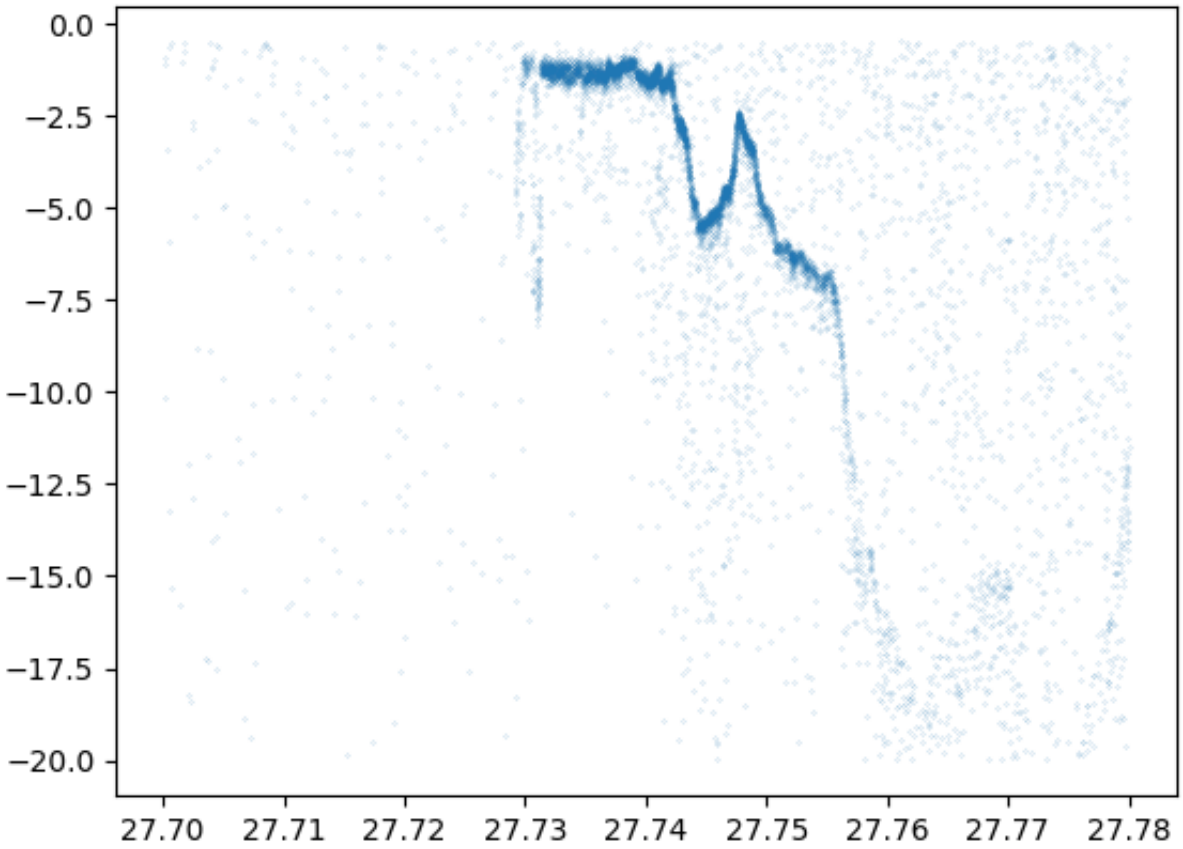
Determine the bathymetry

We will determine the bathymetry from beam 1. We will therefore extract the values below -0.5 meters.

In [71]:
```
# Get data from beam 1
beam_n = 1
# Make datamask where we select data from beam 1 AND depth below -0.5 m
mask = (beam_id == beam_n) & (depth<-0.5)
# Extract depth
depth = depth[mask]
```

In [ ]:
```
# Plot the data
plt.scatter(lat[mask], depth, 0.01)
```

Out[ ]: <matplotlib.collections.PathCollection at 0x11aa8f4d0>



Correct for refraction

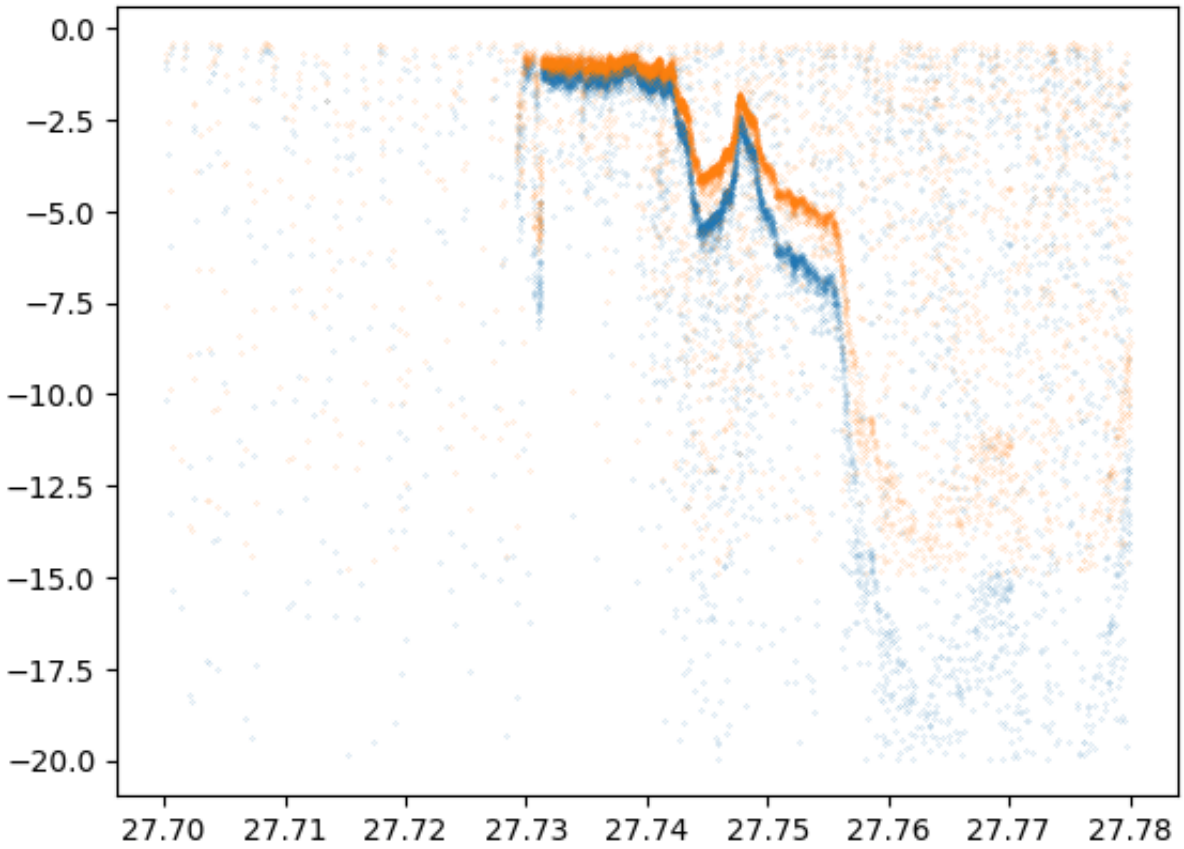
We need to adjust the bathymetry for refraction. A simple correction for refraction is given as Z′ = Z+0.25416*D, where Z′ is the corrected elevation, Z is the uncorrected elevation, and D is the depth

In [105…

```
# Correct for refraction
bath = depth - 0.25416*depth
plt.scatter(data[mask,0], depth, 0.01)
plt.scatter(data[mask,0], bath, 0.01)
```

Out[105…

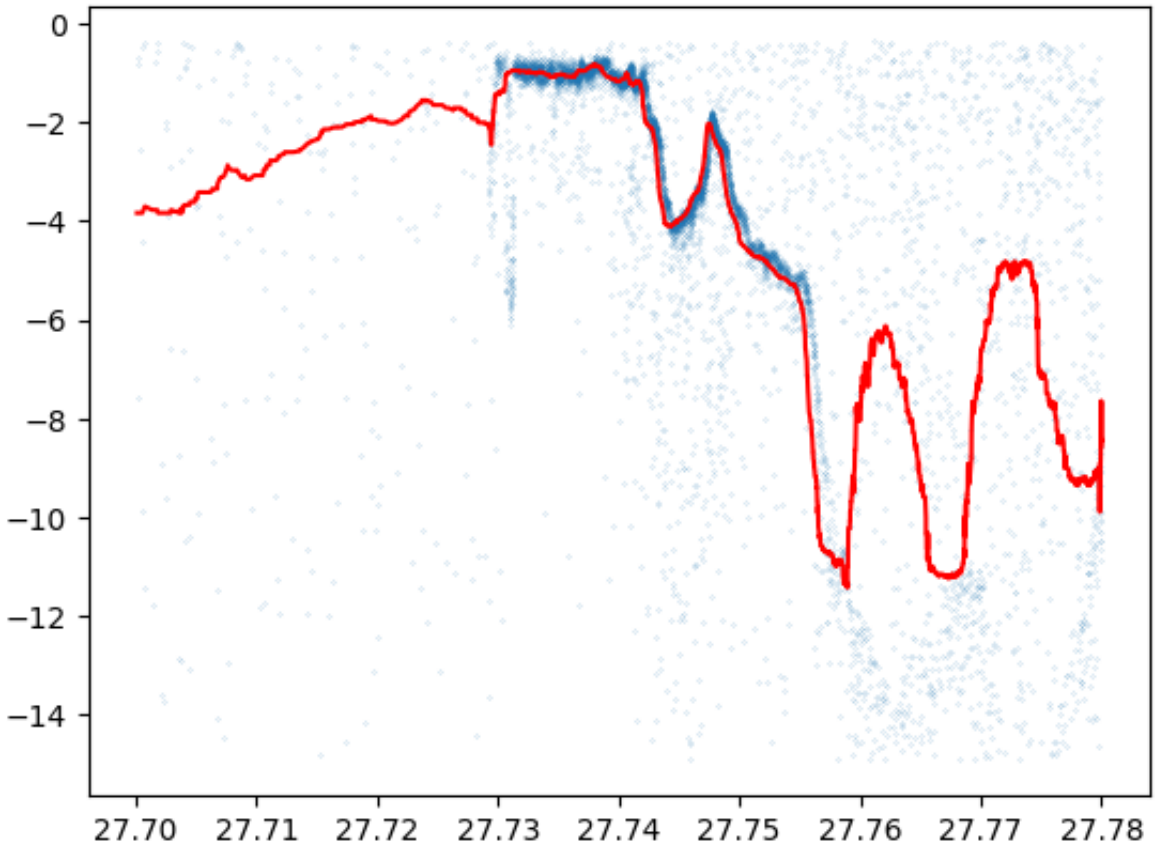
```
<matplotlib.collections.PathCollection at 0x11b21c190>
```



In [104…

```
# Compute the running median to determine the bathymetry
median_width = 200
running_median = []
for i in range(lat[mask].size):
    running_median.append(np.median(bath[i:i+median_width]))

plt.scatter(lat[mask], bath, 0.01)
plt.plot(lat[mask], running_median, "r")
plt.show()
```


0
−2
−4
−6
−8
−10
−12
−14
27.70
27.71
27.72
27.73
27.74
27.75
27.76
27.77
27.78