.ipynb

July 16, 2019

1 Part 1: how to use the AFS algorithm

[1]:
```
import os
# set your working directory to the location where you downloaded the repository
os.chdir("/Users/jiguangli/Blaze_Function")
```

[3]:
```
# load essential packages, make sure you have downloaded the repository
import pandas as pd
from AFS import *
import matplotlib.pyplot as plt

# read the csv file as pandas dataframe
data= pd.read_csv('ExampleSpectrum.csv', sep=',')
print(data)

# Visualize the input spectrum
plt.clf()
plt.figure(figsize=(10,5))
plt.plot(data["wv"], data['intens'], 'brown', linewidth=1)
plt.title("Input Spectrum")
```

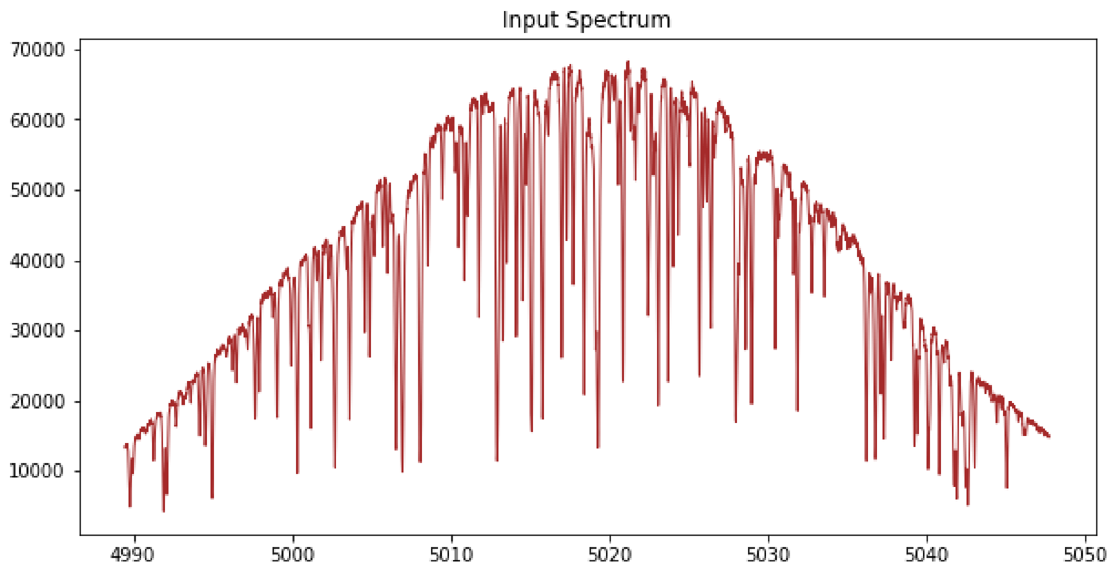
```
                wv        intens
0      4989.459023  13275.410160
1      4989.470168  13555.553710
2      4989.481313  13590.049800
3      4989.492456  13561.420900
4      4989.503600  13615.585940
5      4989.514742  13681.700200
6      4989.525884  13517.300780
7      4989.537025  13829.336910
8      4989.548166  13291.321290
9      4989.559306  13492.753910
10     4989.570445  13673.135740
11     4989.581584  13606.767580
12     4989.592723  13769.799800
13     4989.603860  13808.119140
```

14	4989.614997	13613.567380
15	4989.626133	13411.534180
16	4989.637269	13344.519530
17	4989.648404	13188.013670
18	4989.659539	13041.805660
19	4989.670673	12914.448240
20	4989.681806	12221.254880
21	4989.692939	12096.913090
22	4989.704071	11186.958980
23	4989.715202	10589.657230
24	4989.726333	9549.683594
25	4989.737463	8359.782227
26	4989.748593	7440.698242
27	4989.759722	6332.926758
28	4989.770850	5787.969238
29	4989.781978	5055.215332
...	...	...
6170	5047.488092	15108.690430
6171	5047.495870	15130.281250
6172	5047.503647	15248.315430
6173	5047.511424	15288.139650
6174	5047.519201	15414.486330
6175	5047.526977	15346.845700
6176	5047.534752	15283.971680
6177	5047.542527	15107.726560
6178	5047.550302	15496.790040
6179	5047.558076	15183.424800
6180	5047.565850	15263.188480
6181	5047.573623	15358.850590
6182	5047.581395	15152.034180
6183	5047.589167	15053.594730
6184	5047.596939	15148.112300
6185	5047.604710	15376.279300
6186	5047.612480	15004.496090
6187	5047.620250	15328.253910
6188	5047.628020	15225.638670
6189	5047.635789	15043.571290
6190	5047.643557	15286.645510
6191	5047.651325	15104.936520
6192	5047.659093	15168.670900
6193	5047.666860	15137.547850
6194	5047.674626	14937.758790
6195	5047.682392	14821.161130
6196	5047.690158	14790.353520
6197	5047.697923	14949.046880
6198	5047.705687	14762.092770
6199	5047.713451	14897.256840

```
[6200 rows x 2 columns]
```

[3]: Text(0.5, 1.0, 'Input Spectrum')

```
<Figure size 432x288 with 0 Axes>
```



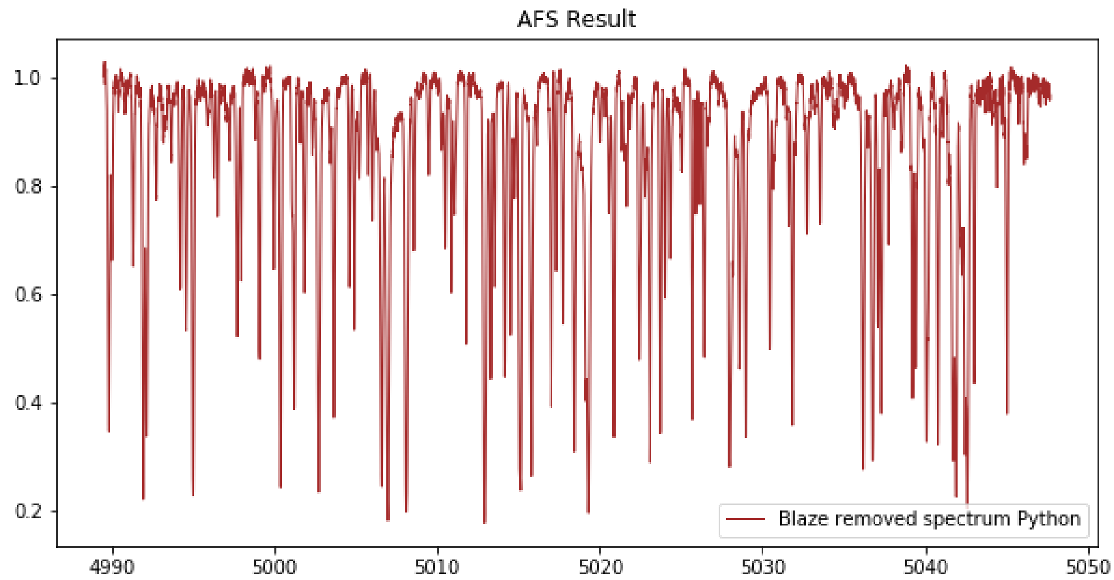
[4]:
```
# result is a 1-dimensional vector recording the blaze-removed spectrum␣
↪(normalized density)
result= AFS(data,0.95,0.25)
print(result)
```

```
[1.00152283 1.02063311 1.02120869 ... 0.96906359 0.95767683 0.96718523]
```

[5]:
```
# If you want to plot the blaze-removed spectrum
plt.clf()
plt.figure(figsize=(10,5))
plt.plot(data["wv"], result, 'brown', linewidth=1, label='Blaze removed spectrum␣
↪Python')
plt.legend(loc='lower right')
plt.title("AFS Result")
```

[5]: Text(0.5, 1.0, 'AFS Result')

```
<Figure size 432x288 with 0 Axes>
```



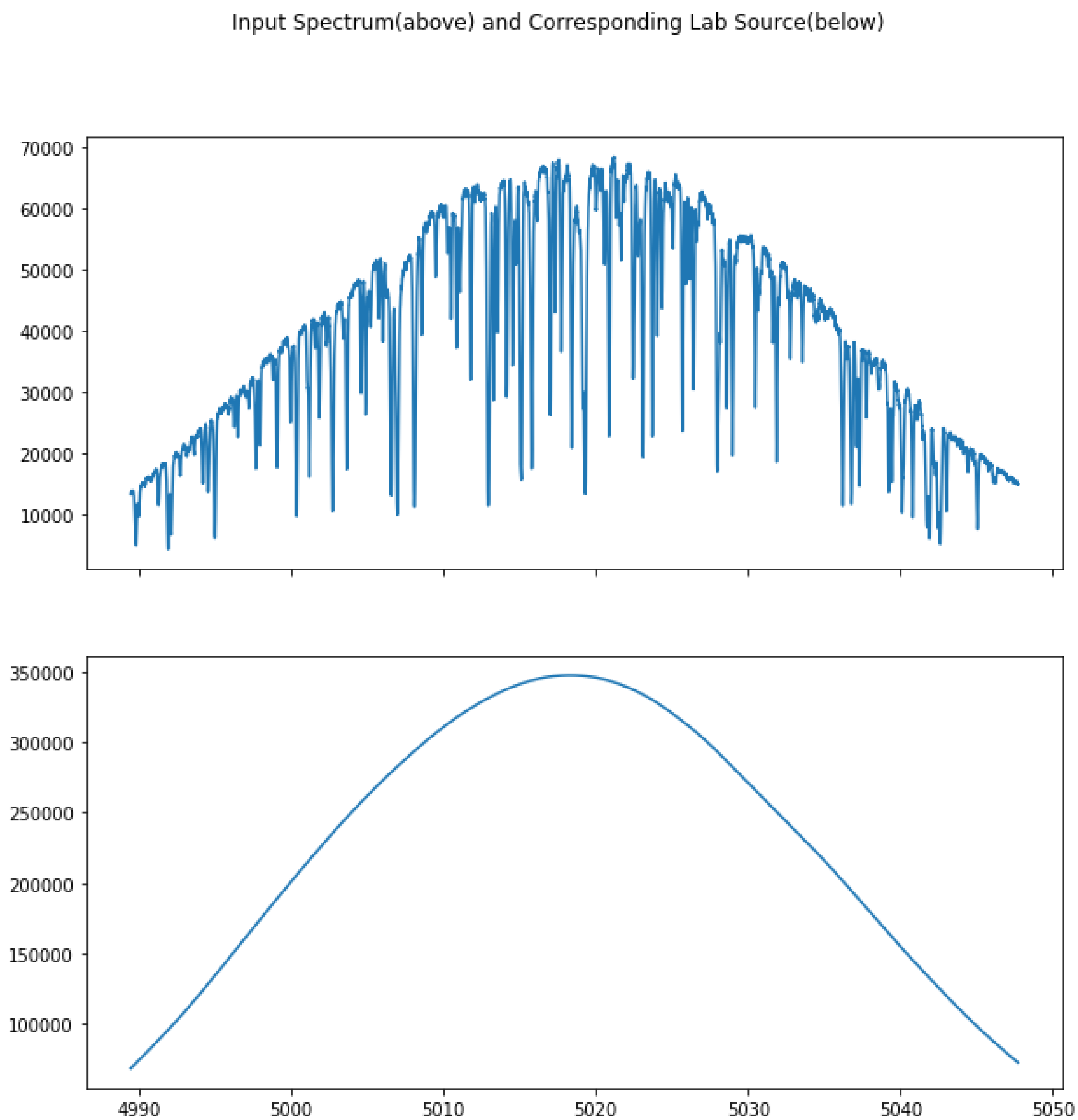
2 Part 2: how to use the ALSFS algorithm

[6]:
```
# import the ALSFS Function
from ALSFS import *

# read the input csv files as a pandas dataframe
data= pd.read_csv('ExampleSpectrum.csv', sep=',')
source= pd.read_csv('LabSource.csv', sep=',')


# Visualize Input Spectrum and Lab Source Spectrym
fig, (ax1, ax2) = plt.subplots(2, sharex=True, figsize=(10,10))
fig.suptitle('Input Spectrum(above) and Corresponding Lab Source(below)')
ax1.plot(data["wv"], data['intens'])
ax2.plot(source["wv"], source['intens'])
```

[6]:
```
[<matplotlib.lines.Line2D at 0x81856a160>]
```



[7]:
```
# Run the ALSFS Algorithm, result is a one dimensional vector recording the␣
↪blaze-removed spectrum
result= ALSFS(data,source,0.95,0.25)
print(result)
```

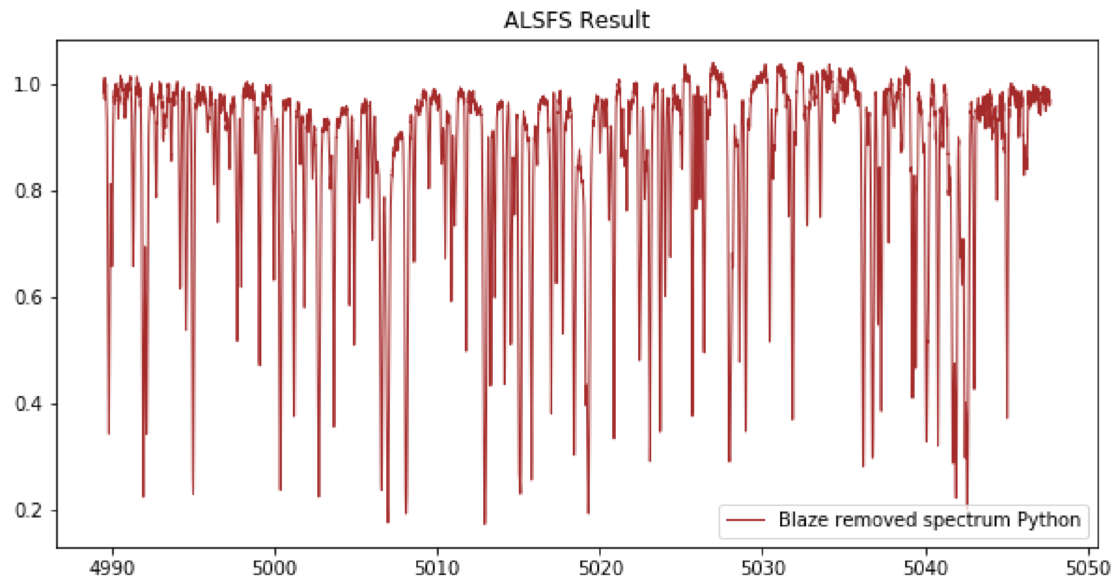
```
[0.98300525 1.0020198  1.0028414  ... 0.97167375 0.96036808 0.9700164 ]
```

[8]:
```
#If you want to plot the blaze-removed spectrum
plt.clf()
plt.figure(figsize=(10,5))
plt.plot(data["wv"], result, 'brown', linewidth=1, label='Blaze removed spectrum␣
↪Python')
```

```
plt.legend(loc='lower right')
plt.title("ALSFS Result")
```

[8]: Text(0.5, 1.0, 'ALSFS Result')

<Figure size 432x288 with 0 Axes>



3 Part 3: How to Use Boundary_Correction Algorithm

[9]:
```
from Boundary_Correction import*
```

[10]:
```
# change left order to a dataframe
left_data= pd.read_csv('left_order.csv', sep=',')
print(left_data)
```

```
           wv   removed
0  4614.33849  0.981654
1  4614.35024  0.971405
2  4614.36199  0.965729
3  4614.37373  0.968820
4  4614.38548  0.983130
5  4614.39723  0.983517
6  4614.40898  0.975318
7  4614.42073  0.976888
8  4614.43248  0.982651
9  4614.44423  0.980175
```

10	4614.45597	0.982718
11	4614.46772	0.951651
12	4614.47947	0.953646
13	4614.49122	0.924437
14	4614.50297	0.926782
15	4614.51472	0.926670
16	4614.52647	0.913938
17	4614.53822	0.929724
18	4614.54997	0.910702
19	4614.56172	0.909433
20	4614.57347	0.913216
21	4614.58522	0.919419
22	4614.59697	0.923979
23	4614.60872	0.958884
24	4614.62047	0.962657
25	4614.63222	0.960293
26	4614.64397	0.961711
27	4614.65572	0.973147
28	4614.66747	0.972136
29	4614.67922	0.958981
...	...	...
4279	4665.16342	0.914919
4280	4665.17543	0.903363
4281	4665.18744	0.920106
4282	4665.19945	0.921641
4283	4665.21146	0.934475
4284	4665.22347	0.955175
4285	4665.23548	0.953388
4286	4665.24749	0.955434
4287	4665.25949	0.957431
4288	4665.27150	0.967732
4289	4665.28351	0.987294
4290	4665.29552	0.984599
4291	4665.30753	0.985681
4292	4665.31954	0.994476
4293	4665.33155	0.990601
4294	4665.34356	1.001491
4295	4665.35557	0.994554
4296	4665.36758	0.991368
4297	4665.37959	0.998752
4298	4665.39160	0.993433
4299	4665.40361	1.001060
4300	4665.41562	1.001152
4301	4665.42763	0.985247
4302	4665.43964	0.983737
4303	4665.45165	0.993622
4304	4665.46366	1.004652
4305	4665.47567	0.980617

```
4306  4665.48768  0.983943
4307  4665.49969  0.970095
4308  4665.51170  0.956747

[4309 rows x 2 columns]
```

[12]:
```python
# change right order to a data frame
right_data= pd.read_csv('right_order.csv', sep=',')
print(right_data)
```

```
              wv   removed
0     4649.61625  0.881196
1     4649.62818  0.902400
2     4649.64011  0.872820
3     4649.65204  0.874761
4     4649.66397  0.886186
5     4649.67590  0.908613
6     4649.68783  0.907275
7     4649.69976  0.933970
8     4649.71169  0.939637
9     4649.72361  0.949615
10    4649.73554  0.957694
11    4649.74747  0.953862
12    4649.75940  0.930728
13    4649.77133  0.920842
14    4649.78326  0.865157
15    4649.79519  0.846730
16    4649.80712  0.823478
17    4649.81905  0.806060
18    4649.83098  0.812489
19    4649.84291  0.857787
20    4649.85484  0.933896
21    4649.86677  0.928635
22    4649.87870  0.980403
23    4649.89063  0.997962
24    4649.90256  1.021045
25    4649.91449  1.004422
26    4649.92642  1.009026
27    4649.93835  1.011504
28    4649.95028  1.002443
29    4649.96221  0.978740
...          ...       ...
4244  4700.79885  1.004832
4245  4700.81105  0.992999
4246  4700.82324  0.986981
4247  4700.83543  0.982247
4248  4700.84763  0.974182
```

```
4249  4700.85982  0.978199
4250  4700.87201  0.969992
4251  4700.88421  0.960515
4252  4700.89640  0.957301
4253  4700.90859  0.952892
4254  4700.92079  0.946925
4255  4700.93298  0.958879
4256  4700.94517  0.963840
4257  4700.95737  0.964340
4258  4700.96956  0.945322
4259  4700.98175  0.911974
4260  4700.99395  0.851625
4261  4701.00614  0.787175
4262  4701.01833  0.689637
4263  4701.03053  0.604627
4264  4701.04272  0.559797
4265  4701.05492  0.560852
4266  4701.06711  0.598490
4267  4701.07930  0.675669
4268  4701.09150  0.751273
4269  4701.10369  0.811960
4270  4701.11589  0.841055
4271  4701.12808  0.850696
4272  4701.14028  0.869019
4273  4701.15247  0.880296

[4274 rows x 2 columns]
```

[13]:
```
# run Boundary_correction
# result is a two-element tuple (corrected_order1,corrected_order 2),␣
↪representing the corrected version of the left order and the right order.
result= Boundary_correction(left_data,right_data)
print(result[0])
print(result[1])
```

```
              wv    intens
0     4614.33849  0.981654
1     4614.35024  0.971405
2     4614.36199  0.965729
3     4614.37373  0.968820
4     4614.38548  0.983130
5     4614.39723  0.983517
6     4614.40898  0.975318
7     4614.42073  0.976888
8     4614.43248  0.982651
9     4614.44423  0.980175
10    4614.45597  0.982718
```

11	4614.46772	0.951651
12	4614.47947	0.953646
13	4614.49122	0.924437
14	4614.50297	0.926782
15	4614.51472	0.926670
16	4614.52647	0.913938
17	4614.53822	0.929724
18	4614.54997	0.910702
19	4614.56172	0.909433
20	4614.57347	0.913216
21	4614.58522	0.919419
22	4614.59697	0.923979
23	4614.60872	0.958884
24	4614.62047	0.962657
25	4614.63222	0.960293
26	4614.64397	0.961711
27	4614.65572	0.973147
28	4614.66747	0.972136
29	4614.67922	0.958981
...	...	...
4279	4665.16342	0.916278
4280	4665.17543	0.914989
4281	4665.18744	0.922117
4282	4665.19945	0.925803
4283	4665.21146	0.933855
4284	4665.22347	0.943869
4285	4665.23548	0.946596
4286	4665.24749	0.954120
4287	4665.25949	0.960675
4288	4665.27150	0.964927
4289	4665.28351	0.978639
4290	4665.29552	0.987325
4291	4665.30753	0.993651
4292	4665.31954	0.995315
4293	4665.33155	0.996908
4294	4665.34356	0.996552
4295	4665.35557	0.999316
4296	4665.36758	0.996100
4297	4665.37959	1.002575
4298	4665.39160	1.007226
4299	4665.40361	0.997760
4300	4665.41562	1.005323
4301	4665.42763	1.001552
4302	4665.43964	0.991090
4303	4665.45165	0.993320
4304	4665.46366	0.991856
4305	4665.47567	0.996148
4306	4665.48768	0.986833

```
4307  4665.49969  0.975771
4308  4665.51170  0.964930

[4309 rows x 2 columns]
              wv    intens
0     4649.61625  0.841633
1     4649.62818  0.838301
2     4649.64011  0.835155
3     4649.65204  0.840271
4     4649.66397  0.849046
5     4649.67590  0.856683
6     4649.68783  0.870652
7     4649.69976  0.883986
8     4649.71169  0.893217
9     4649.72361  0.905011
10    4649.73554  0.911919
11    4649.74747  0.914028
12    4649.75940  0.905797
13    4649.77133  0.882099
14    4649.78326  0.844571
15    4649.79519  0.807494
16    4649.80712  0.774981
17    4649.81905  0.763428
18    4649.83098  0.783893
19    4649.84291  0.822640
20    4649.85484  0.870485
21    4649.86677  0.907679
22    4649.87870  0.941860
23    4649.89063  0.961260
24    4649.90256  0.966617
25    4649.91449  0.974139
26    4649.92642  0.971856
27    4649.93835  0.969387
28    4649.95028  0.953604
29    4649.96221  0.931749
...          ...       ...
4244  4700.79885  1.004832
4245  4700.81105  0.992999
4246  4700.82324  0.986981
4247  4700.83543  0.982247
4248  4700.84763  0.974182
4249  4700.85982  0.978199
4250  4700.87201  0.969992
4251  4700.88421  0.960515
4252  4700.89640  0.957301
4253  4700.90859  0.952892
4254  4700.92079  0.946925
4255  4700.93298  0.958879
```

```
4256  4700.94517  0.963840
4257  4700.95737  0.964340
4258  4700.96956  0.945322
4259  4700.98175  0.911974
4260  4700.99395  0.851625
4261  4701.00614  0.787175
4262  4701.01833  0.689637
4263  4701.03053  0.604627
4264  4701.04272  0.559797
4265  4701.05492  0.560852
4266  4701.06711  0.598490
4267  4701.07930  0.675669
4268  4701.09150  0.751273
4269  4701.10369  0.811960
4270  4701.11589  0.841055
4271  4701.12808  0.850696
4272  4701.14028  0.869019
4273  4701.15247  0.880296

[4274 rows x 2 columns]
```

4 Part 4: How to Use LS_Smoothing

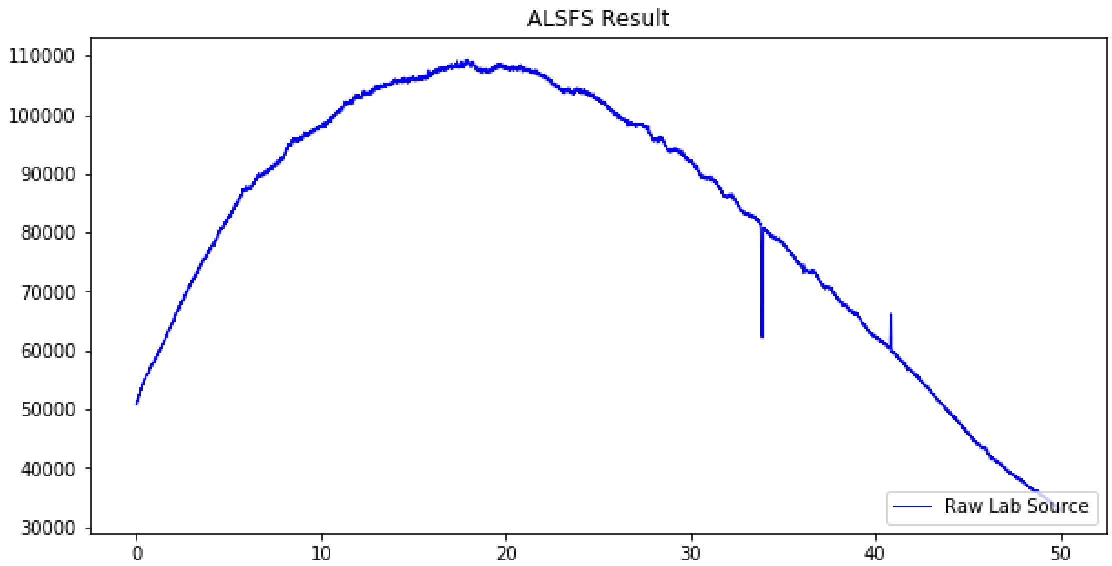
[14]:
```
from LS_Smoothing import*

# read Raw Lab Source as Data Frame
data= pd.read_csv("RawLabSource.csv", sep=',')

# Visualize Raw Lab Souce
plt.clf()
plt.figure(figsize=(10,5))
plt.plot(data["wv"], data["intens"], 'b', linewidth=1, label='Raw Lab Source')
plt.legend(loc='lower right')
plt.title("ALSFS Result")
```

[14]:
```
Text(0.5, 1.0, 'ALSFS Result')
```

```
<Figure size 432x288 with 0 Axes>
```



[15]:
```
# Run Lab Source Smoothing
# Result is a dataframe representing the smoother version of the spectrum
result= LSS(data, 0.98, 0.25, 0.97)
print(result)
```

```
          wv        intens
0     0.007576  51052.694370
1     0.015152  51112.225002
2     0.022727  51171.723935
3     0.030303  51231.191148
4     0.037879  51290.626650
5     0.045455  51350.030418
6     0.053030  51409.402463
7     0.060606  51468.742770
8     0.068182  51528.051325
9     0.075758  51587.328137
10    0.083333  51646.573184
11    0.090909  51705.786476
12    0.098485  51764.967990
13    0.106061  51824.117736
14    0.113636  51883.235699
15    0.121212  51942.321865
16    0.128788  52001.376243
17    0.136364  52060.398811
18    0.143939  52119.389578
19    0.151515  52178.348530
20    0.159091  52237.275652
21    0.166667  52296.170953
```

```
22      0.174242  52355.034412
23      0.181818  52413.866037
24      0.189394  52472.665805
25      0.196970  52531.433726
26      0.204545  52590.169786
27      0.212121  52648.873969
28      0.219697  52707.546285
29      0.227273  52766.186711
...          ...           ...
6570   49.780303  33624.305979
6571   49.787879  33607.746645
6572   49.795455  33591.199342
6573   49.803030  33574.664091
6574   49.810606  33558.140849
6575   49.818182  33541.629636
6576   49.825758  33525.130452
6577   49.833333  33508.643318
6578   49.840909  33492.168190
6579   49.848485  33475.705090
6580   49.856061  33459.254016
6581   49.863636  33442.814990
6582   49.871212  33426.387968
6583   49.878788  33409.972971
6584   49.886364  33393.569998
6585   49.893939  33377.179071
6586   49.901515  33360.800145
6587   49.909091  33344.433241
6588   49.916667  33328.078359
6589   49.924242  33311.735520
6590   49.931818  33295.404680
6591   49.939394  33279.085860
6592   49.946970  33262.779060
6593   49.954545  33246.484301
6594   49.962121  33230.201539
6595   49.969697  33213.930796
6596   49.977273  33197.672070
6597   49.984848  33181.425382
6598   49.992424  33165.190690
6599   50.000000  33148.968014

[6600 rows x 2 columns]
```

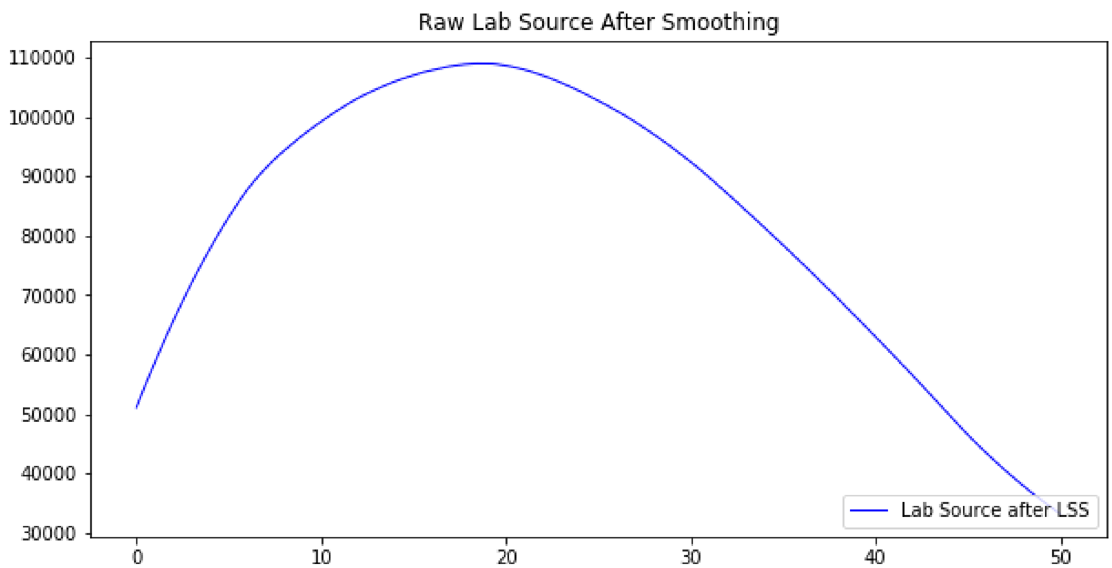
[16]:

```
# Visualize the raw Lab Source after smoothing
plt.clf()
plt.figure(figsize=(10,5))
plt.plot(result["wv"], result["intens"], 'b', linewidth=1, label='Lab Source␣
 ↪after LSS')
```

```
plt.legend(loc='lower right')
plt.title("Raw Lab Source After Smoothing")
```

[16]: Text(0.5, 1.0, 'Raw Lab Source After Smoothing')

<Figure size 432x288 with 0 Axes>



[ ]: