

Results

May 14, 2019

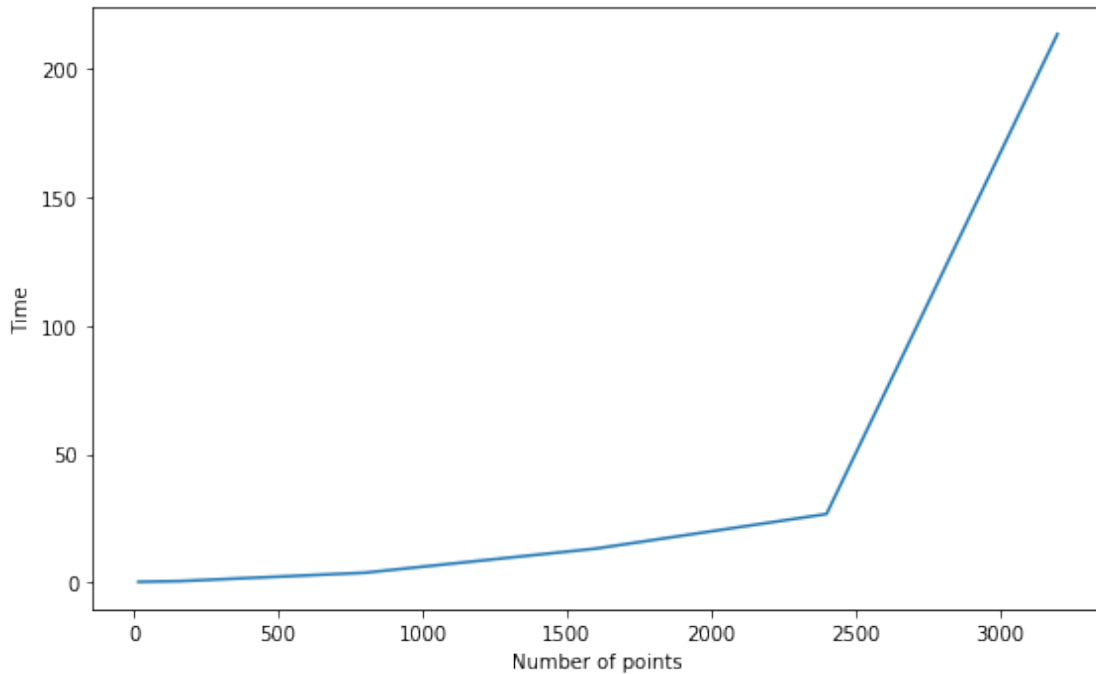
```
In [1]: import numpy as np
import pandas as pd
import sys
import matplotlib.pyplot as plt
import time
import itertools
import pickle
import operator

#default size of the graph
plt.rcParams['figure.figsize'] = (10.0, 8.0)

%load_ext autoreload
%autoreload 2
```

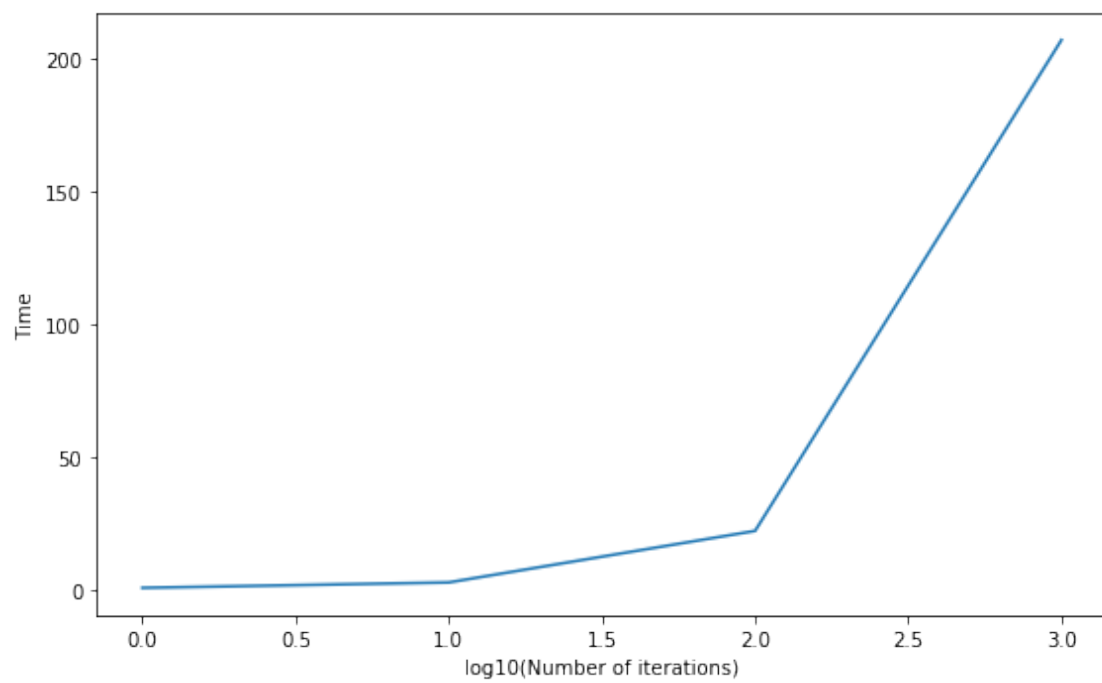
0.0.1 The experiment was run on Prince with 16 processes. The graph below shows the time against the number of points when fixing the number of iterations to be 100000.

```
In [11]: plt.figure(figsize=(8, 5))
#fig, ax = plt.subplots(1,1)
xs = [16 * x for x in [1,5, 10, 50, 100,150, 200]]
ys = [0.3,0.44, 0.58,3.84, 13.23,26.69, 213.45 ]
plt.plot(xs,ys)
plt.xlabel('Number of points')
plt.ylabel('Time')
# ax.tick_params(labelsize=)
# ax.set_xticks(range())
# ax.set_xticklabels()
# plt.legend()
#plt.savefig()
plt.tight_layout()
plt.show()
```



0.0.2 The experiment was run on Prince with 16 processes. The graph below shows the time against the number of iteration when fixing the number of points to be 32000.

```
In [12]: plt.figure(figsize=(8, 5))
         #fig, ax = plt.subplots(1,1)
         xs = [0,1,2,3]
         ys = [0.75,2.79,22.2,207.06]
         plt.plot(xs,ys)
         plt.xlabel('log10(Number of iterations)')
         plt.ylabel('Time')
         # ax.tick_params(labelsize=)
         # ax.set_xticks(range())
         # ax.set_xticklabels()
         # plt.legend()
         #plt.savefig()
         plt.tight_layout()
         plt.show()
```



In []: