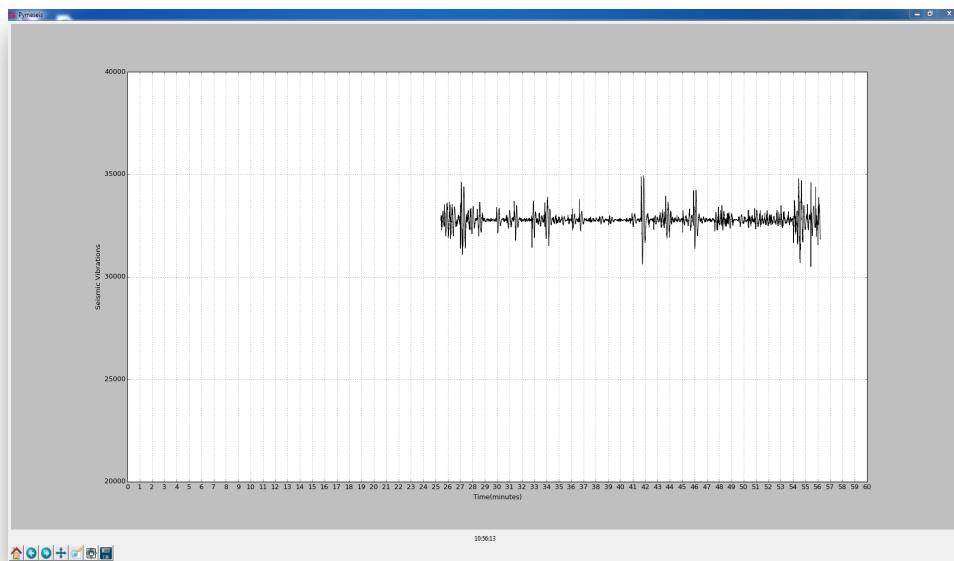


# Project Log Book – Semester Two

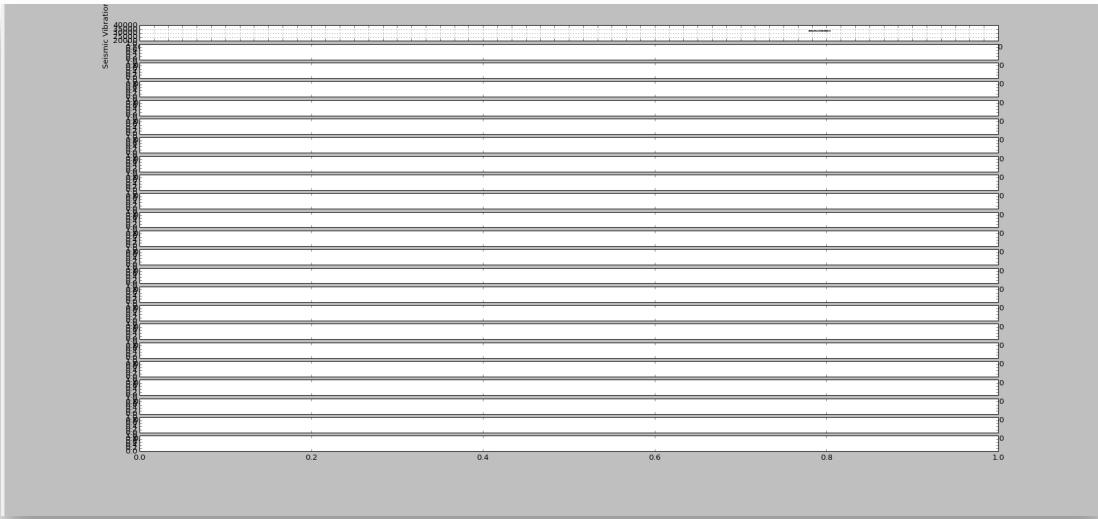
## Week 1

- I realised that plotting the array containing all values (old and new) was inefficient so I decided to keep 2 arrays, one that stored all the values from the beginning of the collection and another array which stored only the last 10 values, and this array was drawn onto the plot without clearing the old points. This took sometime but was successful. I am using the `plot.draw()` method here. The array that contains all the points will be used for saving mseed after every hour.
- Worked on getting the x axis to go from 0 - 60 minutes



## Week 2

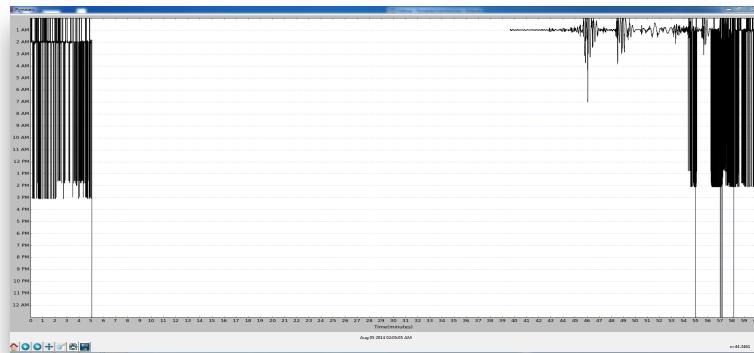
- Had Meeting with Dr Patrice Delmas
- Initially I was just saving points collected from TC1 (that go on the y axis) in an array along with another array saving number starting from 0 that incremented by 0.1, element by element. This was meant to represent the time axis. However I was getting inaccurate plots so started saving the exact time down to the millisecond for each value was read from the TC1. This allowed the plot to display values exactly where they are meant to be on the plot. This solved inaccuracy issue in the plot.
- I tried to implement 24 individual plots using the `plot.add_subplot()` however the result was not pleasing due to each subplot being too small in size to see the plot



- To tackle this issue and be able to get 24 plots, I'm using the same plot, but scaling it enough to be able to see it well and have 24 plots in 1 subplot. Then translating the first plot right to the top of the subplot to begin the plotting. After every hour they are shifted down by an hour.

## Week 3

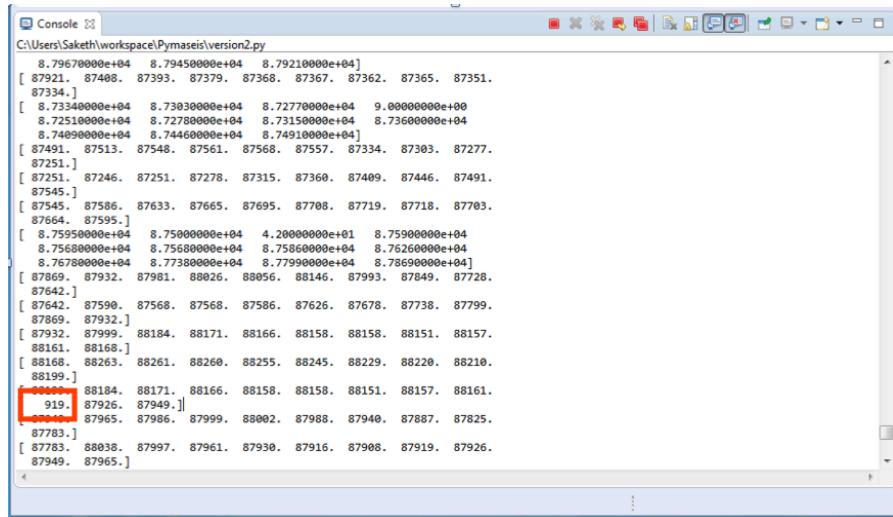
- Once the first plot reaches an hour, the next incoming values are shifted one level down from top plot
- Y axis is now auto generated according to current system time, and shows 24 hours from the start hour (allowing 24 - 1 hour plots to be plotted)
- Worked on handling erroneous values that come from TC1, however not fully successful - Issue will be discussed with Dr Kasper van Wijk on Friday's meeting



- Created and Updated Log Book and Project Website
- Preparing slides and practising for BTech seminar
- Received presentation guidance from Dr. Patrice Delmas
- Presentation went well, constructive feedback received about project and direction

## Week 4

- Started looking into reasons that caused random values appearing in arrays that caused the plots to distort



```
C:\Users\Saketh\workspace\Pymaseis\version2.py
  8.79670000e+04  8.79450000e+04  8.79210000e+04]
[ 87921.  87408.  87393.  87379.  87368.  87367.  87362.  87365.  87351.
 87334.]
[ 8.73340000e+04  8.73030000e+04  8.72770000e+04  9.00000000e+00
 8.72510000e+04  8.72780000e+04  8.73150000e+04  8.73600000e+04
 8.74090000e+04  8.74460000e+04  8.74910000e+04]
[ 87491.  87513.  87548.  87561.  87568.  87557.  87334.  87303.  87277.
 87251.]
[ 87251.  87246.  87251.  87278.  87315.  87360.  87409.  87446.  87491.
 87545.]
[ 87545.  87586.  87633.  87665.  87695.  87708.  87719.  87718.  87703.
 87664.  87595.]
[ 8.75950000e+04  8.75000000e+04  4.20000000e+01  8.75900000e+04
 8.75680000e+04  8.75680000e+04  8.75860000e+04  8.76260000e+04
 8.76780000e+04  8.77380000e+04  8.77990000e+04  8.78690000e+04]
[ 87869.  87932.  87981.  88026.  88056.  88146.  87993.  87849.  87728.
 87642.]
[ 87642.  87590.  87568.  87568.  87586.  87626.  87678.  87738.  87799.
 87869.  87932.]
[ 87932.  87999.  88184.  88171.  88166.  88158.  88158.  88151.  88157.
 88161.  88168.]
[ 88168.  88263.  88261.  88260.  88255.  88245.  88229.  88220.  88210.
 88199.]
[ 88199.  88184.  88171.  88166.  88158.  88158.  88151.  88157.  88161.
 87926.  87949.]
[ 87926.  87965.  87966.  87999.  88002.  87988.  87940.  87887.  87825.
 87783.]
[ 87783.  88038.  87997.  87961.  87930.  87916.  87908.  87919.  87926.
 87949.  87965.]
```

- I was offered access to Dr. Kasper's office
- I setup a station with a TC-1 seismometer there and continued development of the project there
- From this week onwards I met with Dr. Kasper almost every day if not once in two days
- Contact with Dr. Patrice Delmas is kept by email as he is overseas
- Problem was explained to Dr. Kasper who suggested to remove the sampling of values in the application and move towards a continues data collection process

## Week 5

- Along with making the data collection process continuous I removed all of the data decoding code I had written and the application now is not plotting distorted plots.
- Plotting delay issue recognised, after 10 minutes plotting starts to lag behind the actual incident time
- I re-designed the application from a single process with 1 thread of execution to a multi threaded application where the data collection and data plotting are running on separate threads
- For this I am learning how threads work in Python and how GIL works
- Plotting delay problem still exists even after making application into a multi-threaded application. The delay now takes 25 minutes into running the application where as before the delay began by 10 minutes into runtime

## Week 6

- I have spent time learning the difference between multi threading and multi processing as I felt that the problem might be because of GIL in python, as it doesn't allow multiple threads to be executed at once.
- I re-designed the application again to see if it is more efficient if I move away from a multi threaded application to a multi-process application

## Week 7

- I took most of the week learning how to write a multi processing application in python using the Multiprocessing module and also learning about inter-process communication using pipes
- I re-developed the application with multiple processes – that split the application into two processes – a plotting process and a collecting process
- Delay issues was unfortunately still unsolved, where the delay can be noticeable after 3 hours into run time and delay becomes larger and larger
- Upon consulting Dr. Robert Sheehan who taught me Python, Multi-Threading and Multi-Processing in CompSci 340, he recommended me to create two time stamps around the threads and methods and run efficiency tests to see which processes or methods took a lot of time and also showed signs of their time increasing

## Week 8

- Upon following Dr. Robert's advice I found that the plot.draw() method I was running was plotting all the points again on the plot although I was sending it only the last 10 collected values. This draw method was highly inefficient where it started plotting which took 0.39seconds and kept increasing by 0.02seconds by every array. This delay took longer and longer as I moved from 1 thread to multiple threads to multiple processes as these architectures were progressively more efficient but the method was the problem and hence I was unable to address the issue although redesigning the application
- Now that I understood the problem, I researched into ways of fast plotting without using the draw method. I immediately noticed that many others had the same problem as me with the delay issues caused by the draw function of the Matplotlib module. To address this a person stated to use the Blitting technique to fix the issue. I learnt what blitting meant and searched ways of implementing it in python. To "blit" is to copy bits from one part of a computer's graphical memory to another part. This technique deals directly with the pixels of an image, and draws them directly to the screen, which makes it a very fast rendering technique.
- I successfully managed to rewrite the code and instead of using the draw method it now uses blitting. The result of using blitting is remarkable, what took the draw method 0.39 seconds to begin with and carried on increasing, with the use of blitting I was able to achieve the exact same result but this time the time it took was 0.032 seconds and increased to

0.069seconds by the end of 24 hours where as the draw method increased to 4 minutes and 53 seconds!

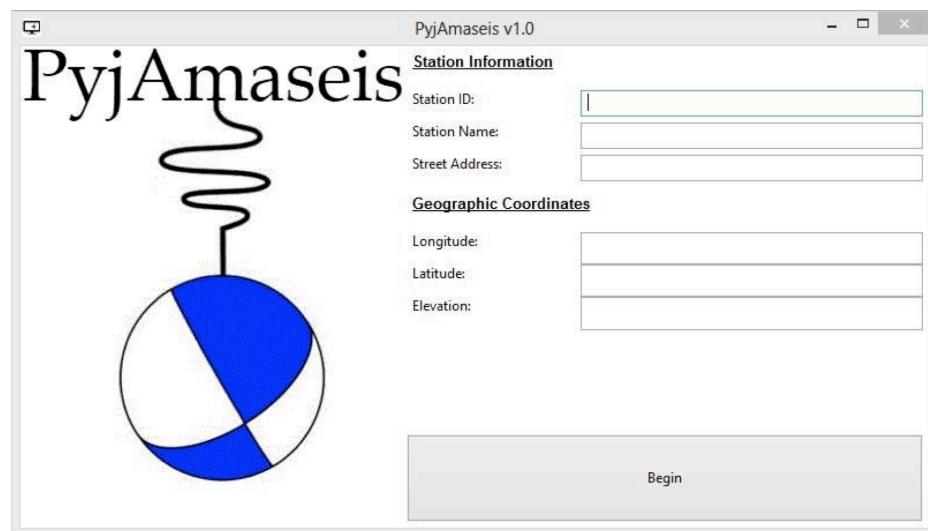
- I reverted back to the multithreading architecture using queues for inter thread communication as this seemed to work just as efficiently now with Blitting as the multiprocessing architecture but it was much more easier to share variables as they are in a shared memory.

## Week 9

- I had to change the data format from saving the data into mseed file formats to SAC file format. This is the same kind of file structure and is used just as much as mseed. The reason for the change was due to some of the header information now being available in the mseed, such as longitude and latitude headers, and this was important information that Dr. Kasper has said we need to save hence the shift from mseed to SAC. This was relatively easy to implement as the ObsPy module allowed me to save traces into SAC format just the same way as saving mseed.
- Had a discussion with Dr. Kasper and received new requirements
  - 1 Hour plot
  - Y shift
  - Header information UI
  - Screenshots

## Week 10

- Created a simple interface that is used to collect station information at the beginning. This was written in wxpython but I used a gui builder called wx glade to create this -

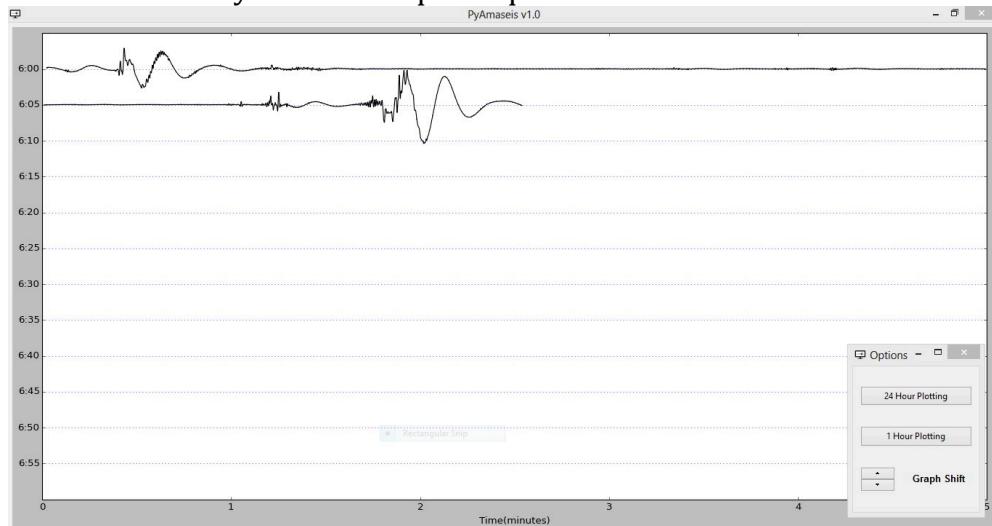


- Upon entering the street address, with the help of the pygeocoder module, the exact geo coordinates of the location – longitude, latitude, and elevation are queried and auto filled
- After entering this data the first time the next time plotting begins directly without the requirement of entering these values

- With the help of the PIL module, using the Image grab class, I am able to take screen shots of the application-plotting window every hour. Will look to make that every 10 minutes in the future after the project is completed

## Week 11

- I worked on creating a new options side window that allows the user to shift between a 24-hour plot to 1-hour plot. And also implement the Y shift functionality that Dr. Kasper requested



- I also implemented an Auto usb function that saves the user from having to know which usb port the TC1 is connected by automatically detecting this by running several property checks.
- I met with Matiu Carr over this week to look into uploading the hourly SAC and PNG screenshots that are taken locally to the NZSeis server that is hosted by the Science IT.
- I was offered some space to work next to Mat, where I spent the rest of the week working.
- I learnt how to write a multi form HTTPS request using PycURL. Mat is working on php script that authenticates request and payload

## Week 12

- Managed to get SAC and Screenshots uploading to server successfully
- Received presentation guidance from Dr. Kasper
- Prepared for final seminar
- Final seminar went successfully
- Application would crash if TC1 not connected when application was launched, fixed this bug by putting in a loop check
- Received guidance from Dr. Patrice on Final Report Structure

## Post Week 12

- Redesigning Project webpage
- Completing Log book
- Completing Final Report