# Solution8

October 24, 2015

```
In [1]: %matplotlib inline
```

```
In [2]: import sys
        sys.path.append("/home/3y9/Mantid/Build/bin")
        from mantid.simpleapi import *
        for run in range(17880,17884):
            Load(Filename='/SNS/users/shared/MantidTrainingCourseData/SEQ_'+str(run)+'_event.nxs',
                 OutputWorkspace='temp')
            Rebin(InputWorkspace='temp', OutputWorkspace='temp', Params='20')
            SumSpectra(InputWorkspace='temp', OutputWorkspace='sum_'+str(run))
            SaveNexus(InputWorkspace='sum_'+str(run),
                      Filename='/SNS/users/3y9/MantidCourse/sum_'+str(run)+'.nxs')
```

```
In [3]: mtd.importAll()
```

```
In [4]: mtd.getObjectNames()
```

```
Out[4]: ['sum_17880', 'sum_17881', 'sum_17882', 'sum_17883', 'temp']
```

```
In [5]: sum_17880.blocksize()
```

```
Out[5]: 834
```

```
In [6]: x0=sum_17880.readX(0)
        y0=sum_17880.readY(0)
        index=y0.argmax()
        print "Maximum value of ",y0.max(), " counts is found at ",(x0[index]+x0[index+1])*0.5, "micros
```

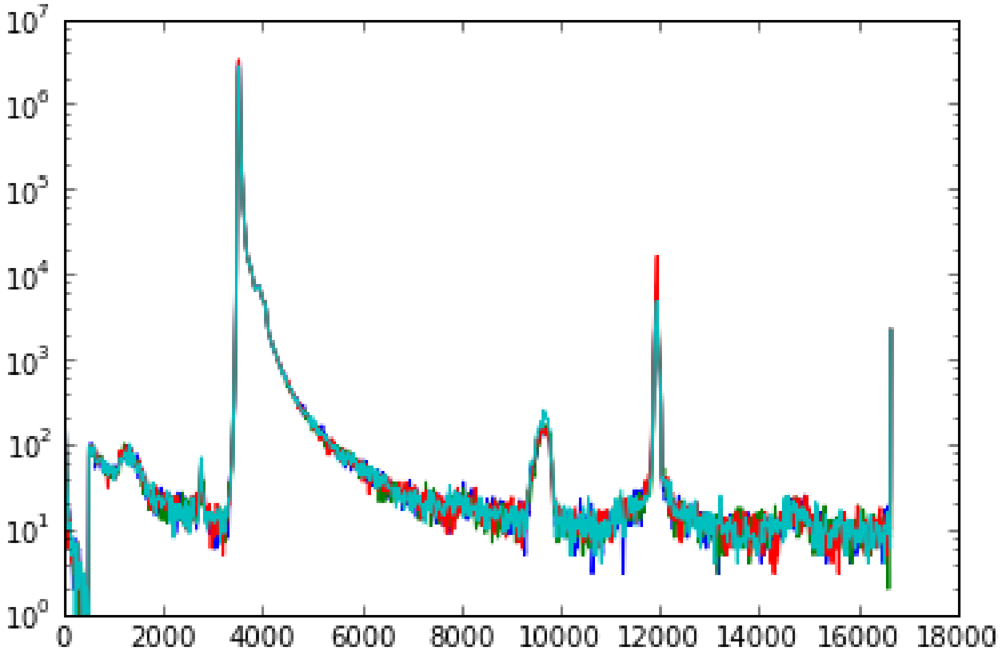
```
Maximum value of  3052689.0  counts is found at  3529.0 microseconds
```

```
In [7]: x1=sum_17881.readX(0)
        y1=sum_17881.readY(0)
        x2=sum_17882.readX(0)
        y2=sum_17882.readY(0)
        x3=sum_17883.readX(0)
        y3=sum_17883.readY(0)
        import matplotlib.pyplot as plt

        xp0=(x0[:-1]+x0[1:])/2
        xp1=(x1[:-1]+x1[1:])/2
        xp2=(x2[:-1]+x2[1:])/2
        xp3=(x3[:-1]+x3[1:])/2

        plt.plot(xp0,y0)
        plt.plot(xp1,y1)
        plt.plot(xp2,y2)
        plt.plot(xp3,y3)

        plt.yscale('log')
        plt.show()
```



```
In [8]: import numpy as np
        x=np.append(x0,x1)
        x=np.append(x,x2)
        x=np.append(x,x3)
        y=np.append(y0,y1)
        y=np.append(y,y2)
        y=np.append(y,y3)
        w=CreateWorkspace(x,y,Nspec=4)
        print w.getNumberHistograms()
        w.setX(2,w.readX(2)+1000)
        SaveNexus(w,Filename="/SNS/users/3y9/MantidCourse/newWS.nxs")
```

```
4
```

```
In [9]: w=Load('/SNS/users/shared/MantidTrainingCourseData/SEQ_17880_event.nxs')
        inst=w.getInstrument()
        print type(inst)
        print inst.nelements()
```

```
<class 'mantid.geometry._geometry.Instrument'>
6
```

```
In [10]: print inst.getSource().getPos()
```

```
[0,0,-20.0114]
```

```
In [11]: for i in range(inst.nelements()):
             print inst[i].getName(), type(inst[i])
```

```
moderator <class 'mantid.geometry._geometry.ObjComponent'>
sample-position <class 'mantid.geometry._geometry.ObjComponent'>
monitors <class 'mantid.geometry._geometry.CompAssembly'>
B row <class 'mantid.geometry._geometry.CompAssembly'>
C row <class 'mantid.geometry._geometry.CompAssembly'>
D row <class 'mantid.geometry._geometry.CompAssembly'>
```

```
In [12]: for i in range(3,6):
             print inst[i].getName(),' has ',inst[i].nelements(), ' banks'
```

```
B row  has  37  banks
C row  has  39  banks
D row  has  37  banks
```

```
In [13]: type(inst[3][0][0][0][0])
```

```
Out[13]: mantid.geometry._geometry.Detector
```

```
In [14]: import mantid
         comp=inst[3]
         while type(comp)!=mantid.geometry.Detector:
             print comp.getName(),'/',
             comp=comp[0]
         print comp.getName()
         print 'ID=',comp.getID()
         print '2Theta=',np.degrees(comp.getTwoTheta(inst.getSample().getPos(),inst.getSource().getPos(
```

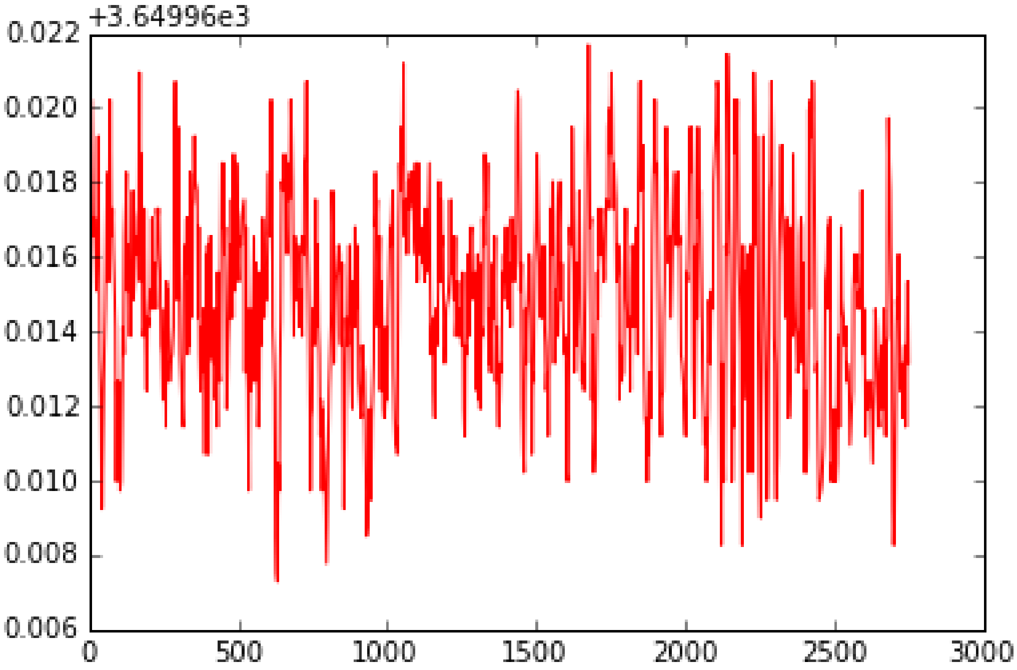
```
B row / B1 / eightpack / tube1 / pixel1
ID= 37888
2Theta= 61.9261260274
```

```
In [15]: phase1=w.getRun()['Phase1']
         stat=phase1.getStatistics()
         print 'Mean=',stat.mean
         print 'Min=',stat.minimum
         print 'Max=',stat.maximum
         print 'TimeAverage=',phase1.timeAverageValue()
```

```
Mean= 3649.97487646
Min= 3649.96728516
Max= 3649.98168945
TimeAverage= 3649.97483767
```

```
In [16]: times=[]
         for t in phase1.times:
             times.append((t-phase1.times[0]).total_seconds())
```

```
In [17]: plt.plot(times,phase1.value,'r-')
         plt.show()
```


+3.64996e3
0.022
0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0
500
1000
1500
2000
2500
3000

```
In [17]:
```