Plot1D

March 13, 2017

1 One curve

1.1 plot a simple curve and play with it

- $x = [0, pi]$
- $y = e^x$
- see documentation:
  - http://www.silx.org/doc/silx/dev/modules/gui/plot/plotwindow.html#silx.gui.plot.PlotWindow.
- see tutorial:
  - http://www.silx.org/doc/silx/dev/modules/gui/plot/getting_started.html

play with the interface: - log scale - grid - display points - . . .

```
In [ ]: import numpy
        from silx.gui.plot import Plot1D
        %gui qt
```

```
In [ ]: x=numpy.linspace(0, numpy.pi, 1000)
        y=numpy.exp(x)
```

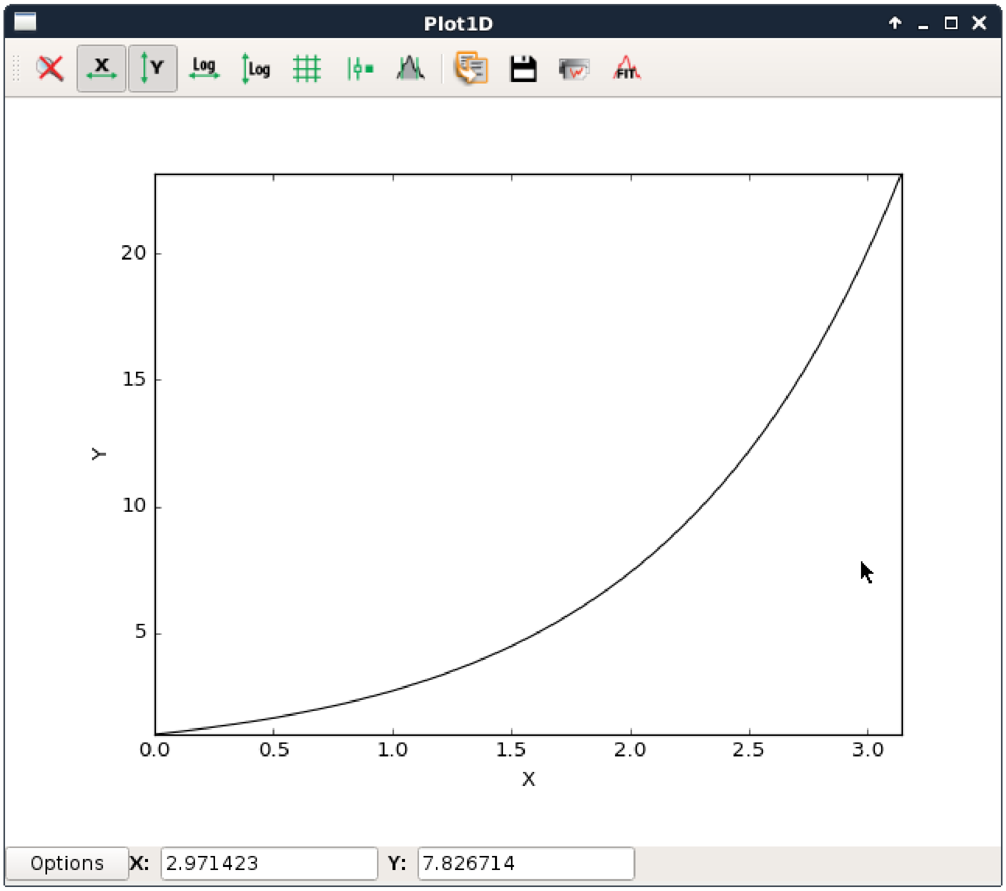
```
In [ ]: p=Plot1D()
        p.addCurve(x, y, legend='exp')
        p.show()
```

1.2 Shift the curve

get back the curve and add an offset in y axis

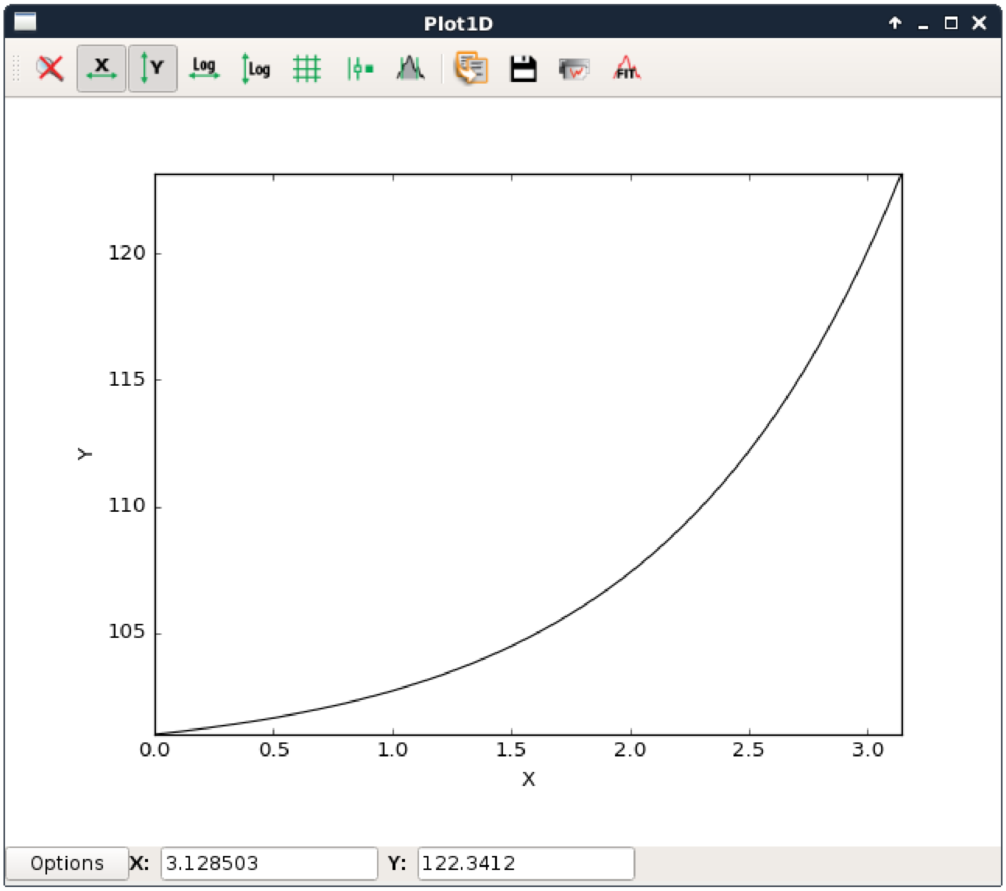
- $y = y + 100.0$
- get all needed data from the 'Plot1D' object

```
In [ ]: x, y, legend, info, _params=p.getCurve('exp')
        y=y+100
        p.show()
        p.addCurve(x, y, color='red', legend='exp')
        p.show()
```



X: 2.971423 Y: 7.826714

larger-iso-surface-using-silx-plot3d



X: 3.128503 Y: 122.3412

shift exponential

2 Many curves

2.1 plot the following functionin the same plot window

- $y = sin(x)$
- $y = cos(x)$
- $y = x$
- play with the curve selection from options->legend

```
In [ ]: y_sin = numpy.sin(x)
        y_cos = numpy.cos(x)
        p.clear()
        p.addCurve(x, y_sin, color='green', legend='sin')
        p.addCurve(x, y_cos, color='red', legend='cos', linestyle=' ', symbol='o')
        p.addCurve(x, x, color='blue', legend='x', linestyle='-', symbol='s')
        p.show()
```

2.2 remove one curve by the id

- using the 'Plot1D' object

```
In [ ]: p.remove('x')
        p.show()
```

2.3 shift curves by 30 in the x axis

- by using the functions of the 'Plot1D' object
- keep at least the color of the curve
- Result should be close to

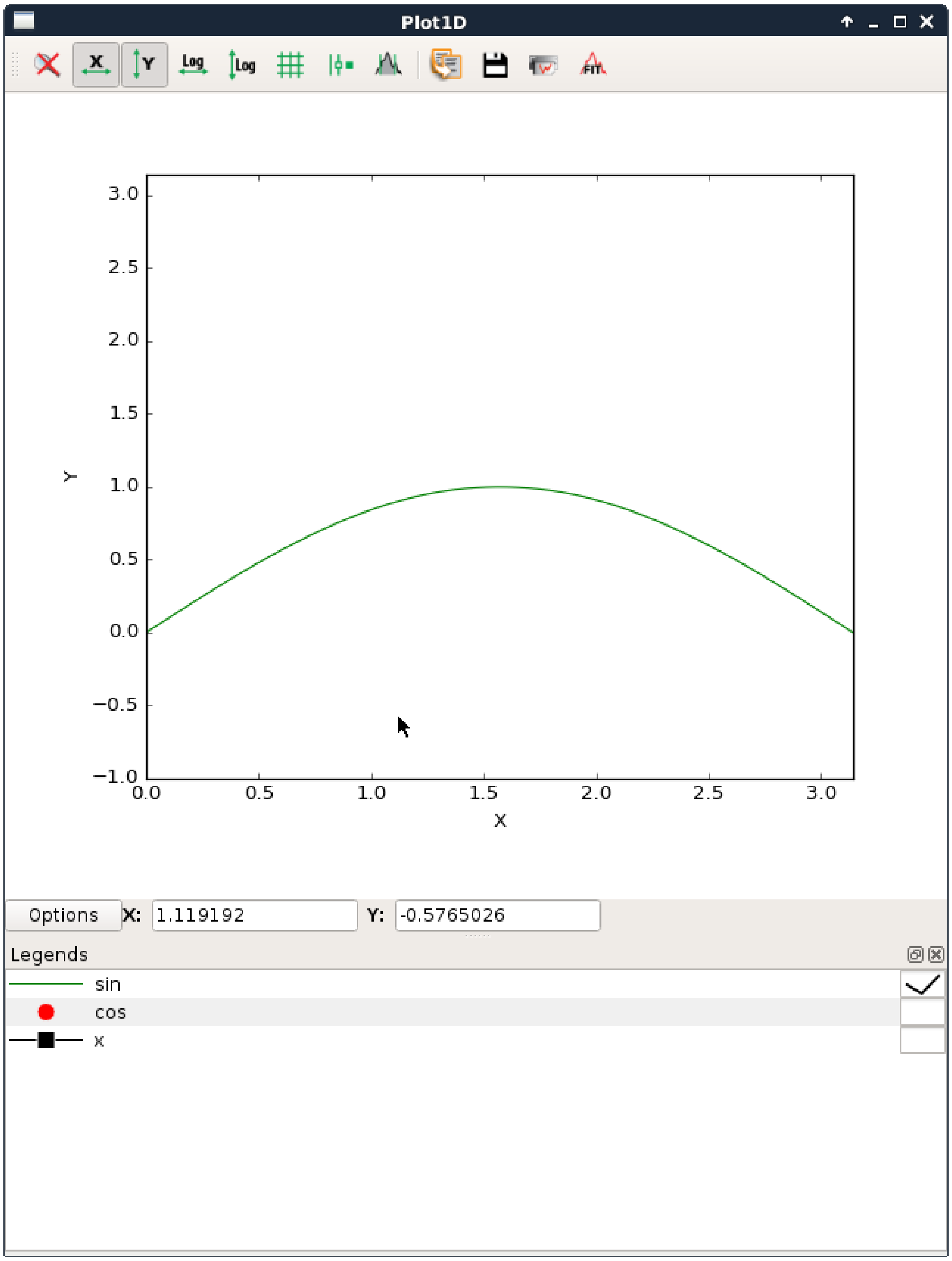
```
In [ ]: curves=p.getAllCurves()
        p.clear()
        for curve in curves:
            x, y, legend, info, params=curve
            x=x+10.0
            p.addCurve(x, y, legend=legend, color=params['color'] )

        p.show()
```

3 ROI

3.1 load data from data/spectrum.dat

```
In [ ]: import silx.io
        sf = silx.io.open("
                                  data/spectrum.dat")
        x_data=sf['1.1/measurement/channel']
        y_data=sf['1.1/measurement/counts']
```

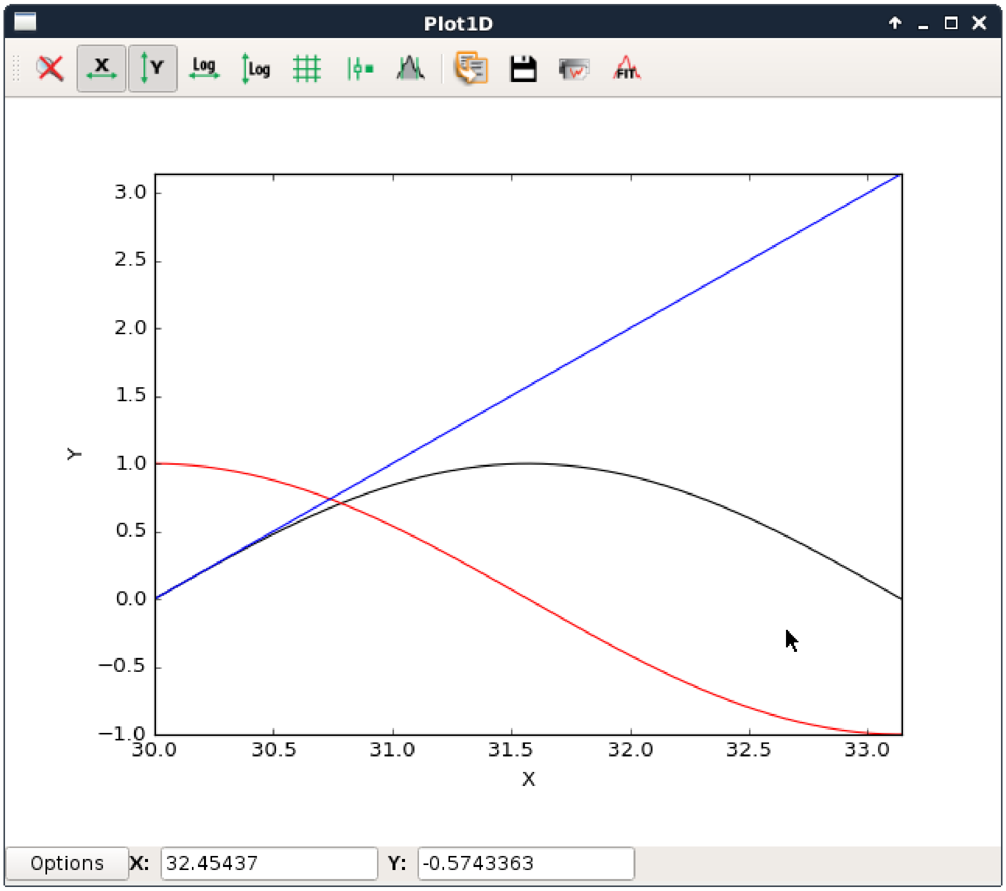


X: 1.119192 Y: -0.5765026

Legends

—	sin	✓
●	cos	
—■—	x	

plot_legends



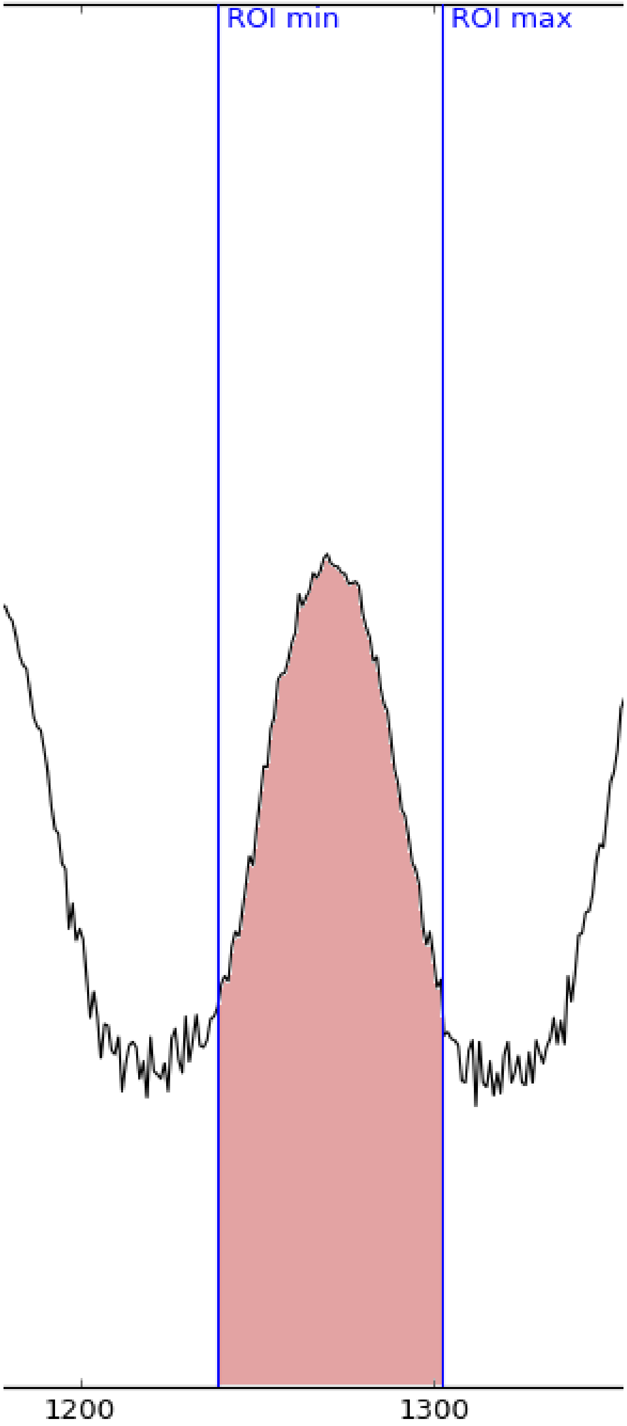
X: 32.45437 Y: -0.5743363

plot1D_shiftcurves

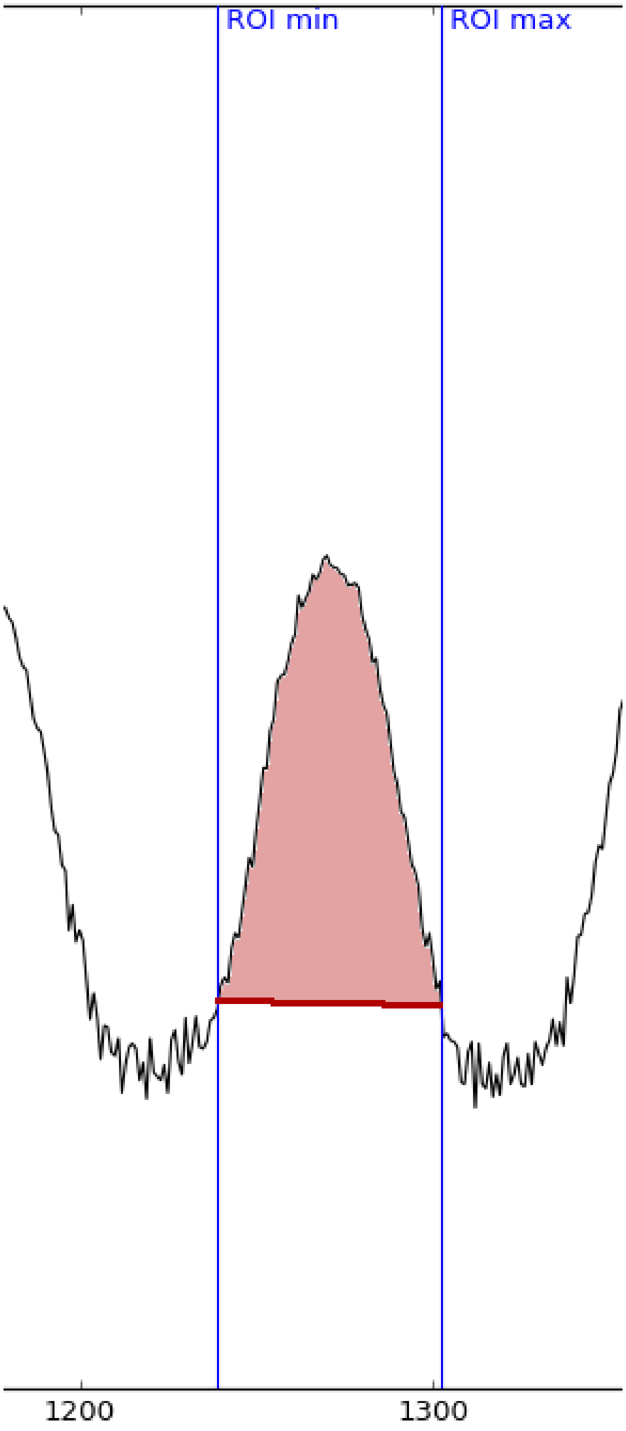
3.2 Plot the data

```
In [ ]: plot=Plot1D()
        x=numpy.linspace(0.0, numpy.pi)
        y=numpy.sin(x)
        plot.addCurve(x_data, y_data)
        plot.setYAxisLogarithmic(True)
        plot.show()
```

options -> ROI -> add ROI -> select min and max limits. estimate integral between lower



and upper limits - Raw counts - Net counts



```
In [ ]:
```