Plot1DExercise

November 14, 2016

1 One curve

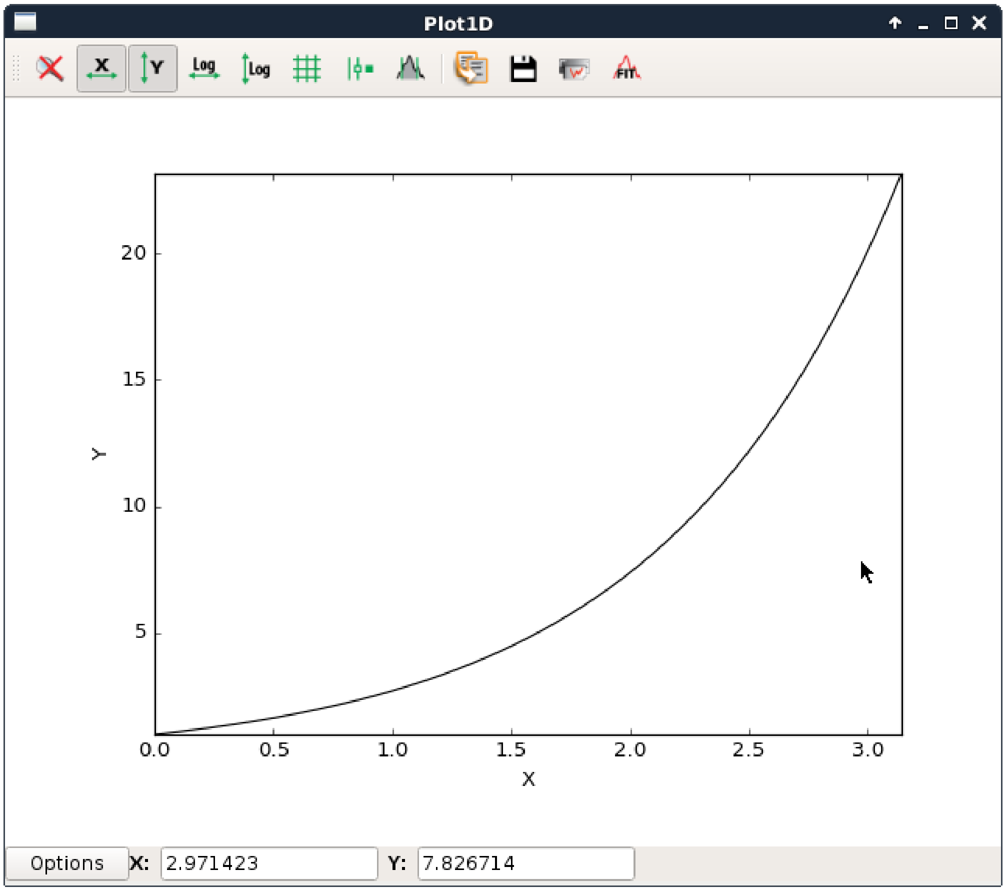
1.1 plot a simple curve and play with it

- $x = [0, pi]$
- $y = e^x$
- see documentation: http://www.silx.org/doc/silx/dev/modules/gui/plot/plotwindow.html#silx.gui.plo
- see tutorial: http://www.silx.org/doc/silx/dev/modules/gui/plot/getting_started.html
- use Plot1D and Plot1D.addCurve
  - legend is used as the ID of the curve. So if a new curve is setted with an existing id it will erase the first curve

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

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

```
In [ ]: ...
```



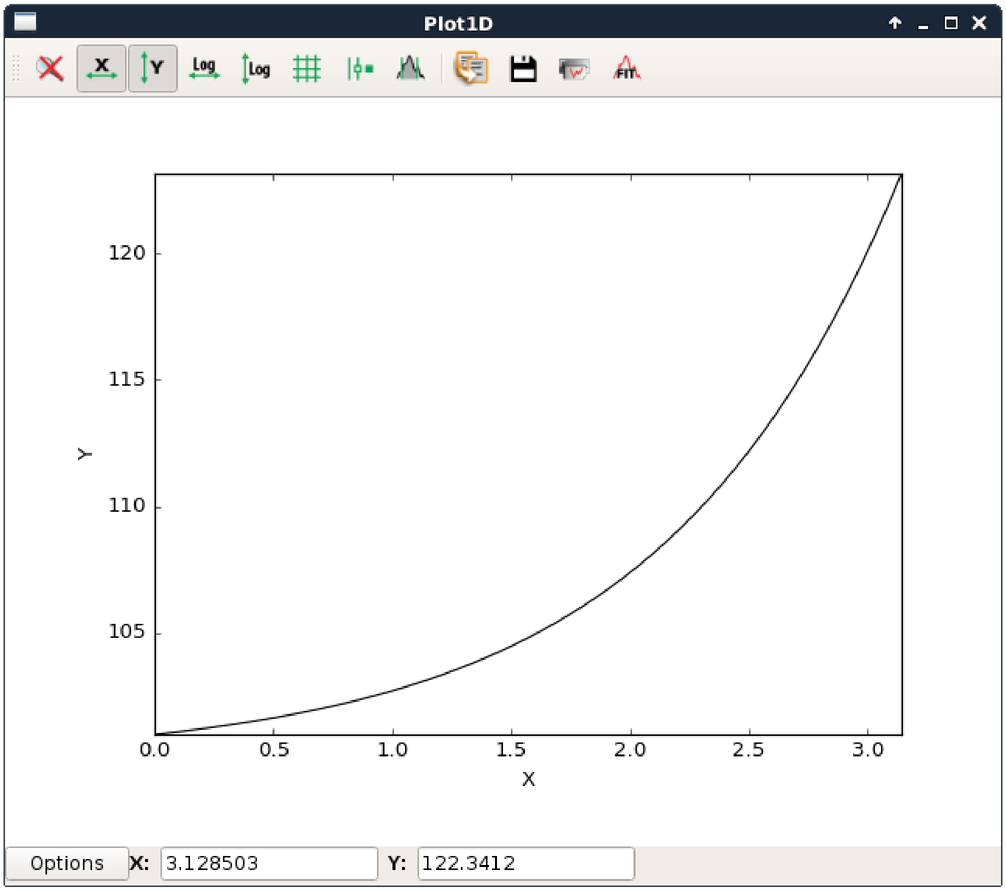
X: 2.971423 Y: 7.826714

larger-iso-surface-using-silx-plot3d

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
- use getCurve([curveID]) function. Return :
    - x
    - y
    - legend
    - info (if some informations has been added)
    - params (color, linewidth...)



X: 3.128503 Y: 122.3412

shift exponential

```
In [ ]: ...
```

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 [ ]: ...
```

2.2 remove one curve by the id

- using the 'Plot1D' function 'remove([curveID])'

```
In [ ]: ...
```

2.3 shift curves by 30 in the x axis

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

```
In [ ]: ...
```

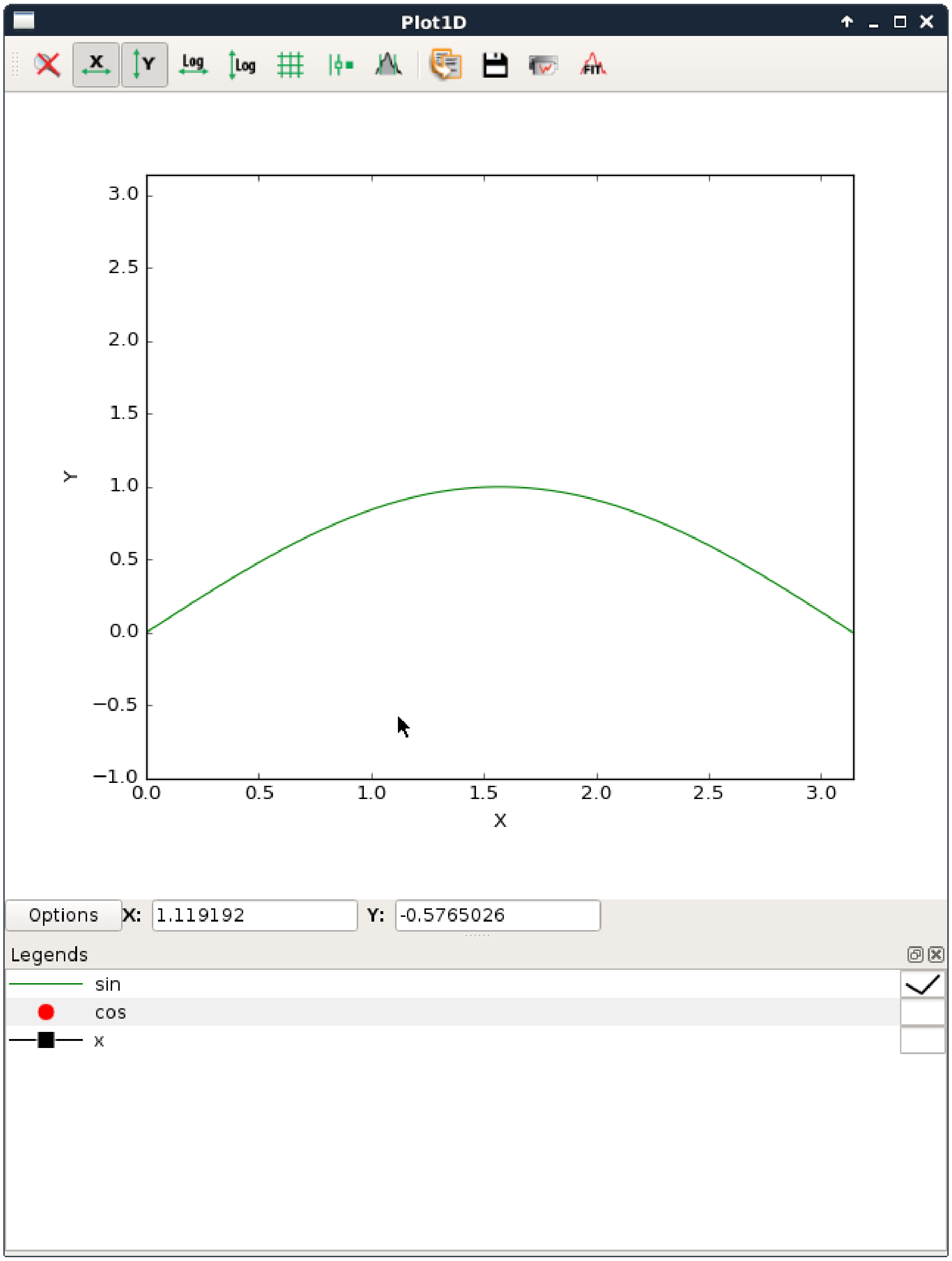
3 ROI

3.1 load data from data/spectrum.dat

```
In [ ]: from silx.io import spech5
        sf = spech5.SpecH5("data/spectrum.dat")
        #print(specfile['1.1/measurement/'].keys())
        x_data=sf['1.1/measurement/channel']
        y_data=sf['1.1/measurement/counts']
```

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()
```

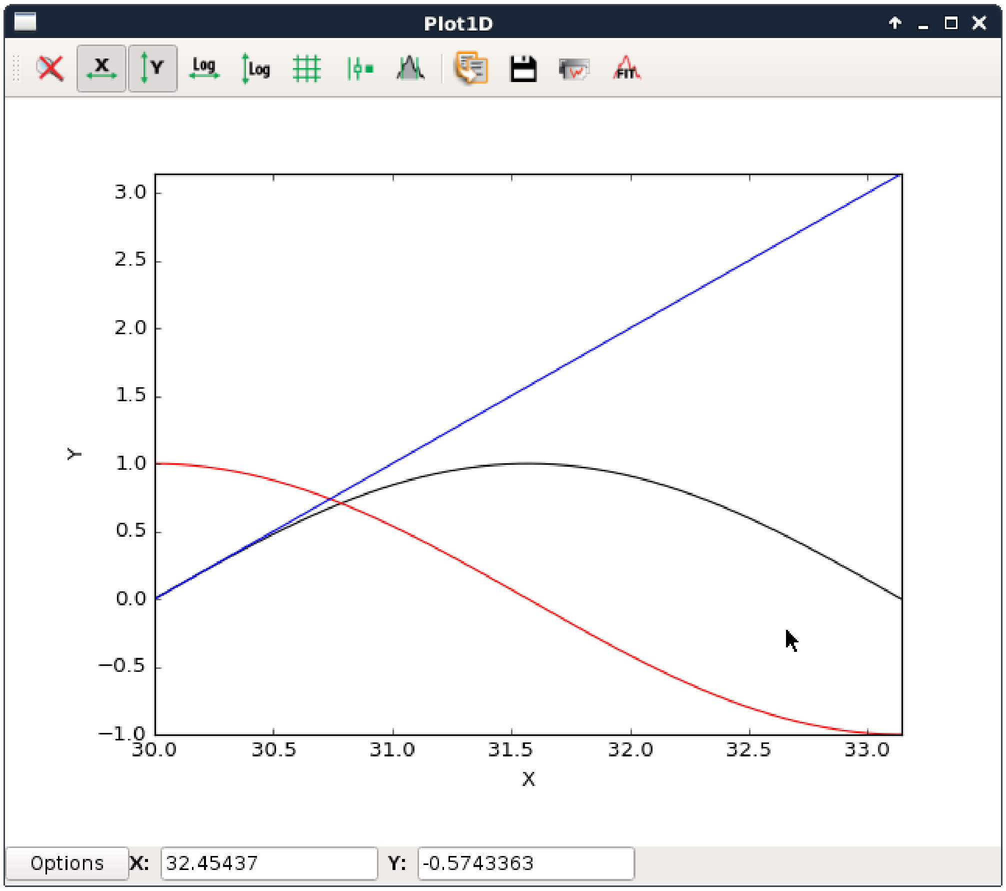


X: 1.119192 Y: -0.5765026

Legends

—	sin	✓
●	cos	
—■—	x	

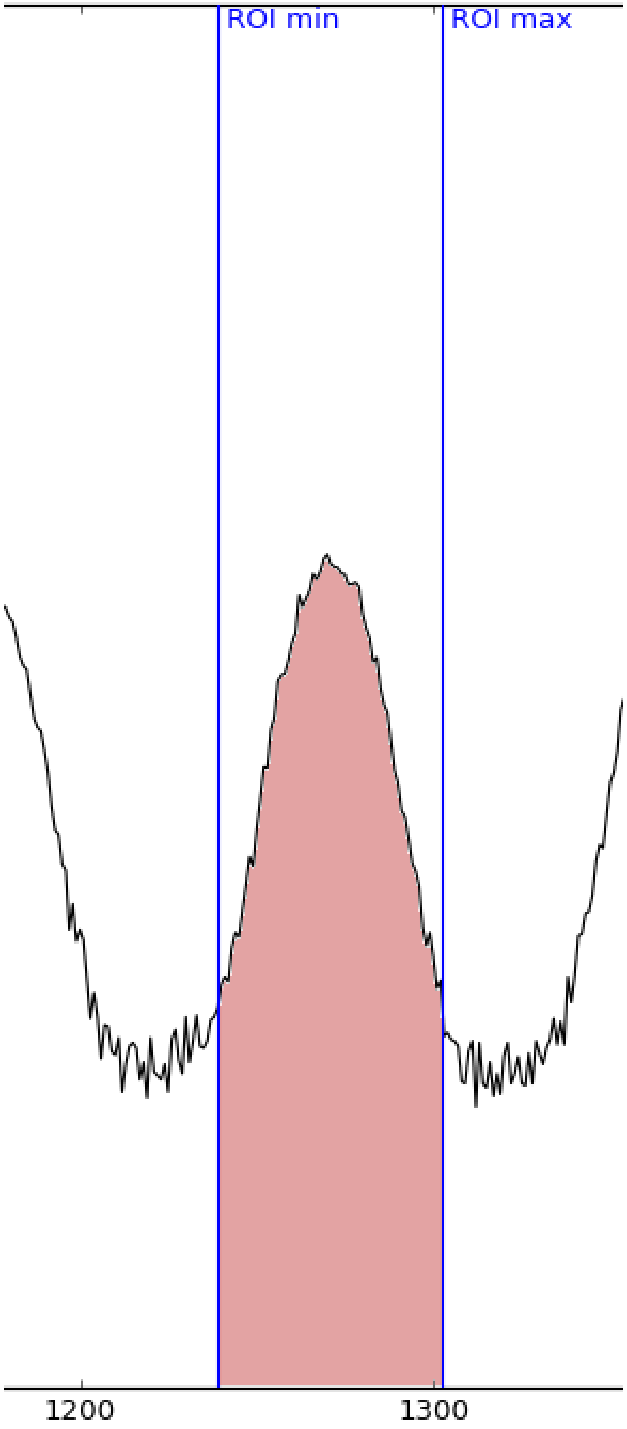
plot_legends



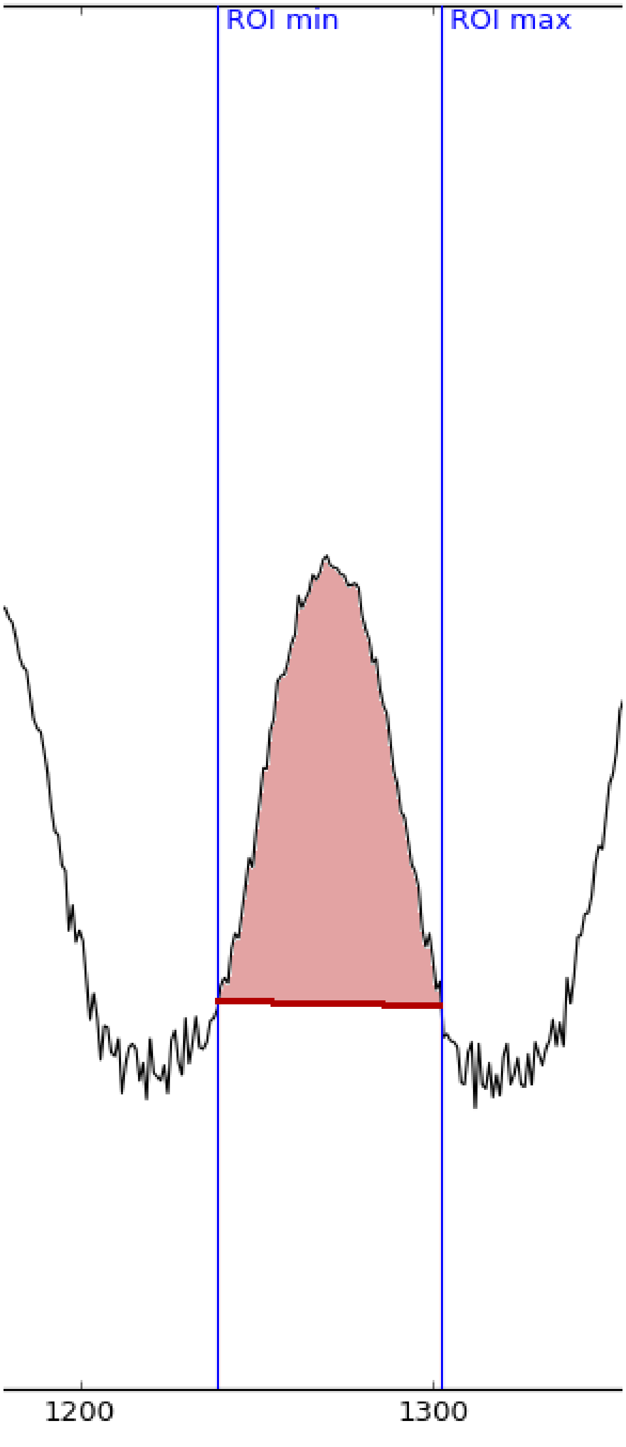
X: 32.45437 Y: -0.5743363

plot1D_shiftcurves

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



and upper limits - Raw counts - Net counts



```
In [ ]:
```