

BornAgain - Feature #948

Feature # 287 (Rejected): IntensityData campaign

Implement new IntensityData object with improved usability

02 Feb 2015 11:05 - pospelov

Status: Archived	Start date: 02 Feb 2015
Priority: Normal	Due date:
Assignee: pospelov	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version: Sprint 29	
Description Consider creation of new IntensityData object to replace current typedefs in PythonAPI typedef OutputData<double> IntensityData; <ul style="list-style-type: none">• New IntensityData has to have a convenient set of constructors to create either 1D or 2D objects• with histogramming functionality (Fill(Phi_f, alpha_f, weight))• with convenient access to axes bin information• with slicing possibility (2D->1D)• with integrated masking• with integrated weights (for fitting)• with rebinning functionality What will be the relation of this class with the Detector class?	

History

#1 - 02 Feb 2015 11:06 - pospelov

- Description updated

- Parent task set to #287

#2 - 13 Apr 2015 14:39 - pospelov

- Status changed from Sprint to Backlog

- Target version deleted (Sprint 26)

#3 - 22 Apr 2015 13:17 - herck

- Target version set to Sprint 27

#4 - 22 Apr 2015 13:18 - herck

- Status changed from Backlog to Sprint

#5 - 26 May 2015 10:38 - herck

- Status changed from Sprint to Backlog

#6 - 26 May 2015 10:38 - herck

- Target version deleted (Sprint 27)

#7 - 10 Jun 2015 13:33 - pospelov

- Status changed from Backlog to Sprint

- Target version set to Sprint 28

#8 - 29 Jul 2015 12:54 - herck

- Status changed from Sprint to Backlog

#9 - 29 Jul 2015 12:54 - herck

- Target version deleted (Sprint 28)

#10 - 03 Sep 2015 11:38 - pospelov

- Status changed from Backlog to Sprint

- Target version set to Sprint 29

#11 - 14 Sep 2015 16:16 - pospelov

- Assignee set to pospelov

#12 - 23 Sep 2015 14:49 - pospelov

- File figure_1.png added

- File figure_2.png added

- Status changed from Sprint to Resolved

Two new classes are available: Histogram1D and Histogram2D. See code examples
Suggestions on API are welcome.

```
def test_histograms_from_intensity():
    """
    Making 2D histograms from intensity data, making profiles
    """
    data =
ba.IntensityDataOfactory.readIntensityData("/home/pospelov/development/git/gisas/support/001_ElisabethJosten/data/004_230_P144_cut1.int.gz")

h2 = ba.Histogram2D(data)

plt.figure(1)

# standard color map for 2D histogram
plt.subplot(2, 2, 1)
plt.imshow(h2.getArray(), norm=matplotlib.colors.LogNorm(), extent=[h2.getXmin(), h2.getXmax(), h2.getYmin(), h2.getYmax()])

# making crop
plt.subplot(2, 2, 2)
crop = h2.crop(0.026, 0.008, 0.030, 0.013)
plt.imshow(crop.getArray(), norm=matplotlib.colors.LogNorm(), extent=[crop.getXmin(), crop.getXmax(), crop.getYmin(), crop.getYmax()])

# Projection along Y. Making 3 slices at 3 different x-values
plt.subplot(2, 2, 3)
proj = h2.projectionY(0.015)
plt.semilogy(proj.getBinCenters(), proj.getBinValues()) # proj.getBinValues() is equivalent proj.getArray()
proj = h2.projectionY(0.016)
plt.semilogy(proj.getBinCenters(), proj.getBinValues())
proj = h2.projectionY(0.017)
plt.semilogy(proj.getBinCenters(), proj.getBinValues())

# Projection along Y. comparing slice at certain x-value with averaged slice (averaged between [xlow, xup])
plt.subplot(2, 2, 4)
proj = h2.projectionY(0.016)
plt.semilogy(proj.getBinCenters(), proj.getBinValues())
proj = h2.projectionY(0.015, 0.017)
plt.semilogy(proj.getBinCenters(), proj.getArray(ba.IHistogram.AVERAGE))
```

figure_1.png

```
def test_a_la_ROOT():  
    ""
```

```
ROOT like histogramming, with access to accumulated value, bin average, bin errors and bin number of entries
```

```
#####  
plt.figure(2)  
  
h2 = ba.Histogram2D(100, -10.0, 10.0, 100, 0.0, 20.0)  
  
for i in range(0, 100000):  
    x = -10.0+20.0*random.random()  
    y = 20.0*random.random()  
    h2.fill(x, y, sinc(x, y))  
  
# bin content (accumulated value in the bin)  
plt.subplot(2, 2, 1)  
im = plt.imshow(h2.getArray(), extent=[h2.getXmin(), h2.getXmax(), h2.getYmin(), h2.getYmax()])  
plt.colorbar(im)  
  
# bin average  
plt.subplot(2, 2, 2)  
im = plt.imshow(h2.getArray(ba.IHistogram.AVERAGE), extent=[h2.getXmin(), h2.getXmax(), h2.getYmin(), h2.getYmax()])  
plt.colorbar(im)  
  
# bin error (RMS calculated over entries)  
plt.subplot(2, 2, 3)  
im = plt.imshow(h2.getArray(ba.IHistogram.ERROR), extent=[h2.getXmin(), h2.getXmax(), h2.getYmin(), h2.getYmax()])  
plt.colorbar(im)  
  
# number of entries in the bin  
plt.subplot(2, 2, 4)  
im = plt.imshow(h2.getArray(ba.IHistogram.NENTRIES), extent=[h2.getXmin(), h2.getXmax(), h2.getYmin(), h2.getYmax()])  
plt.colorbar(im)
```

figure_2.png

#13 - 03 Nov 2015 10:01 - pospelov

- Status changed from Resolved to Archived

Files

figure_1.png	140 KB	23 Sep 2015	pospelov
figure_2.png	279 KB	23 Sep 2015	pospelov