

Simulation Pipeline - Bug #3728

k_binspec_new problem

04/19/2013 08:26 AM - Ben Hambrecht

Status:	New	Start date:	04/19/2013
Priority:	Urgent	Due date:	
Assignee:	Laurenz Gamper	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

Execute the following (see Sim_Gal_Spec_Observed_Cunha/test_kbinspec_ben.py). The rebinning from k_binspec_new does not agree with the original function.

```
## Bins a spectrum by just integrating over the pixel edges.
## Note that this will change a flux density to a flux.
import sys
sys.path.append('../Wrapper/')
sys.path.append('../Plotters/')
sys.path.append('kcorrect_python_rewrite/')
sys.path.append('../Utilities/')
import Utilities
import k_binspec_new as k_binspec
import numpy
import pylab
import subprocess
import scipy.integrate
import archive
import scipy.interpolate as interpolate

# import data
with archive.archive('../data/data_bank.h5','r') as ar:
    abswave = numpy.array(ar['/sky/extinction/wave'])
    abstemp = numpy.array(ar['/sky/extinction/power'])
    Nwave = ar['/Sim_Gal_Spec_Observed/number_of_pixels']
    wavelength_min = ar['/Sim_Gal_Spec_Observed/min_wavelength']
    wavelength_max = ar['/Sim_Gal_Spec_Observed/max_wavelength']

# create survey grid (copied from SGSO_Cunha)
dellam = float(wavelength_max - wavelength_min)/Nwave
lambda_survey = numpy.zeros(Nwave, dtype='float64')
lambda_survey[0] = wavelength_min

for i in numpy.arange(1,Nwave):
    lambda_survey[i] = lambda_survey[i-1]+dellam

for i in numpy.arange(1,Nwave-1):
    lambda_survey[i] = lambda_survey[i] + (lambda_survey[i+1]-lambda_survey[i])/2.

# rebin with k_binspec
abstemp_rebinned = numpy.zeros(lambda_survey.shape)
k_binspec.k_binspec(abswave,abstemp,lambda_survey,abstemp_rebinned)

# rebin with interp1d
abstemp_function = interpolate.interp1d(abswave,abstemp)
abstemp_rebinned2 = abstemp_function(lambda_survey)

# plot original data
f1 = pylab.figure()
pylab.plot(abswave,abstemp, '.')
pylab.xlabel('wavelength')
pylab.ylabel('abstemp (original)')
```

```
# plot rebin with k_binspec
f2 = pylab.figure()
pylab.plot(lambda_survey, abstemp_rebinned, '.')
pylab.xlabel('wavelength')
pylab.ylabel('abstemp (k_binspec)')

# plot rebin with interp1d
f3 = pylab.figure()
pylab.plot(lambda_survey, abstemp_rebinned2, '.')
pylab.xlabel('wavelength')
pylab.ylabel('abstemp (interp1d)')

pylab.show()
```

History

#1 - 04/19/2013 08:33 AM - Brian Nord

- *Description updated*

We should be clear about the expectations. of the function. Although we've talked about this before, I'd like to be clear about it on public record. The integration practically will produce something that is slightly different from the original function: the new function is integrated on very small intervals, such that the units originally are photons/Angstrom, and then they become photons; the interval sizes are non-negligible, so the function shouldn't be exactly the same, but close.

#2 - 04/22/2013 08:13 AM - Ben Hambrecht

Noted. This is also why I'd like a clear documentation of the variables and their units in SGSO.

Additional information concerning the bug: the function works fine if the rebinding grid is finer than the original grid. The problem appears when the bins are larger than the data spacing, where an actual integration should be performed.