pyastro2018_synphot

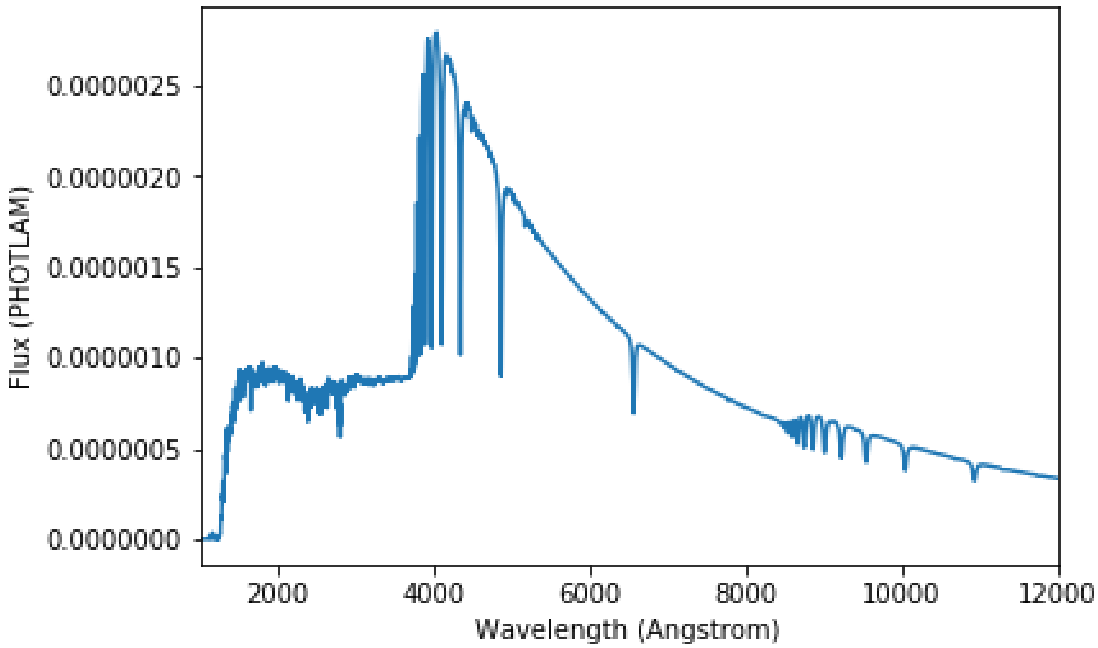
May 7, 2018

1 Synthetic Photometry with Astropy Models

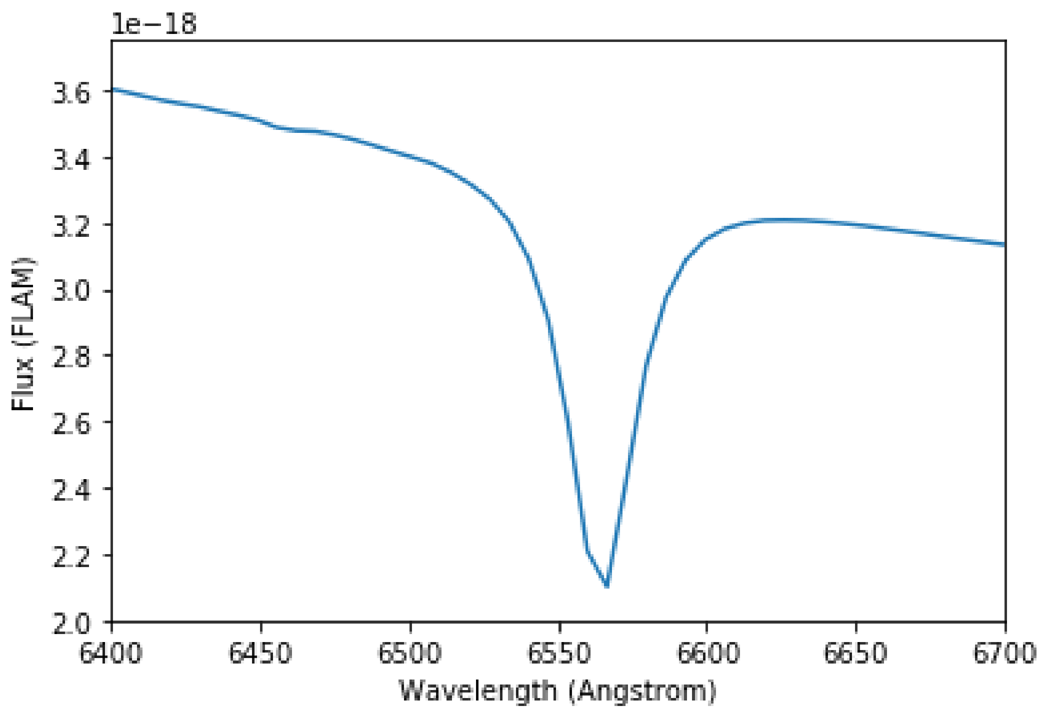
This is a short demo for "synphot" package (http://synphot.readthedocs.io/en/latest/), which does synthetic photometry using Astropy models. This 5-minute demo is for pyastro2018 (May 2018, NYC) lightning talk series.

This package is implemented in Python and not related to the IRAF task `synphot`.

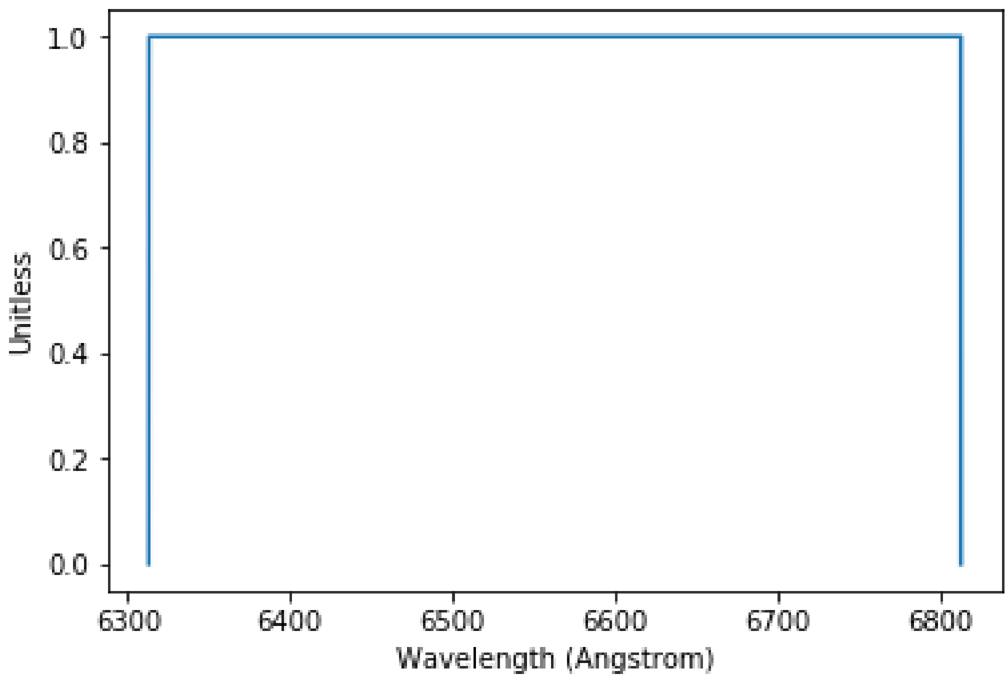
```
In [1]: from __future__ import division, print_function  # Only if you use Python 2

        %matplotlib inline

In [2]: from astropy import units as u
        from synphot import SourceSpectrum, SpectralElement

        # Load a Vega spectrum.
        # Data can be obtained from http://ssb.stsci.edu/cdbs/calspec/alpha_lyr_stis_008.fits
        datafile = 'alpha_lyr_stis_008.fits'
        vega = SourceSpectrum.from_file(datafile)

        # Normalize it to 22 STMAG in V-band.
        v_band = SpectralElement.from_filter('johnson_v')
        sp = vega.normalize(22 * u.STmag, v_band)

        # Visualize it.
        sp.plot(left=1000, right=12000)
```



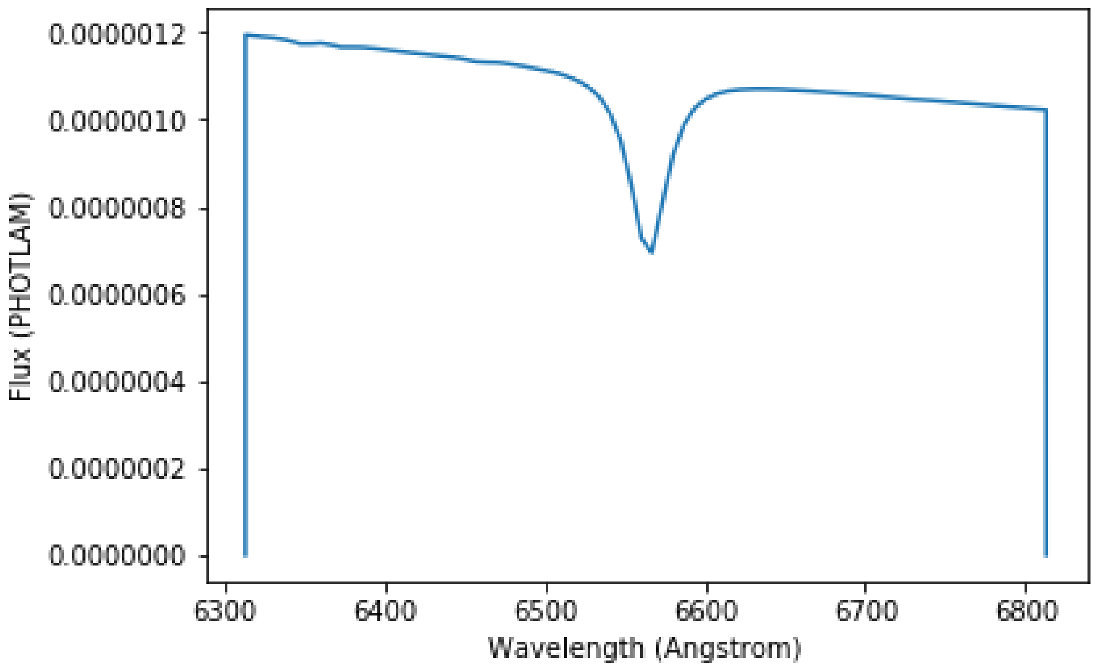
In [3]:
```
# Zoom in on feature of interest in a different flux unit.
sp.plot(left=6400, right=6700, bottom=2e-18, top=3.75e-18, flux_unit='flam')
```



```
In [4]: # This is identical to astropy.models.Box1D but with some extra properties.
        from synphot.models import Box1D

        # A boxy bandpass around feature of interest.
        bp = SpectralElement(Box1D, x_0=6563*u.AA, width=50*u.nm)
        bp.plot()
```




```
In [5]: from synphot import Observation

        # A simulated observation of Vega through the boxy bandpass.
        obs = Observation(sp, bp)
        obs.plot()
```



In [6]:
```
# Predicted count rate for given telescope (HST) collecting area.
obs.countrate(area=45238.93416*(u.cm*u.cm))
```

Out[6]:

$24.284528\ \frac{\mathrm{ct}}{\mathrm{s}}$

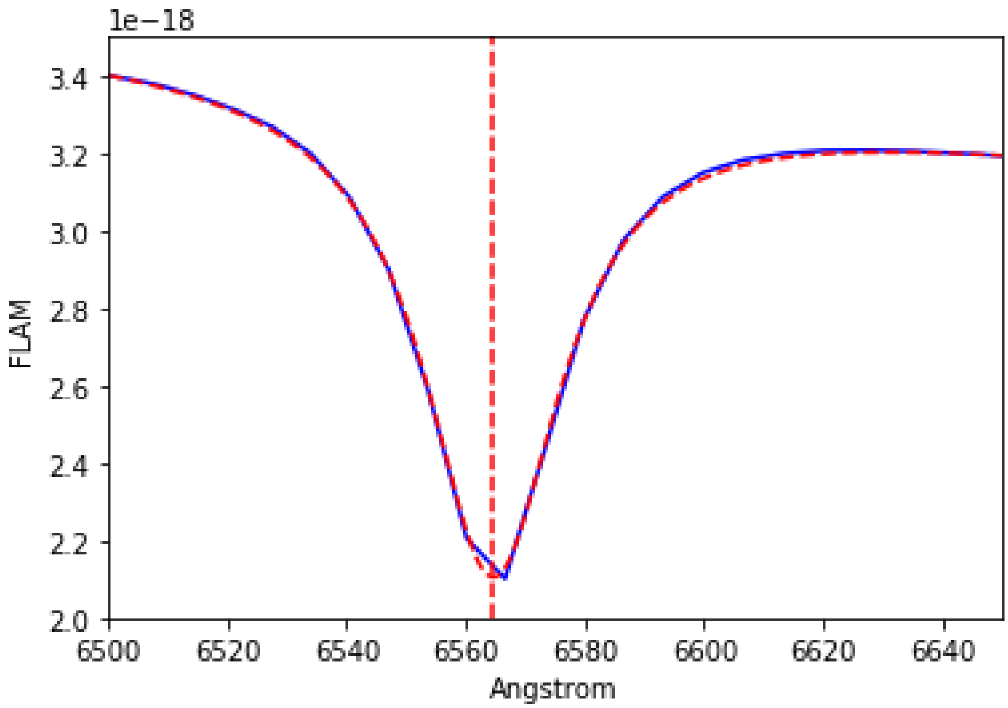
In [7]:
```
from astropy.modeling import models, fitting

# Build a composite model to fit observed feature of interest.
# Some reasonable initial guess is recommended.
bg = models.Linear1D()
ab = models.Lorentz1D(x_0=6560, amplitude=1e-18)
init_model = bg - ab

# Astropy models and fitting do not support units yet,
# so for now, we only use unitless portion for fitting.
x = bp.waveset.value  # Angstrom
y = obs(bp.waveset, flux_unit='flam').value  # FLAM

# Do the fitting.
fitter = fitting.LevMarLSQFitter()
fit_model = fitter(init_model, x, y)
y_fit = fit_model(x)

# Components only list initial guess.
```

```python
        # Parameters are the actual fitted values (background + line).
        print(fit_model)
```

```
Model: CompoundModel4
Inputs: ('x',)
Outputs: ('y',)
Model set size: 1
Expression: [0] - [1]
Components:
    [0]: <Linear1D(slope=1., intercept=0.)>

    [1]: <Lorentz1D(amplitude=0., x_0=6560., fwhm=1.)>
Parameters:
            slope_0              intercept_0        ...       fwhm_1
    ----------------------- ---------------------- ... ------------------
    -1.5390917855809249e-21 1.3461240425145207e-17 ... 27.931246779642834
```

```python
In [8]: import matplotlib.pyplot as plt

        # Plot the fitted model against observed data.
        fig, ax = plt.subplots()
        ax.plot(x, y, 'b')
        ax.plot(x, y_fit, 'r--')
        ax.set_xlim(6500, 6650)
        ax.set_ylim(2e-18, 3.5e-18)
        ax.set_xlabel('Angstrom')
        ax.set_ylabel('FLAM')

        # Fitted center of the absorption line.
        fitted_center = fit_model.x_0_1.value
        ax.axvline(fitted_center, ls='--', color='r')
        print(fitted_center)
```

```
6564.498146523282
```




```
In [9]: import math

        # Area inside curve.
        a_in_curve = (math.sqrt(2 * math.pi) * fit_model.amplitude_1 * fit_model.fwhm_1)

        # Approx. continuum level.
        h_at_center = fit_model.slope_0 * fitted_center + fit_model.intercept_0

        # Equivalent width.
        print('EW = {:.4f} Angstrom'.format(a_in_curve / h_at_center))
```

```
EW = 26.1135 Angstrom
```

1.0.1 stsynphot: HST specific add-on

Documentation at http://stsynphot.readthedocs.io/en/latest/

```
In [10]: # Need to download a bunch of data first
         import os

         os.environ['PYSYN_CDBS'] = 'C:\\Users\\lim\\cdbs\\grp\\hst\\cdbs\\'
```

```
In [11]: import stsynphot
```

```
WARNING: Failed to load Vega spectrum from C:\Users\lim\cdbs\grp\hst\cdbs\\calspec\alpha_lyr_st
```

```
In [12]: bp = stsynphot.band('acs,wfc1,f555w')
```

```
C:\Users\lim\AppData\Local\Continuum\Anaconda\envs\py36\lib\site-packages\stsynphot\stio.py:238
  if not np.issubdtype(data[key].dtype, val):
C:\Users\lim\AppData\Local\Continuum\Anaconda\envs\py36\lib\site-packages\stsynphot\stio.py:238
  if not np.issubdtype(data[key].dtype, val):
```

```
In [13]: bp.plot(left=4300, right=6500)
```