Sherpa Notebooks

Omar Laurino
Smithsonian Astrophysical Observatory
Chandra X-Ray Center

In this Talk

Sherpa Jupyter Notebooks

Sherpa Features

Packages using or extending Sherpa

[Some slides borrowed from A. Siemiginowska]

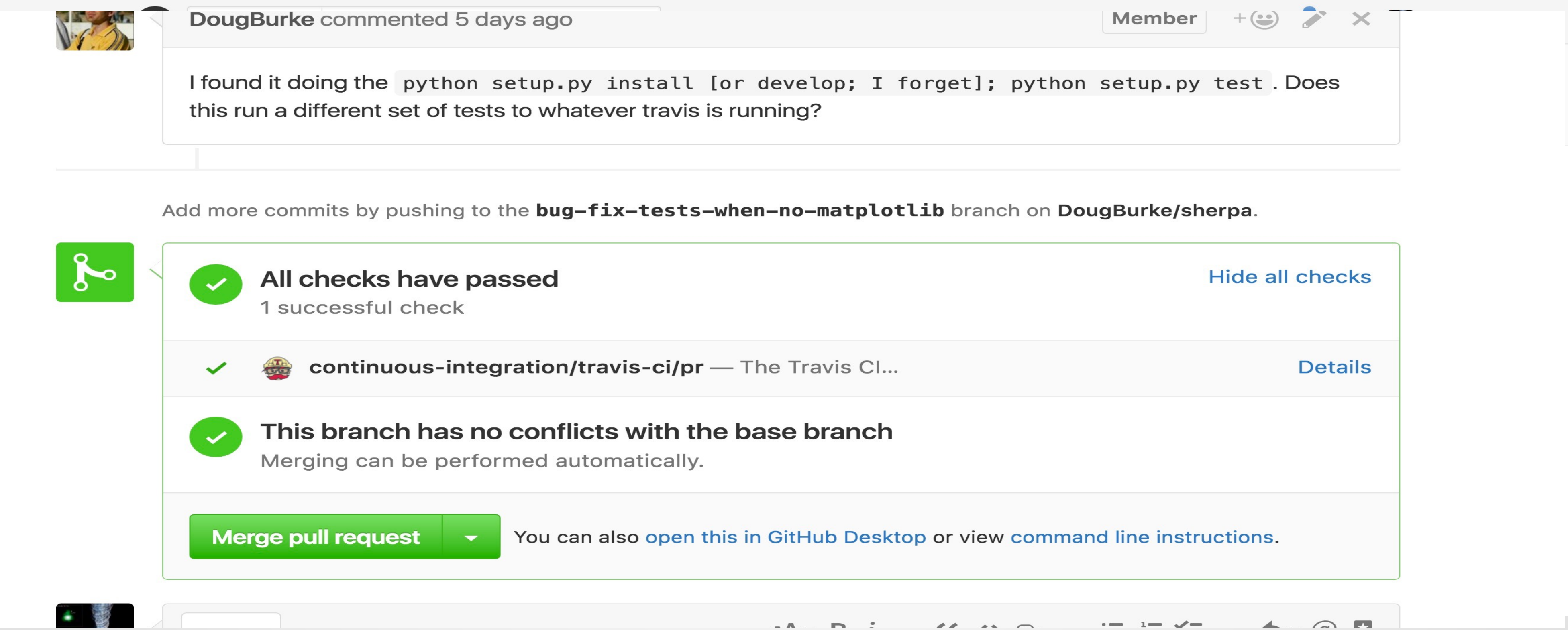
Open Development on GitHub

Core Team:

Omar Laurino, Doug Burke, Warren McLaughlin, Dan Nguyen, Aneta Siemiginowska

Code contributions:

Tom Aldcroft, Jamie Budynkiewicz, Christoph Deil, Brigitta Sipocz



DougBurke commented 5 days ago

I found it doing the `python setup.py install [or develop; I forget]; python setup.py test`. Does this run a different set of tests to whatever travis is running?

Add more commits by pushing to the **bug-fix-tests-when-no-matplotlib** branch on **DougBurke/sherpa**.

✓ **All checks have passed**
1 successful check

✓ continuous-integration/travis-ci/pr — The Travis CI...

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

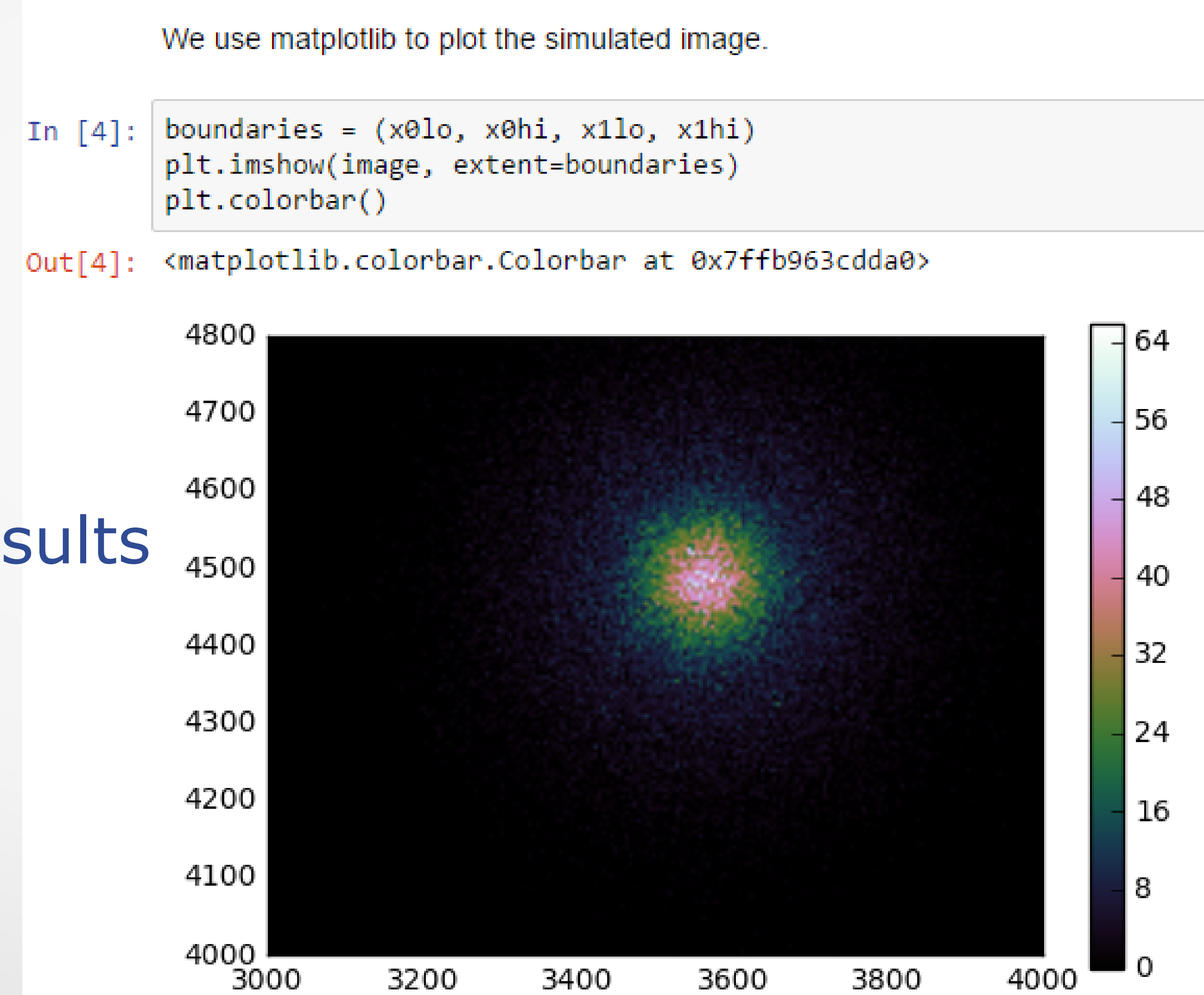
Jupyter Notebooks

Human Readable Rich Text

Computer Code

Output:

- tables
- figures
- computation results



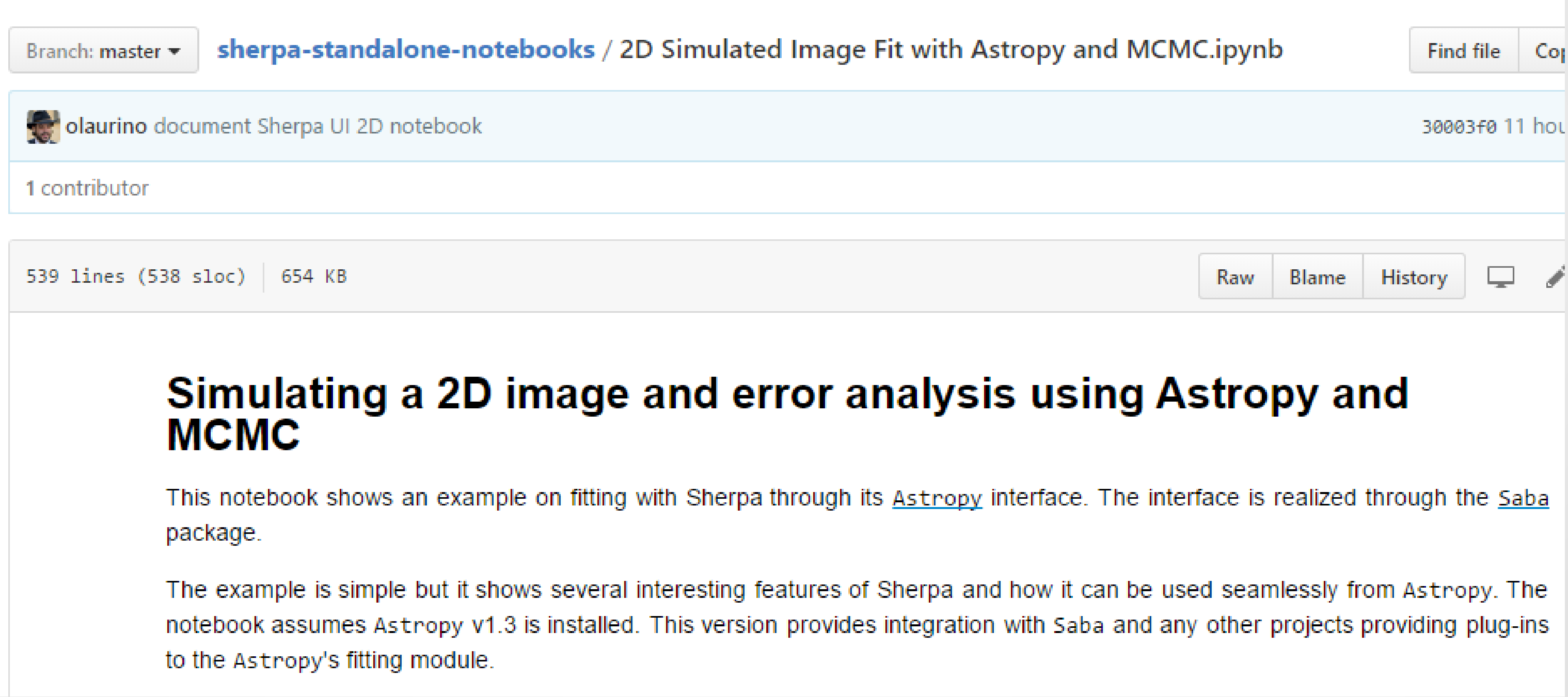
We use matplotlib to plot the simulated image.

```
In [4]:  boundaries = (x0lo, x0hi, x1lo, x1hi)
         plt.imshow(image, extent=boundaries)
         plt.colorbar()

Out[4]:  <matplotlib.colorbar.Colorbar at 0x7ffb963cdda0>
```

Browse and Read from GitHub

http://bit.ly/sherpa-github



sherpa-standalone-notebooks / 2D Simulated Image Fit with Astropy and MCMC.ipynb

olaurino document Sherpa UI 2D notebook 30003f0 11 hou

1 contributor

539 lines (538 sloc) | 654 KB

Simulating a 2D image and error analysis using Astropy and MCMC

This notebook shows an example on fitting with Sherpa through its Astropy interface. The interface is realized through the Saba package.

The example is simple but it shows several interesting features of Sherpa and how it can be used seamlessly from Astropy. The notebook assumes Astropy v1.3 is installed. This version provides integration with Saba and any other projects providing plug-ins to the Astropy's fitting module.

Run Locally

Build from Sources or Install Conda Binaries

Install your favorite software alongside Sherpa

Straightforward on Linux and macOS

Ask me about the prototype Docker image if interested!

Run in the Cloud

Live Notebooks with No Installation Required!

Run on any device, including Android/iOS

No Data Persistence, No Warranty, Demo only!!

http://bit.ly/sherpa-cloud (temporary AAS service)

http://bit.ly/sherpa-mybinder

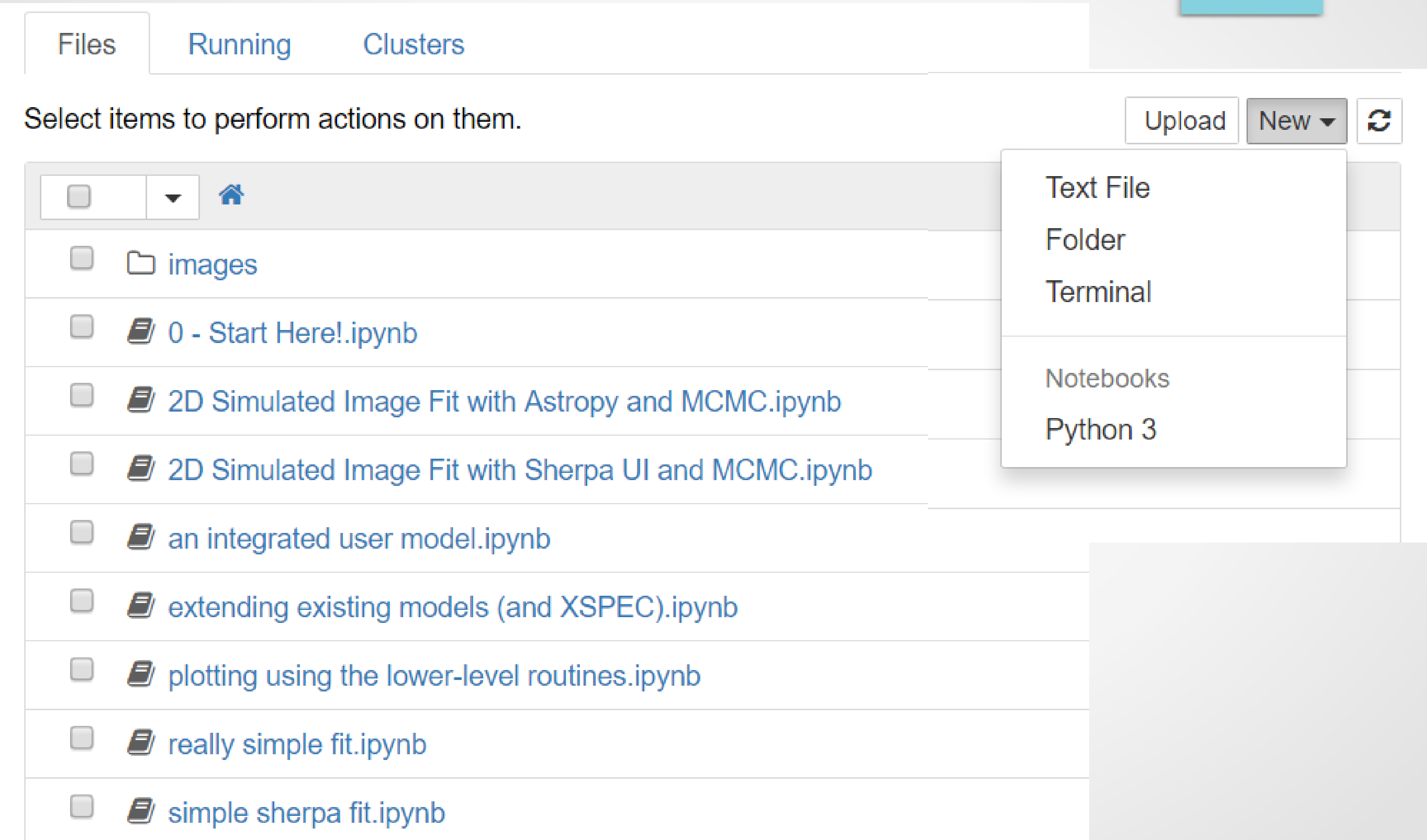
Evolving Documentation

Sherpa is well documented in CIAO, with examples tailored for X-Ray that we are migrating to a Pythonic format.

Our Jupyter notebooks aim to be multi-wavelength.

They are examples and do not exhaust all of the Sherpa capabilities.

Dashboard

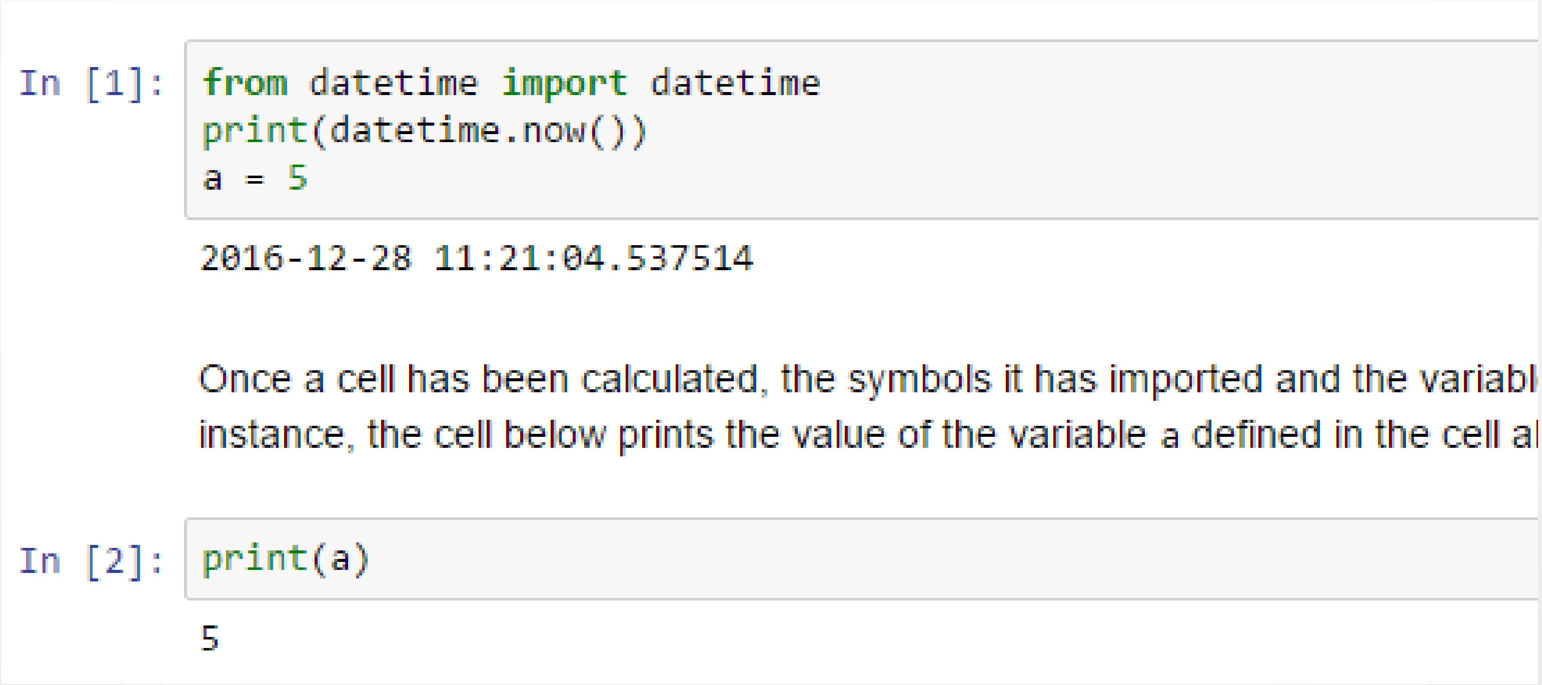


Select items to perform actions on them.

- □ images
- □ 0 - Start Here!.ipynb
- □ 2D Simulated Image Fit with Astropy and MCMC.ipynb
- □ 2D Simulated Image Fit with Sherpa UI and MCMC.ipynb
- □ an integrated user model.ipynb
- □ extending existing models (and XSPEC).ipynb
- □ plotting using the lower-level routines.ipynb
- □ really simple fit.ipynb
- □ simple sherpa fit.ipynb

0 – Start Here!

Gentle introduction to Jupyter notebooks



In [1]:
```
from datetime import datetime
print(datetime.now())
a = 5
```

```
2016-12-28 11:21:04.537514
```

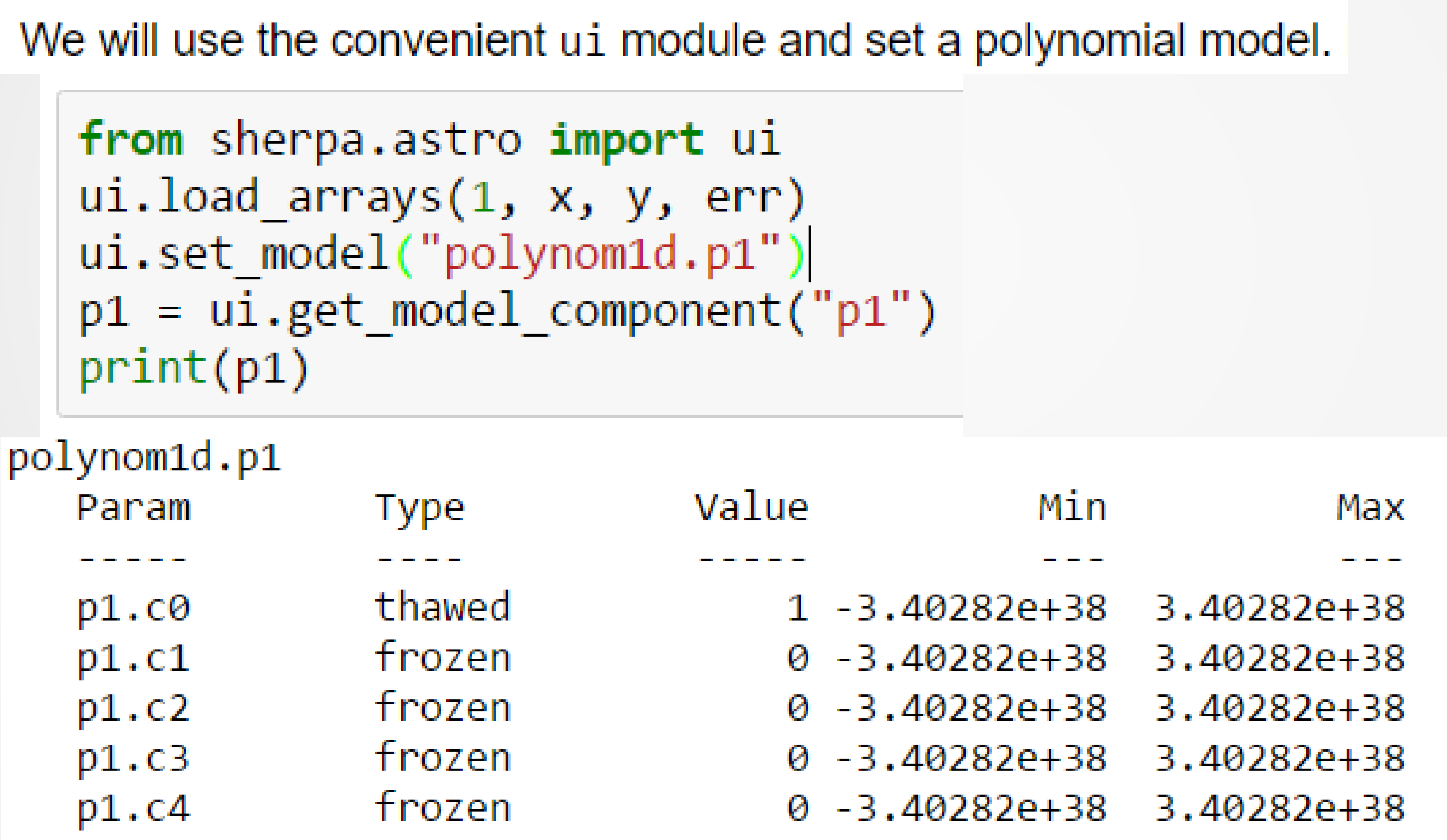
Once a cell has been calculated, the symbols it has imported and the variabl
instance, the cell below prints the value of the variable a defined in the cell al

In [2]:
```
print(a)
```

```
5
```

0 – Start Here!

Gentle introduction to Sherpa



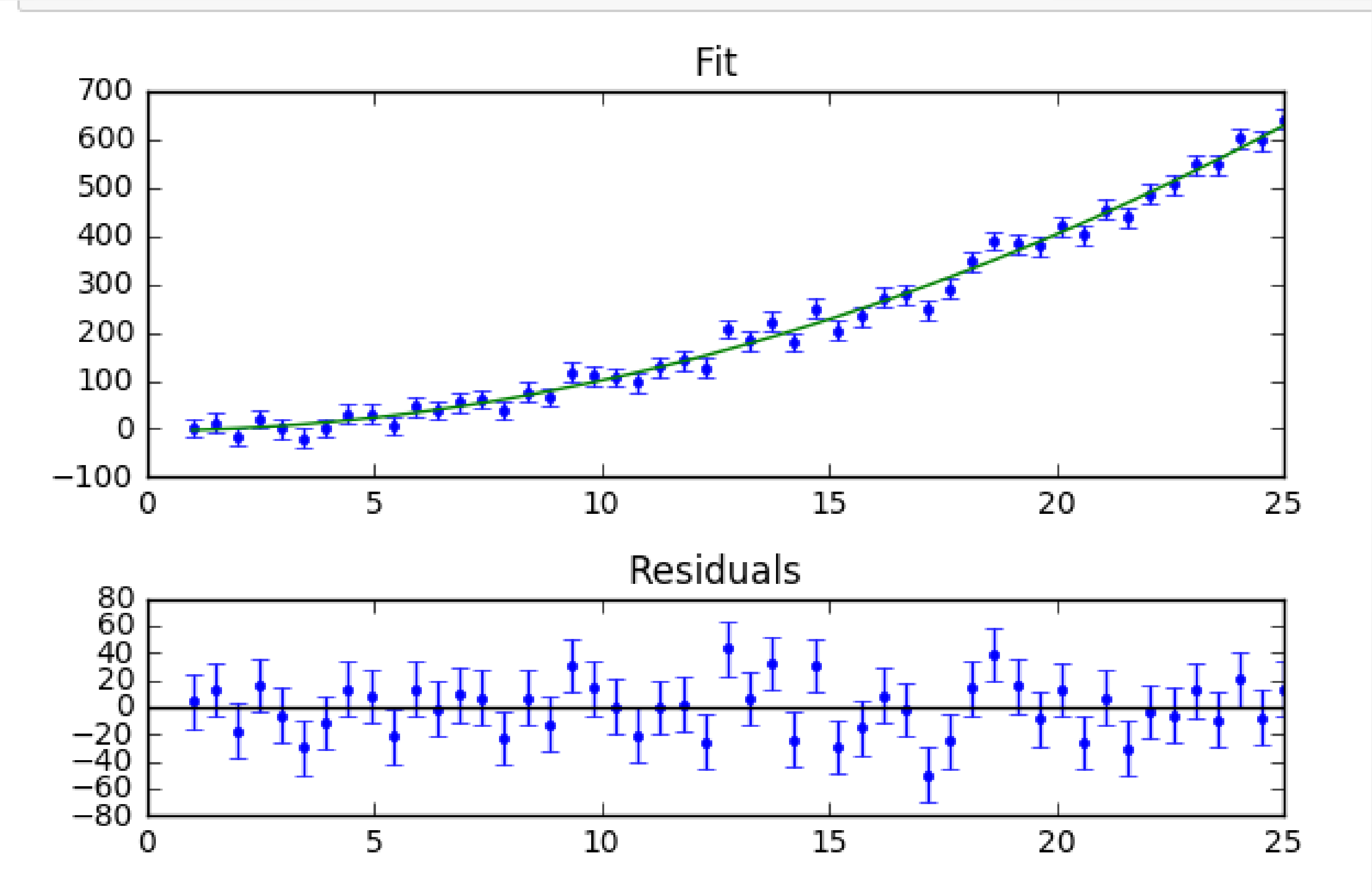
We will use the convenient `ui` module and set a polynomial model.

```python
from sherpa.astro import ui
ui.load_arrays(1, x, y, err)
ui.set_model("polynom1d.p1")
p1 = ui.get_model_component("p1")
print(p1)
```

```
polynom1d.p1
   Param        Type          Value          Min          Max
   -----        ----          -----          ---          ---
   p1.c0        thawed            1 -3.40282e+38  3.40282e+38
   p1.c1        frozen            0 -3.40282e+38  3.40282e+38
   p1.c2        frozen            0 -3.40282e+38  3.40282e+38
   p1.c3        frozen            0 -3.40282e+38  3.40282e+38
   p1.c4        frozen            0 -3.40282e+38  3.40282e+38
```

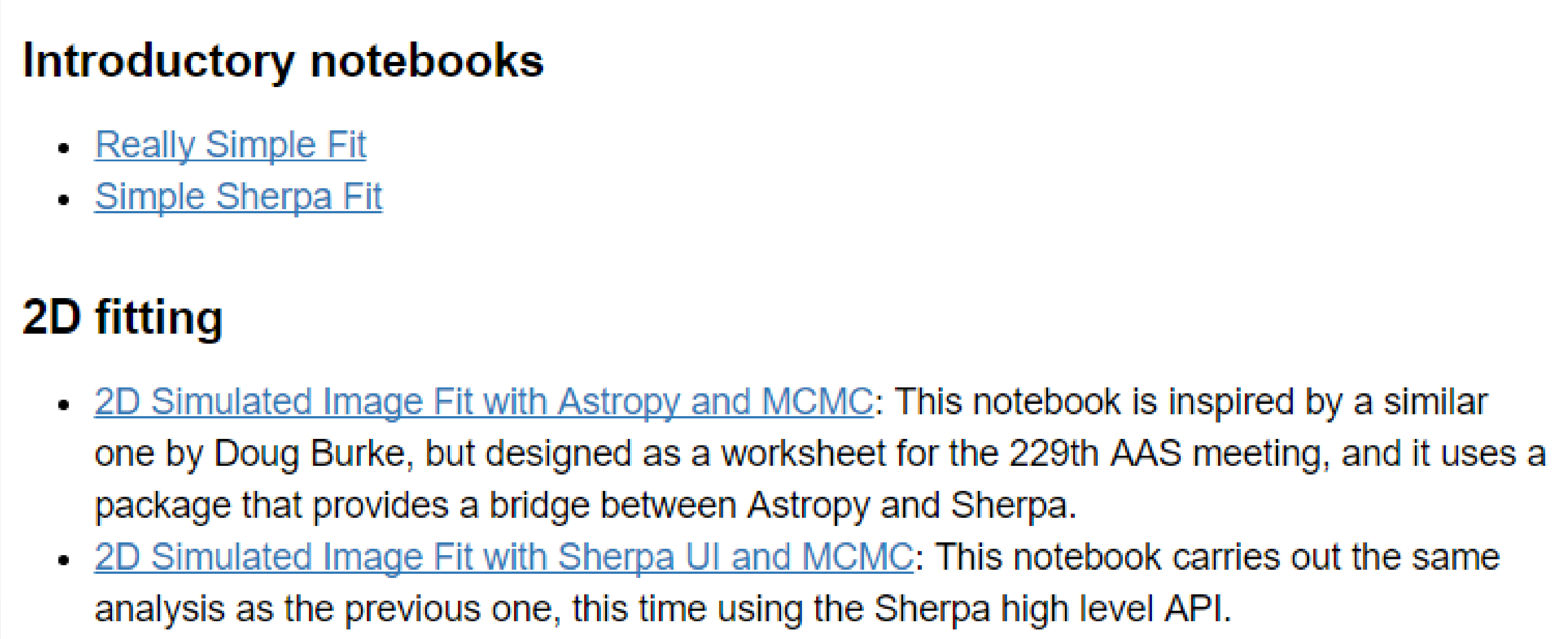
0 – Start Here!

Gentle introduction to matplotlib



0 – Start Here!

Introduction to Sherpa notebooks



Introductory notebooks

- Really Simple Fit
- Simple Sherpa Fit

2D fitting

- 2D Simulated Image Fit with Astropy and MCMC: This notebook is inspired by a similar one by Doug Burke, but designed as a worksheet for the 229th AAS meeting, and it uses a package that provides a bridge between Astropy and Sherpa.
- 2D Simulated Image Fit with Sherpa UI and MCMC: This notebook carries out the same analysis as the previous one, this time using the Sherpa high level API.

Research is complex

Data I/O and pre-processing

- Download 2D long-slit image off the web
- Remove cosmic rays
- Fit and subtract background
- Extract spectrum

Data Analysis

- Modeling
- Uncertainty estimation

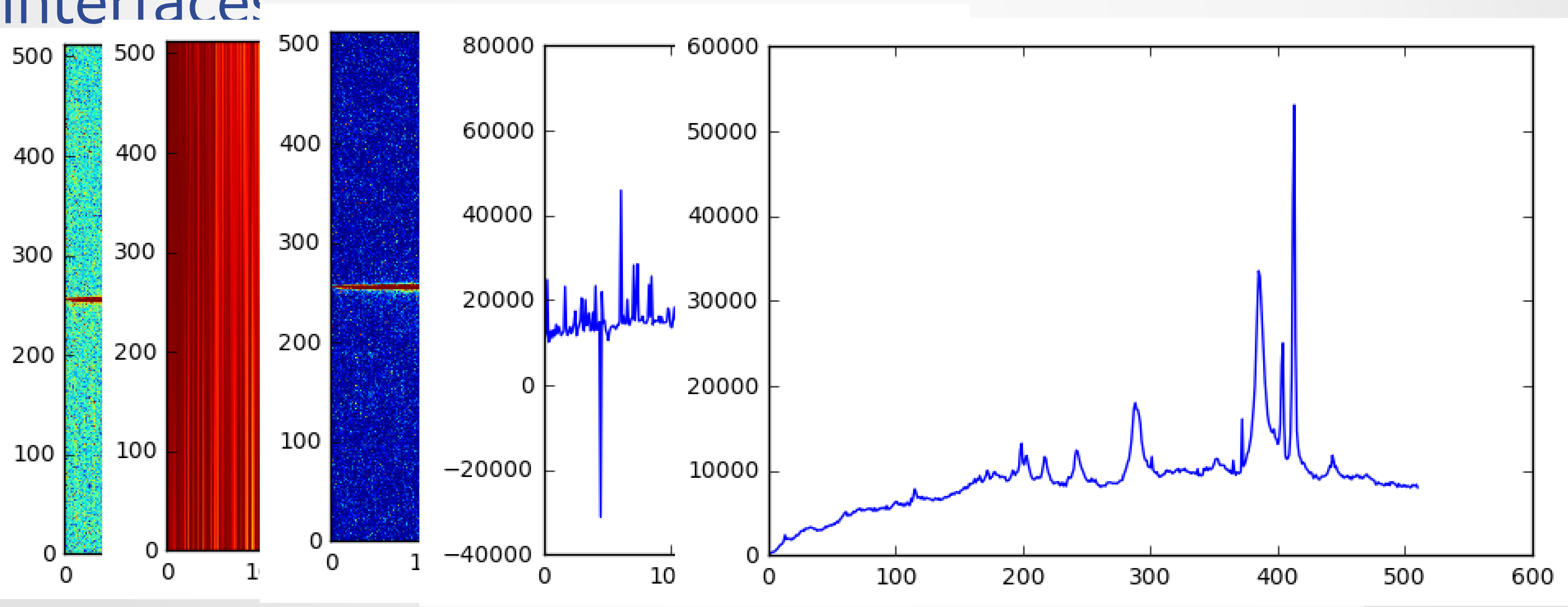
Extract new knowledge

No single tool can do everything

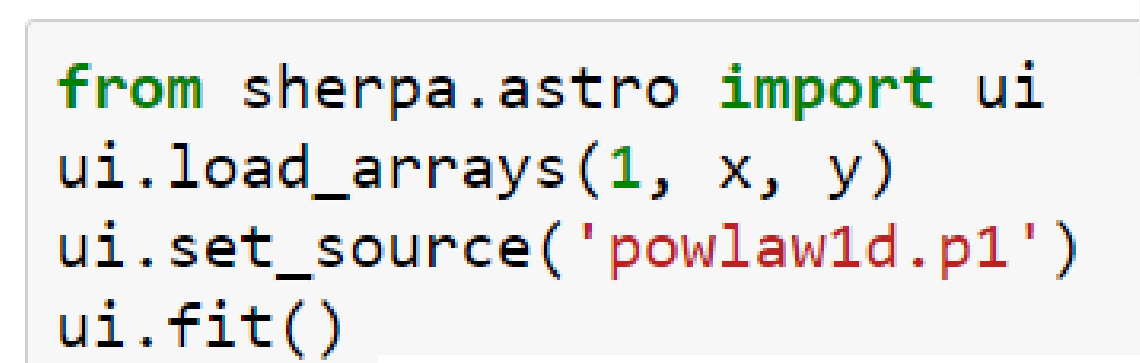
Simple Sherpa Fit

Fit and Subtract the Background from a 2D long-slit STIS spectrum.

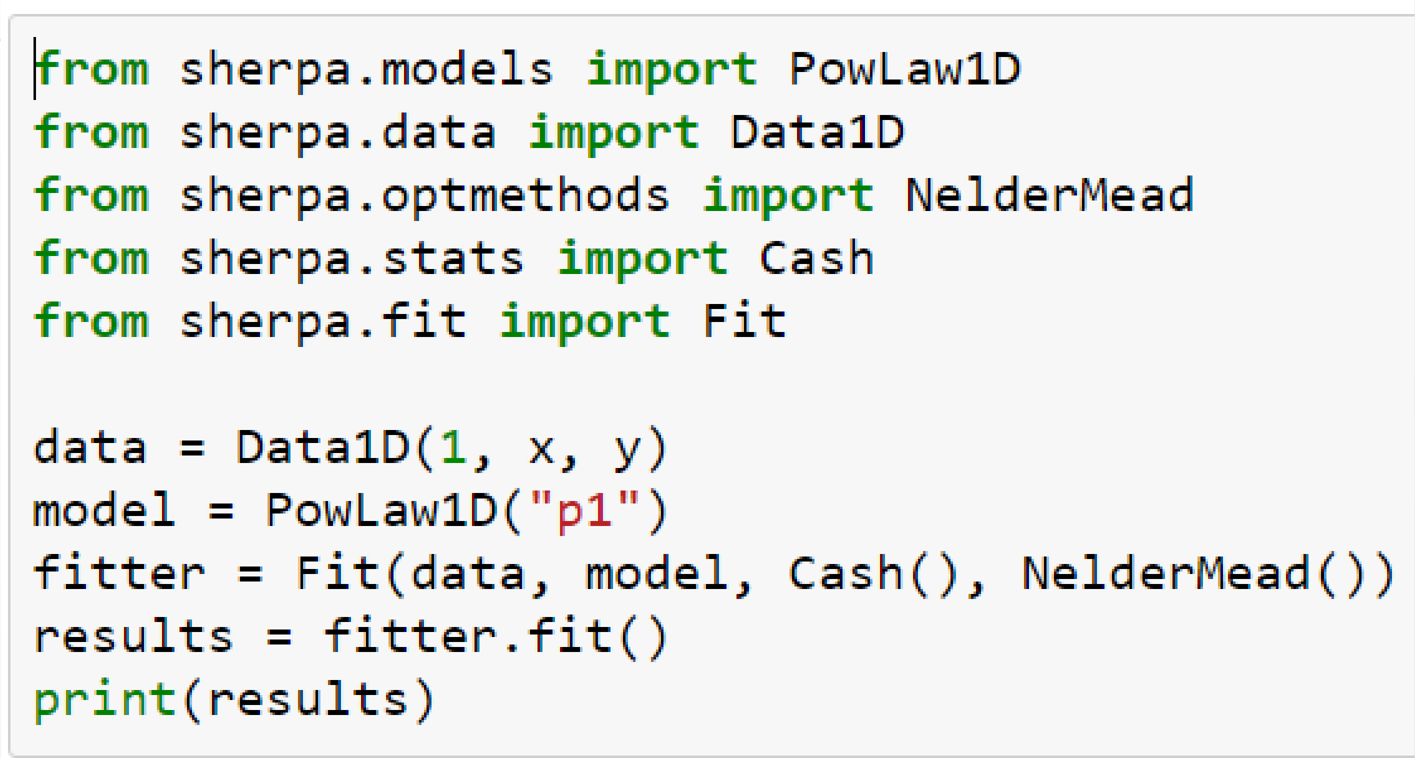
Illustrates both Procedural and Object Oriented interfaces



Procedural vs Object Oriented



```
from sherpa.astro import ui
ui.load_arrays(1, x, y)
ui.set_source('powlaw1d.p1')
ui.fit()
```



```
from sherpa.models import PowLaw1D
from sherpa.data import Data1D
from sherpa.optmethods import NelderMead
from sherpa.stats import Cash
from sherpa.fit import Fit

data = Data1D(1, x, y)
model = PowLaw1D("p1")
fitter = Fit(data, model, Cash(), NelderMead())
results = fitter.fit()
print(results)
```

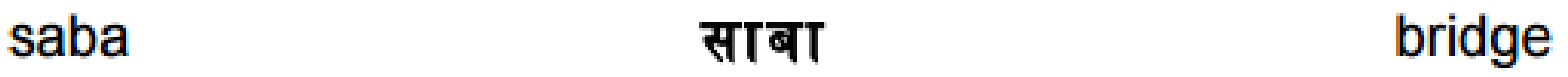
Integration with Astropy

Saba – Sherpa/Astropy Bridge

Google Summer of Code 2016

Install Sherpa, Astropy (v1.3), Saba, then:

```
from astropy.modeling.fitting import SherpaFitter
```



saba साबा bridge

Integration with Astropy

`astropy.modeling` package:

Weighted least square

No uncertainties

`astropy.modeling.fitting.SherpaFitter` class:

Select optimization algorithm

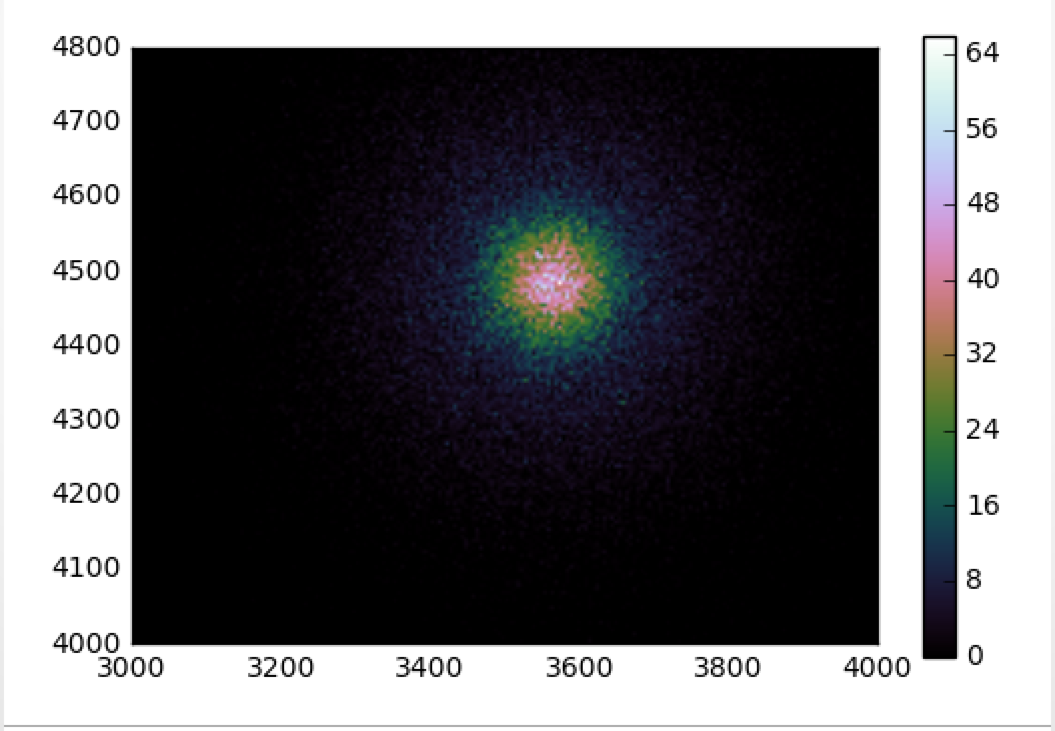
Configurable fit statistics

Estimate parameter confidence intervals, including coupled non-Gaussian errors.

MCMC sampler for exploration of posterior probability distribution

2D Simulated Image w/ Astropy

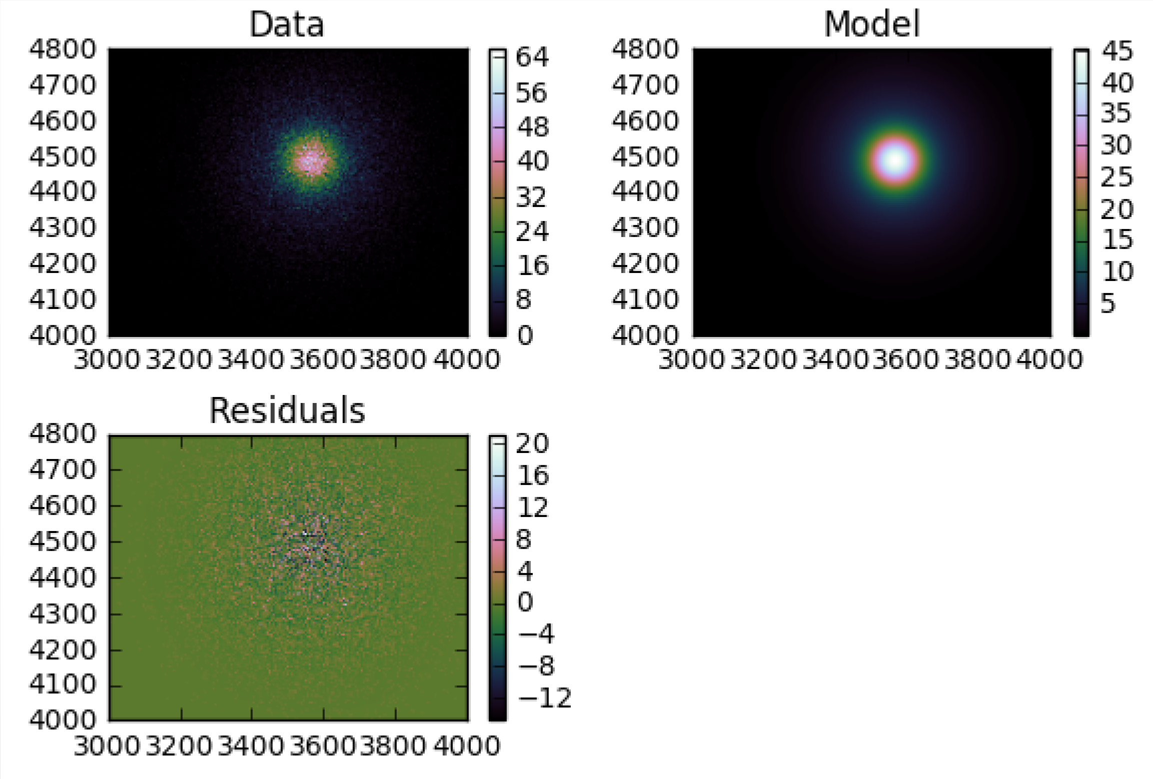
Simulate 2D image in Poissonian regime



2D Simulated Image w/ Astropy

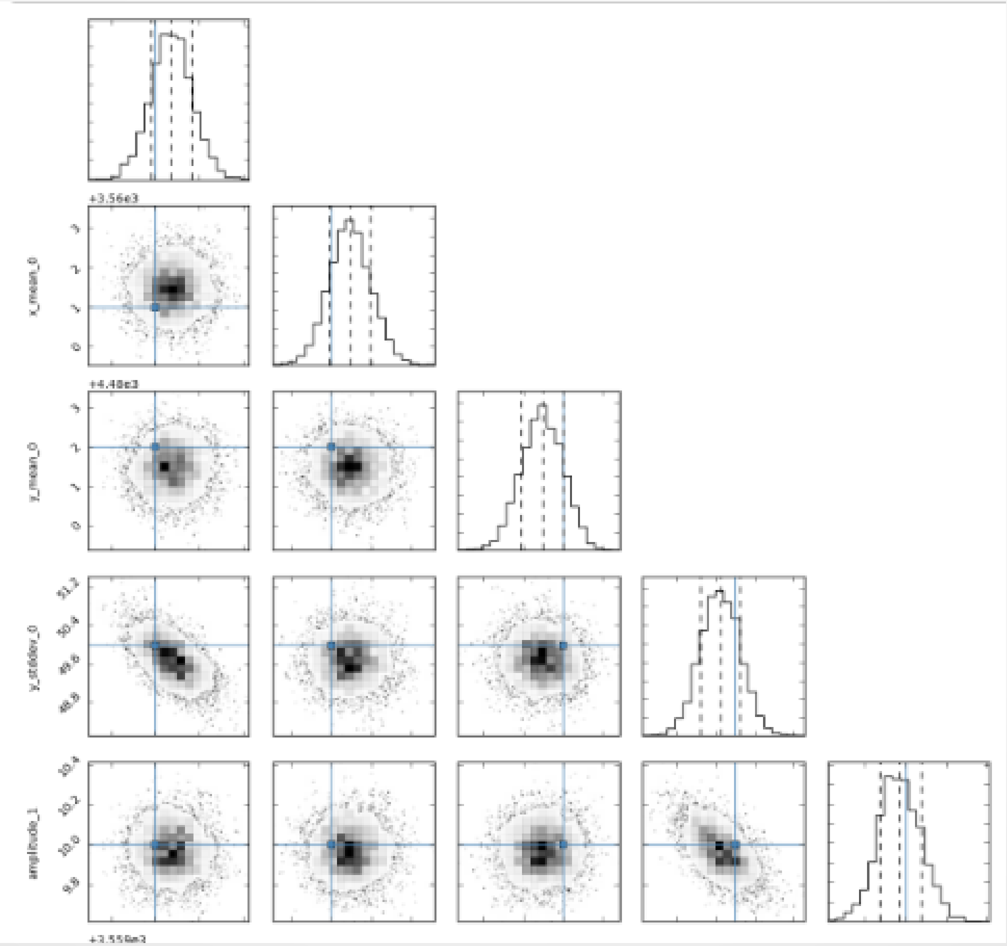
Guess initial parameter values from the data

Fit using Astropy models and Sherpa optimization



2D Simulated Image w/ Astropy

Estimate posterior distributions using MCMC



Models in Sherpa

Parameterized models:

Built-in (1-D, 2-D)

User-provided (N-D)

Convolution kernels

Model language to build compound model expressions

Combine built-in, user-provided, psf, templates (also with interpolation!)

Models in Sherpa

Instantiate models through strings:

```
atten.abs1  atten.abs2  powlaw1d.p1 powlaw1d.p2
from sherpa.astro import ui
[ … load datasets 1 and 2 … ]
ui.set_model(1, ‘atten.abs1 * atten.abs2 * powlaw1d.p1’)
ui.set_model(2, ‘abs1 * abs2 * powlaw1d.p2’)
```

Instantiate models through classes:

```
from sherpa.astro.optical import AbsorptionVoigt
line_one = AbsorptionVoigt(‘line_one’)
```

Combine convolved and unconvolved models:

```
ui.set_full_model(1, ‘psf(gauss2d.g2) + const2d.c1’)
```

User Models

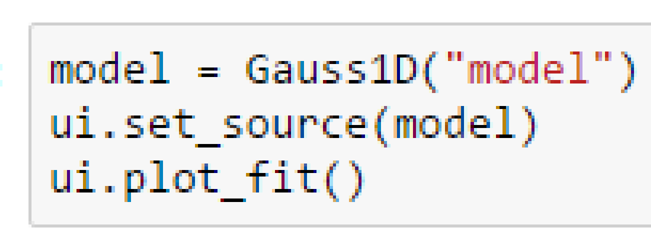
User model types

- Python function
- Python class
- Table model (x, y columns)
- Parametric templates library

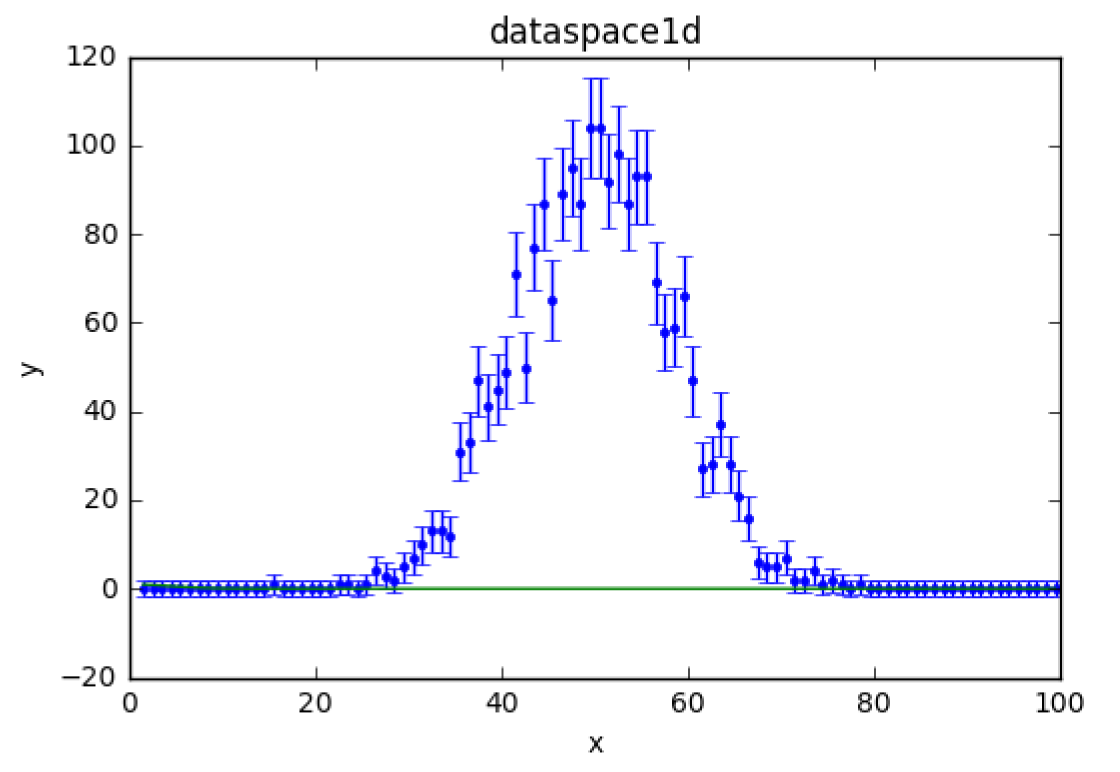
Sherpa model **classes** can know how to *guess* their parameters from the data.

Guessing parameters

Default instances have default, off parameter values, driving optimizer into local minima

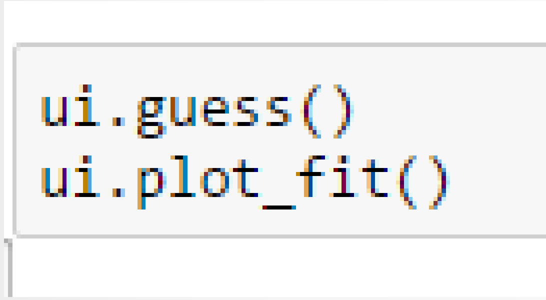


```
model = Gauss1D("model")
ui.set_source(model)
ui.plot_fit()
```

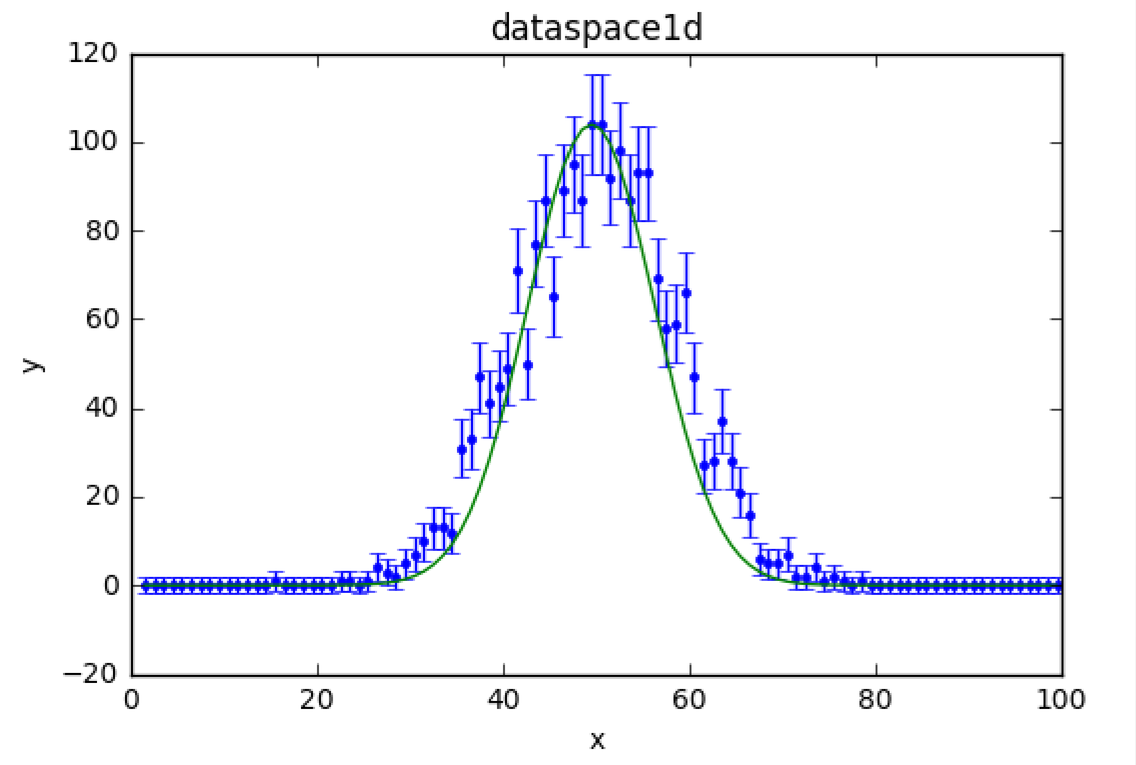


Guessing parameters

Self-guessing can fairly reduce modeling challenges



```
ui.guess()
ui.plot_fit()
```



Projects using/extending Sherpa

Saba/Astropy

BXA – Bayesian X-Ray Analysis

Connects the nested sampling algorithm MultiNest to Sherpa for Bayesian Parameter Estimation and Model comparison.

XMM-Newton Source Catalog

Web interface to spectral fitting of 3XMM-DR6 sources.

Projects using/extending Sherpa

Naima

Extend Sherpa models for Gamma Ray modeling

Gammapy

[We thank the] Sherpa developers and the Chandra X-ray observatory (CXC) for creating and maintaining a wonderful modeling / fitting package, and making Sherpa an open package on GitHub in 2015.

Iris

Multi-Wavelength SED Building (Jamie’s talk coming up!)

Summary

Sherpa is a flexible, robust, extensible modeling tool

Jupyter notebooks are great for documenting, prototyping, and sharing

Sherpa and its documentation are now openly developed, and we welcome contributions!

Example Notebook

Where to go next

Browse or run our notebooks!

On your system **http://bit.ly/aas-setup**

In the cloud **http://bit.ly/sherpa-cloud**

Browse them on GitHub **http://bit.ly/sherpa-github**

We have pen drives with complete software suite and examples.

Ask us questions!

Come visit us at the Chandra booth!