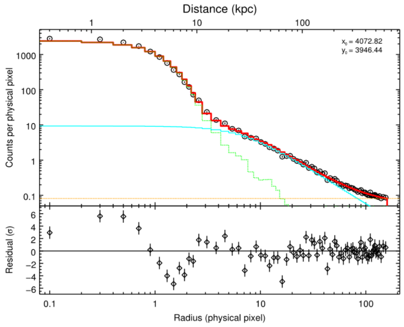
Modeling and Fitting in Python

Sherpa

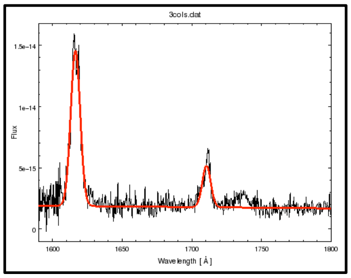
Sherpa enables the user to construct complex models from simple definitions and fit those models to data; optimization methods and fit statistics are used to characterize the results.

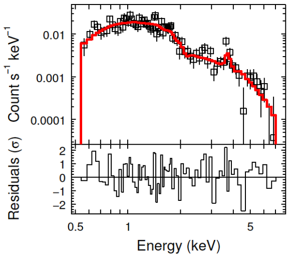
Sherpa can be run from the command line or imported as a Python module. Both options allow users to write their own Python extension code that could be contributed via GitHub.

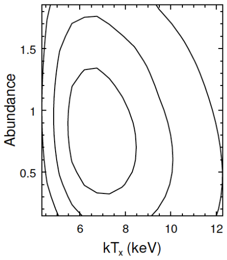


Surface brightness: model fit and residuals

Fitting optical and X-ray spectra







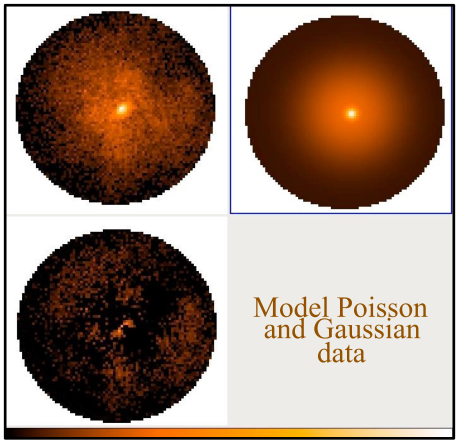


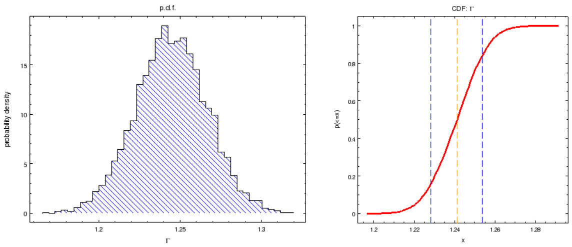
Image in ds9 of the data, model, and residuals

What you can do with Sherpa:

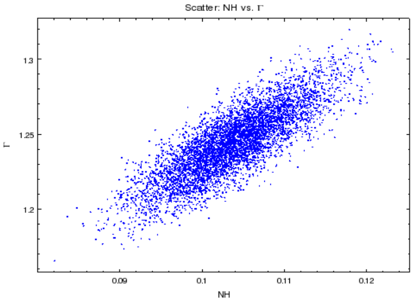
- **fit 1D data**: spectra, surface brightness profiles, light curves, general ASCII arrays
- **fit 2D images/surfaces** in the Poisson/Gaussian regime
- access the internal data arrays
- build **complex models**
- define **user models** including XSpec-style table models
- select **statistics** for modeling Poisson or Gaussian data
- calculate **confidence levels** on the best-fit parameters
- choose a **robust optimization method**: Levenberg-Marquardt, Nelder-Mead Simplex or Monte Carlo/ Differential Evolution
- **Bayesian analysis with Poisson Likelihood** using Metropolis or Metropolis-Hastings MCMC algorithm
- include **non-linear systematic errors** (calibration uncertainties) in the analysis
- **visualize a parameter space** with simulations or by using 1D/2D cuts of the parameter space

A full list of features in Sherpa is available online.

http://cxc.cfa.harvard.edu/sherpa/



Visualization of probability distribution



Code on GitHub:

https://github.com/sherpa/sherpa

Chandra X-ray Center