GalSim

Galaxy Simulation Software Package

Mike Jarvis

LSST Simulations Meeting

May 19, 2016

https://github.com/GalSim-developers/GalSim

Overview

- Simulates images of galaxies, stars, noise, etc.
- Originally designed for Great3 WL challenge.
- Can draw using FFTs, photon shooting, or real space convolution/rendering as appropriate.
- Was designed to have shapes accurate to 2e-4.
- Has a variety of options for all aspects of the rendering (PSF, galaxy profile, WCS, noise, etc.)

Code Design

- Open source.
- Modular structure (easy to add new features).
- Ability to trade between speed and accuracy.
- Intuitive Python layer wrapping fast C++.
- Even more intuitive configuration file interface.
- All code reviewed by typically 2 team members.

PSF Profiles

- Gaussian
- Moffat
- Kolmogorov
- Airy (with optional central obscuration)
- OpticalPSF (incl. Zernike to arbitrary order, obscuration, struts, or arbitrary spider pattern)
- InterpolatedImage (arbitrary image, e.g. PSFEx)

Galaxy Profiles

- Exponential
- DeVaucouleurs
- Sersic
- InterpolatedImage (can use any provided image as a starting profile)
- RealGalaxy (several available HST catalogs, properly deconvolving HST PSF)

Transformations

- Any profile can be:
  - sheared
  - dilated
  - rotated
  - shifted
  - scaled in flux
  - added to another profile
  - convolved with another profile
  - deconvolved by another profile (FFT only!)

Weak Lensing

- Realistic shear models
  - NFWHalo model
  - Cosmological (or other) PowerSpectrum
  - Can specify E an B power spectra separately
- These also include the corresponding convergence field so magnification and shear can be done consistently.

Random Numbers

- Can draw random numbers from:
  - UniformDeviate samples from uniform distribution
  - GaussianDeviate samples from normal distribution
  - PoissonDeviate handles Poisson number counts
  - other more exotic distributions (binomial, chi2, Weibull, gamma)
  - DistDevaite can use any arbitrary (user-provided) probability distribution
- "Random" numbers are deterministic given a seed, even when splitting job into multiple sub-jobs.

World Coordinate Systems

- Handles a variety of WCS transformations
  - PixelScale (the simple case of square pixels)
  - JacobianWCS (arbitrary local Jacobian defined by du/dx, du/dy, dv/dx, dv/dy)
  - AffineTransform (Jacobian with an arbitrary offset)
  - FitsWCS (read standard WCS types from FITS file)
  - UVFunction (any arbitrary function from (x,y) to tangent plane (u,v))
  - RaDecFunction (any arbitrary function from (x,y) to (RA, Dec))

Rendering Images

- Most calculations are deferred until image is rendered.
- Can use either FFT or photon shooting
  - Caveat: Photon shooting doesn't work when deconvolving. This includes RealGalaxy profiles.
- Can draw at arbitrary position in image.
- Can place postage stamps in tiles or randomly.
- Can set parameters that trade off accuracy vs speed.

Chromatic Profiles

- Chromatic is simplest case of a profile with a uniform SED.
- ChromaticAtmosphere will apply the correct shift and shear for DCR to a given (e.g. Kolmogorov) profile
- ChromaticAiry properly handles the λ/D parameter across the bandpass.
- ChromaticOpticalPSF properly scales the Zernike coefficients with wavelength (in addition to λ/D).
- Renders images by integrating over Bandpass.

Noise Models

- GaussianNoise
- PoissonNoise
- CCDNoise includes Poisson photon noise (with optional gain) and Gaussian read noise.
- VariableGaussianNoise (sigma different for each pixel)
- CorrelatedNoise
- Can also whiten images to remove existing correlated noise.

Detector Effects

- Brighter-fatter effect
    - Currently has Gruen implementation of Antilogus et al
    - Craig Lage is working on incorporating his physical model tracking individual electrons.
- Non-linearity
- Edge distortion and tree rings could be modeled with custom WCS, but nothing native currently.

Shape Measurement

- Includes HSM module for measuring shapes of galaxies.
- FindAdaptiveMom performs an adaptive moments measurement
- EstimateShear accounts for PSF convolution
  - KSB method
  - Re-Gaussianization (Hirata & Seljak, 2003)
  - ``Linear'' (H&S correction to Bernstein&Jarvis, 2002 method)

GalSim Demos

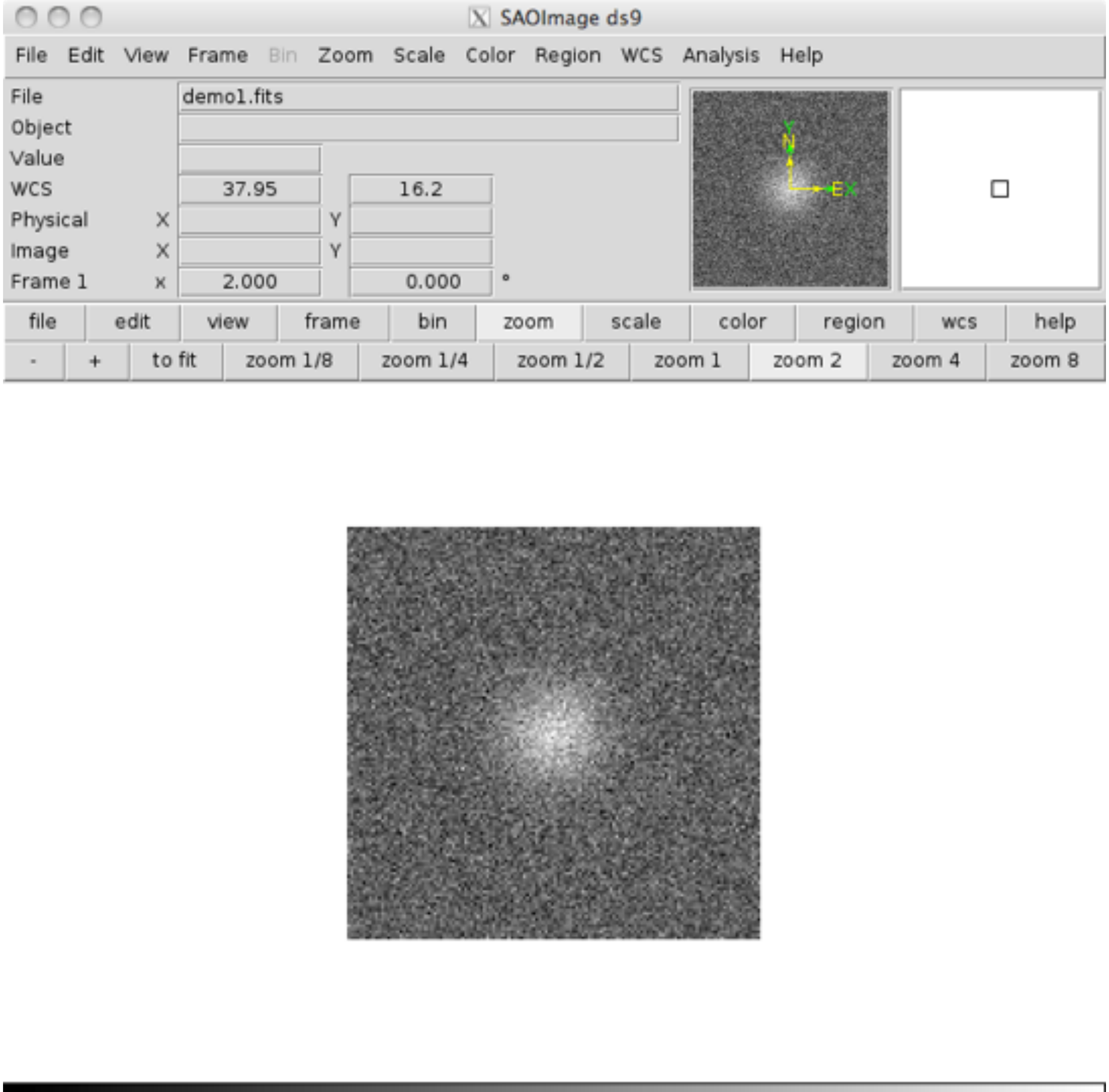
- Demo scripts are in GalSim/examples directory.
  - demo1.py ... demo13.py
  - demo1.yaml ... demo11.yaml
- After installing GalSim, you can run (for example)
  - python demo1.py
  - galsim demo1.yaml
- Most features are in at least one demo script.
- Python and YAML versions produce identical output files.

Demo1.py

```python
import galsim

gal = galsim.Gaussian(flux=1.e5, sigma=2)

psf = galsim.Gaussian(flux=1., sigma=1.)

final = galsim.Convolve(gal, psf)

image = final.drawImage(scale=0.2)

noise = galsim.GaussianNoise(sigma=30.)
image.addNoise(noise)

file_name = 'output/demo1.fits'
image.write(file_name)
```

Demo1.yaml

```yaml
gal :
    type : Gaussian
    sigma : 2
    flux : 1.e5

psf :
    type : Gaussian
    sigma : 1

image :
    pixel_scale : 0.2
    noise :
        type : Gaussian
        sigma : 30

output :
    dir : output_yaml
    file_name : demo1.fits
```



SAOImage ds9

File demo1.fits

Object

Value

WCS 37.95 16.2

Physical X Y

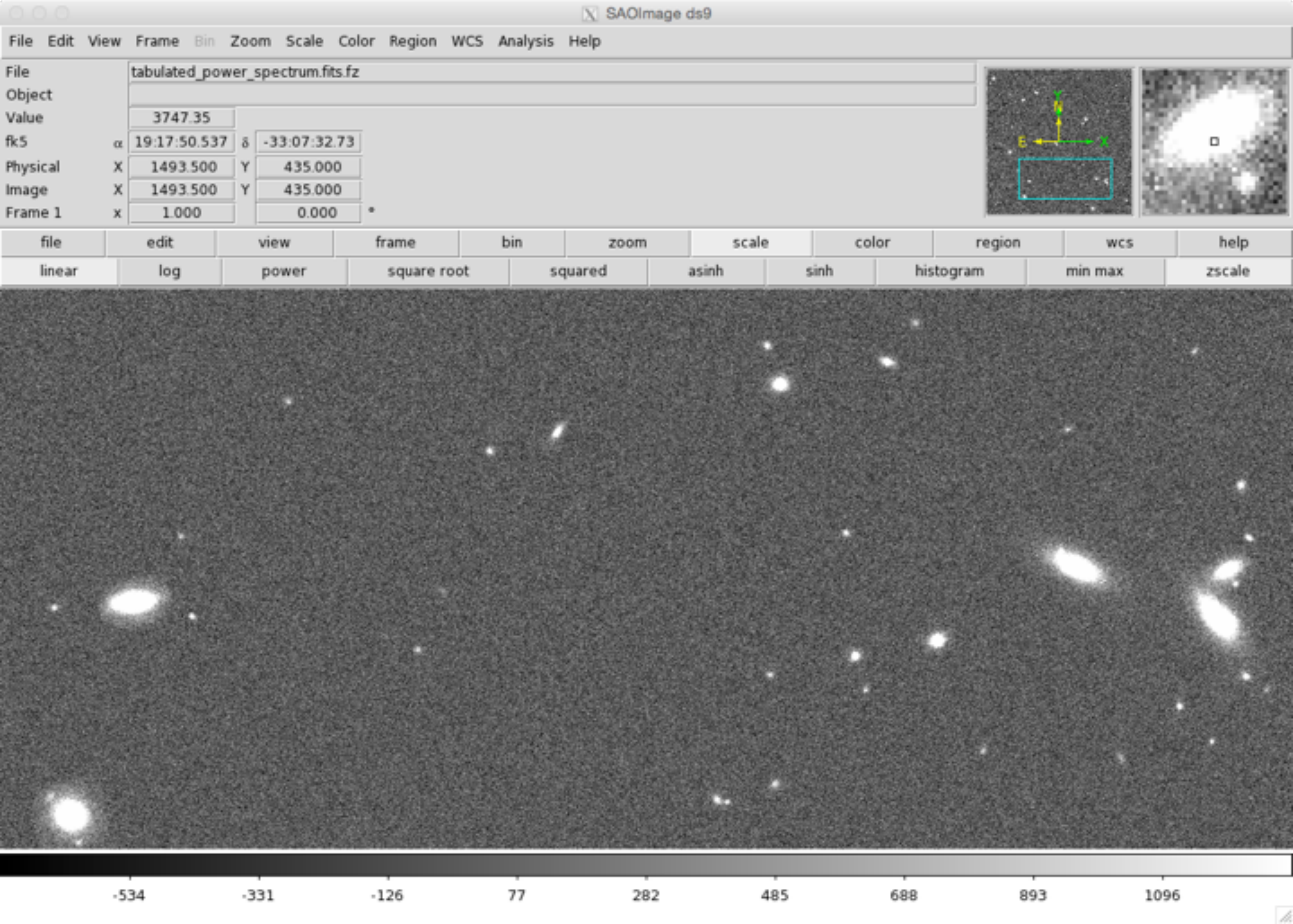
Image X Y

Frame 1 x 2.000 0.000 °

Demo11.yaml

```
eval_variables :
    fpixel_scale : &pixel_scale 0.2
    atheta : &theta 0.17 degrees
    ftel_diam : &tel_diam 4
    fexp_time : &exp_time 300
    fimage_size : &image_size 1800
    inobjects : &nobjects 288

psf :
    type : InterpolatedImage
    image : "data/example_sdss_psf_sky0.fits.bz2"
    scale : *pixel_scale

gal :
    type : COSMOSGalaxy
    gal_type :
        type : List
        items : [ 'parametric', 'real' ]
        index : { type : RandomBinomial, N : 1,  p : 0.3 }
    noise_pad_size : 11.3
    index : { type : Random }
    shear : { type : PowerSpectrumShear }
    magnification : { type : PowerSpectrumMagnification }
    rotation : { type : Random }
    scale_flux :
        type : Eval
        str : "(tel_diam**2 / (2.4**2*(1.-0.33**2))) * exp_time"

stamp :
    draw_method : no_pixel
```

```
image :
    type : Scattered
    size : *image_size
    nobjects : *nobjects
    index_convention : 0
    noise :
        type : Gaussian
        variance : 5.0e4   # Total variance including whatever the symmetrize process needs.
        symmetrize : 8
    wcs :
        type : Tan
        dudx : { type : Eval, str : 'math.cos(theta.rad()) * pixel_scale' }
        dudy : { type : Eval, str : '-math.sin(theta.rad()) * pixel_scale' }
        dvdx : { type : Eval, str : 'math.sin(theta.rad()) * pixel_scale' }
        dvdy : { type : Eval, str : 'math.cos(theta.rad()) * pixel_scale' }
        units : arcsec
        origin : center
        ra : 19.3 hours
        dec : -33.1 degrees
    random_seed : 24783923
    nproc : -1

input :
    cosmos_catalog :
        dir : "data"
        file_name : real_galaxy_catalog_example.fits
        use_real : True
    power_spectrum :
        e_power_function : "data/cosmo-fid.zmed1.00.out"
        units : radians
        grid_spacing : 90 # arcsec

output :
    dir : output_yaml
    file_name : tabulated_power_spectrum.fits.fz
```



File tabulated_power_spectrum.fits.fz

Object

Value 3747.35

fk5 α 19:17:50.537 δ -33:07:32.73

Physical X 1493.500 Y 435.000

Image X 1493.500 Y 435.000

Frame 1 x 1.000 0.000 °

LSST-Specific Features

- LSST WCS
  - Includes knowledge of chip layout as well as expected SIP component of each chip's WCS.
  - Currently in code review, will be in the next release, version 1.4.
  - Requires a LSST-DM stack installation to use.
- Nothing else yet.

Planned Features

- Chromatic RealGalaxy [#640]
  - Can take images from two or more observed bands and simulate a different band.
- Realistic Atmospheric PSF [#549]
  - Track phase screen from multiple atmospheric layers using correct Fourier optics
- Both are currently in code review.

Planned Features

- Physics-based models of brighter/fatter [#722]
  - Craig Lage is working on incorporating his physics-based simulation of the CCD silicon to improve the implementation of brighter/fatter.
  - Intrinsically photon-shooting method, but for FFT methods, we essentially draw the image and then photon shoot that.
  - For now, achromatic, but will add chromaticity.

Planned Features

- Model of LSST PSF [#556]
  - Spider pattern
  - Reference values for aberrations.
  - Approximate rms values of aberrations.
  - Model of how they will vary (correlated?) across the field of view.
  - Election diffusion in CCD

Planned Features

- Chromatic photon shooting [#540]
- Add chromatic options to config module [#510]
- Proper field dependence of aberrations [#716]
- Vignetting, non-uniform QE [#553]
- Extinction, opacity [#541, #550]
- Realistic galaxy profiles from hydro simulations [#669]

Planned Features

- Automatically chose FFT for bright objects and photon shooting for faint objects [#209]
- More image artifacts
  - cosmic rays
  - saturation
  - bad/hot pix, columns, etc.
  - cross-talk
  - charge transfer inefficiency
- Lots more: https://github.com/GalSim-developers/GalSim/issues