- Jesisim v1.0
- Jedisim v2.0
- Jedisim v3.0
  - 1: Generate Physics Config file
  - 2: Interpolate given sed file
  - 3: Create bulge and disk factors (to create scaled galaxies at redshift z)
  - 4: Create scaled galaxies (sb, sd, and sm to be used by jedimaster TDCR)
  - 5: Create bulge and disk weights (for psf at redshift z)
  - 6: Get fraction of scaled_bulge and scaled_disk (for psfm)
  - 7: Create PSF for bulge, disk, and mono
  - 8: Jedisim simulations (lsst, lsst_mono)

Jesisim v1.0

This version of jedisim is obsolete and no longer in use.

Jedisim v2.0

This program **Jedisim** takes in bulge and disk components HST ACS f814w filter images (f814w_gal*.fits) which has pixscale 0.06 and and creates a realistic set of output images with pixscale 0.2 for LSST r band filter.

The bulge and disk component are created using galfit program.

```
def run_7programs_loop():
    for i in range(0, 21):
        run_process("jedicolor", ['./executables/jedicolor',
        run_process("jeditransform"]
        run_process("jedidistort",
        run_process("jedipaste", ['./executables/jedipaste',
        run_process("jediconvolve",
        run_process("jedipaste", ['./executables/jedipaste',
        run_process("jedirescale",


def average21_and_add_noise():
    config = update_config()
    run_process("jediaverage", ['./executables/jediaverage']
    run_process("jedinoise", ['./executables/jedinoise']
    run_process("jedinoise", ['./executables/jedinoise']
```

Jedisim v3.0

The new file structure is given below

- a01_jedisim_config.py
- a02_interpolate_sed.py
- a03_scaled_bd_factors.py
- a04_scaled_gals.py
- a05_bd_weights_psf.py
- a06_scaled_bd_flux_rat.py
- a07_psf_bdmono.py
- a08_jedisim_odirs.py
- a09_jedisim_3cats.py
- a10_jedisimulate.py
- a11_jedisimulate90.py
- jedisim.py
- run_jedisim.py
- util.py

Terminologies: NUMGAL = Number of base galaxies used. (We have used 201 galaxies here. 0-200 inclusive) BDM = Bulge, disk, and monochromatic.

1: Generate Physics Config file

The script `a01_jedisim_config.py` reads the config file `physics_settins/config_template.sh` and creates config files for bulge, disk, and monochormatic cases for a given redshift. It will create three text files: - physics_settings/configb.sh - physics_settings/configd.sh - physics_settings/configm.sh

2: Interpolate given sed file

The script `a02_interpolate_sed.py` reads two SED files, each for bulge and disk, then interpolates the flux with wavelength step size of 1 Angstrom and writes out the interpolated sed files. Our original input SED files have flux for 1 to 12 Gyr ages. For a given redshift we first estimate the age of the star using flat LambdaCDM model of cosmology.

For example, using flat LambdaCDM model for redshift 1.5 galaxy which was assumed to be born at redshift 4.0 , we have following statistics:

```
Age of Universe for z =  0.0 is  13.47 Gyr
Age of Universe for z =  1.5 is  4.20 Gyr
Age of Universe for z =  4.0 is  1.52 Gyr
Difference                   is  2.68 Gyr
Age of Galaxy   for z =  1.5 is     3 Gyr
```

Here, we have rounded off the age of the galaxy to nearest integer. Then from the input SED file, we will take the flux column for the 3Gyr age galaxy for the redshift 1.5. Similarly, we get the flux for other redshift values.

This program will takes in two SED files (bulge and disk) and gives out interpolated sed files.

Inputs are: `sed/ssp_pf.cat` `sed/exp9_pf.cat`

and outputs are: `sed/ssp_pf_interpolated_z1.5.csv` `sed/exp9_pf_interpolated_z1.5.csv`

The output interpolated sed file has one wavelength column and two flux columns. The first column is the wavelength in Angstrom unit. The second column is the flux for a given redshift (e.g. 1.5). The third column is the flux for 12 Gyr old galaxy. Here, in the above example for the redshift 1.5, second column is the flux column of age 3 Gyr and third column is the last column of original input sed file. The second column of flux changes for redshifts but the last column is always the 12 Gyr flux. ## 3: Create bulge and disk factors (to create scaled galaxies at redshift z) The script `a03_scaled_bd_factors.py` creates a text file containing the values for bulge factor and disk factors to be used by next program `a04_scaled_gals.py`

3a: find the flux ratio of LSST to HST

$$f_{ratb} = \frac{\int_{\lambda 0}^{\lambda 20} f_{bz}(\lambda) d\lambda}{\int_{\lambda hst0}^{\lambda_{hst20}} f_{bzcut}(\lambda) d\lambda} \tag{1}$$

$$f_{ratd} = \frac{\int_{\lambda 0}^{\lambda 20} f_{dz}(\lambda) d\lambda}{\int_{\lambda hst0}^{\lambda_{hst20}} f_{dzcut}(\lambda) d\lambda} \tag{2}$$

To calculate the integrals we first choose the relevant wavelenghts for lsst and hst from these websites: https://www.lsst.org/about/camera/features http://www.stsci.edu/hst/acs/documents/handbooks/current/c05_imaging2.html

We divide the given wavelenths by 1 + z and use the interpolated sed file to evaluate the integrals. For redshift 1.5, the new wavelengths now looks like this:

```
laml0  = 5520 / (1 + z)  # 2208.0
laml20 = 6910 / (1 + z)  # 2764.0
lamh0  = ( 8333 - (2511/2) ) / (1 + z_cutout) # 7077.5 / 1.2 = 5897.9 = 5898
lamh20 = ( 8333 + (2511/2) ) / (1 + z_cutout) # 9588.5 / 1.2 = 7990.4 = 7990
```

We get the flux column from the interpolated sed files. The first column of interpolated sed file is for lsst and the second column is for hst.

The original sed file has flux for galaxies of age 1Gyr to 12 Gyrs. For example for exponential disk we have the fluxes for wavelenght 1000 Angstrom is shown below:

```
# Wavelen 1Gyr         2Gyr         3Gyr         4Gyr         5Gyr
# lambda  flux[0]      flux[1]      flux[2]      flux[3]      flux[4]
1000    2.125075e-05 1.905875e-05 1.706275e-05 1.527475e-05 1.36735e-05

# Wavelen 6Gyr       7Gyr         8Gyr        9Gyr         10Gyr     11Gyr        12Gyr
# lambda  flux[5]    flux[6]      flux[7]     flux[8]      flux[9]   flux[10]     flux[11]
1000    1.224e-05  1.095675e-05 9.8095e-06 8.78425e-06 7.868e-06 7.04975e-06 6.319e-06
```

For the lsst we first find the age of the galaxy from the redshift. For example for redshift of 1.5 we found the age of galaxy to be 3 Gyr.

```
Age of Universe for z =  0.0 is  13.47 Gyr
Age of Universe for z =  1.5 is  4.20 Gyr
Age of Universe for z =  4.0 is  1.52 Gyr
Difference                   is  2.68 Gyr
Age of Galaxy   for z =  1.5 is     3 Gyr
```

So we choose the 3 Gyr galaxy, namely flux[2] from this sed file. For the HST case we always choose the last 12 Gyr flux column.

Once we found the wavelenghts and flux columns then we calculated the above integrals.

3b: Find the total fluxes of bulge, disk and hst for all the galaxies:

$$F_b = [F_{b0}, F_{b1}, ..., F_{b200}] \tag{3}$$
$$F_d = [F_{d0}, F_{d1}, ..., F_{d200}] \tag{4}$$
$$F_{hst} = [F_{b0} + F_{d0}, F_{b1} + F_{d1}, ..., F_{b200} + F_{d200}] \tag{5}$$
$$\tag{6}$$

Here F_{b0} is the sum of total pixels of bulge0.fits. Similarly, F_{d0} is the sum of total pixels of disk0.fits. And F_{hst0} is sum of F_{b0} and F_{d0}.

3c: Find scaled value of total flux for all the HST images:

$$F_{hstscale} = F_b * f_{ratb} + F_d * f_{ratd} \tag{7}$$

Then we calculate the correction factor for the flux:

$$F_{cor} = \frac{F_{hst}}{F_{hstscale}} \tag{8}$$

Then we multiply this correction with the flux ratio to get the bulge and disk factors:

$$bf = F_{cor} * f_{ratb} \tag{9}$$
$$df = F_{cor} * f_{ratd} \tag{10}$$
$$\tag{11}$$

Here, bf and df are columns with NUM_GAL (e.g. equal to 200) rows. We write these two columns into a text file `physics_settings/bd_factors.txt` according to the config file.

Now, in the next script `a04_scaled_gals.py` we create scaled_bulge and scaled_disk fitsfiles using these factors:

$$scaled\_bulge0.fits = bf0 * bulge0.fits \tag{12}$$
$$scaled\_disk0.fits = df0 * disk.fits \tag{13}$$
$$\tag{14}$$

In this way we will have NUM_GAL (e.g. equal to 200) scaled galaxies from the next scipt `a04_scaled_gals.py`.

The inputs and outputs of this script are following:

```
:Inputs:
  sed/ssp_pf_interpolated_z1.5.csv
  sed/exp9_pf_interpolated_z1.5.csv


:Outputs: physics_settings/bd_factors_z1.5.txt # depends on redshift
```

4: Create scaled galaxies (sb, sd, and sm to be used by jedimaster TDCR)

This script `a04_scaled_gals.py` reads the bulge factor and disk factor created from previous script `a03_bd_factors_scaled.py`. Then mulitplies these factors to the bulge and disk galaxies and creates scaled_bulge and scaled_disk galaxies. It also sums up these two scaled galaxies to create scaled_bulge_disk (also called scaled_mono) galaxies.

In the actual simulation, the final names are following: `simdatabase/scaled_bulge_f8/f814w_scaled_bulge0.` `simdatabase/scaled_disk_f8/f814w_scaled_disk0.fits` `simdatabase/scaled_bulge_disk_f8/f814w_sc` `Upto 200.fits.`

5: Create bulge and disk weights (for psf at redshift z)

The script `a05_bd_weights_psf.py` takes in interpolated sed files and creates a text file called `physics_settings/bd_weights_z1.5.txt` containing the weights for bulge and disk for a given redshift.

The formula to calculate bulge and disk weights is given below:

$$b[0] = \frac{\int_{\lambda 0}^{\lambda 1} f_b(\lambda) d\lambda}{\int_{\lambda 0}^{\lambda_{20}} f_b(\lambda) d\lambda} \tag{15}$$

$$d[0] = \frac{\int_{\lambda_0}^{\lambda_1} f_d(\lambda) d\lambda}{\int_{\lambda_0}^{\lambda_{20}} f_d(\lambda) d\lambda} \tag{16}$$

Here the wavelengths used are for LSST r band. For example, for redshift z = 1.5 the wavelengths used are:

$$\lambda_0 = \frac{5520}{1+z} = 2208.0 \tag{17}$$

$$\lambda_{20} = \frac{6910}{1+z} = 2764 \tag{18}$$

$$\tag{19}$$

We divide these wavelength range into 21 parts and call them narrowbands.

The fluxes used here (fb and fd) are the interpolated sed files having three columns (wavelength, flux_zAge_Gyr, flux_12Gyr) and we are interested only in the middle column.

The interpolated sed files are `sed/exp9_pf_interpolated_z1.5` and `sed/ssp_pf_interpolated_z1.5`.

In the end of this program, we will get a text file `physics_settings/bd_weights_z1.5.txt` with 21 rows and two columns.

6: Get fraction of scaled_bulge and scaled_disk (for psfm)

We use the script `a06_scaled_bd_flux_rat.py` to find the ratio of the total flux of all scaled bulge files (201 files) to the total flux of all the scaled disk files.

To create monochromatic psf we define a flux ratio quantity f_r as

$$f_r = \frac{\sum(\frac{F_{sb}}{F_{sd}})}{n_g} \tag{20}$$

Here, F_{sb} is the flux of the given scaled bulge, F_{sd} is the flux of the given scaled disk and n_g is number of galaxies. For example n_g = 201.

Then, we calculate disk part and bulge part of f_r as:

$$f_{rd} = \frac{1}{1 + f_r} \tag{21}$$

$$f_{rb} = \frac{f_r}{1 + f_r} \tag{22}$$

For example, for redshift z = 1.5 I got the values: `fr = 0.0022, frb = 0.0021, and frd = 0.99785444`.

In the end we get a text file `physics_settings/bd_flux_rat.txt` which has only two numbers.

7: Create PSF for bulge, disk, and mono

We use the script `a07_psf_bdmono.py` to create the PSF for the scaled bulge, scaled disk and monochromatic images of the galaxy. I.e.

$$p_b = \frac{b0 * p0 + b1 * p1 + ... + b20 * p20}{b0 + b1 + ... + b20} \tag{23}$$

$$p_d = \frac{d0 * p0 + d1 * p1 + ... + d20 * p20}{d0 + d1 + ... + d20} \tag{24}$$

$$p_m = f_{rd}\ p_d + f_{rb}\ p_b \tag{25}$$

Here all the narrowbands PSFs p0, p1, . . . , p20 are all normalized and have the same total flux.

In the end, we create three psf files `psf/psfb.fits`, `psf/psfd.fits`, `psf/psfm.fits`.

8: Jedisim simulations (lsst, lsst_mono)

After we create scaled galaxies and scaled PSFs (for bulge, disk and monochromatic images) for HST images, we run the `jedisim` simulation to create the LSST and LSST_monochromatic images.

The jedisimulation program consist of four sub programs - `a08_jedisim_odirs.py` - `a09_jedisim_3cats.py` - `a10_jedisimulate.py` - `a11_jedisimulate90.py`

I combined all these four programs and call it `jedisim.py`. If we run this script `jedisim.py`, we will get two important outputs: -

`jedisim_out/out0/scaled_bulge_disk/trial1_lsst.fits # chromatic image, and, - jedisim_out/out0/scaled_bulge_disk/trial1_lsst_mono.fits # monochromatic image .`

We also get 90 degree rotated version of these outputs.

In addition to these two main outputs some other outputs are convolved_scaled_bulge, convolved_scaled_disk, catalog.txt, and, dislist.txt. We also keep the three psf files.

The three psf files are same for given redshift. However, other files changes in each run of the program `jedisim.py`.

So, I create another runner program `run_jedisim.py` which runs the main program `jedisim.py` for a given number of times and copies the outputs to a user given directory.

For example if we run `run_jedisim.py` for 1 iteration, it will copy 3 outputs files (4 galaxies, 3 psfs, 2 textfiles) into a new folder.

The process of copying is like this: - from: jedisim_out/out0/scaled_bulge_disk/trial1_lsst.fits - to: jedisim_output/jout_z1.5_2017_Oct05_17_13/z1.5/lsst0.fits # number increases each time

Mechanism of Jedisim Program: We have three folders scaled_bulge, scaled_disk, and scaled_bulge_disk inside simdatabase and each folder contains NUM_GAL (201) galaxies. We also have 3 psf files inside the psf directory and 3 configuration files inside physics_settings folder.

First we run the jedisim routine `lsst_TDCR` to get three convolved scaled fitsfiles, which we call them gcsb, gcsd and gcsm.

Note that the routine lsst_TDCR consists of four subrotunes transform, distort, convolve, and rescale.

The **transform** routine will transform scaled_bulge or scaled_disk fitsfiles according to `jedisim_out/out0/scaled_bulge/trial1_catalog.txt`. It will create 12,420 .gz fitsfiles. For example, `jedisim_out/out0/scaled_bulge/stamp_0/stamp_0_to_999.fits.gz` inside 13 folders `jedisim_out/out0/scaled_bulge/stamp_0 to stamps_12`. Jeditransform will also create a dislist file for the jedidistort, which can be found at `jedisim_out/out0/scaled_bulge/trial1_dislist.txt`.

Then the **distort** routine will distort the 12,420 galaxies from `jedisim_out/out0/scaled_bulge/stamp_0/ to 12` according to dislist.txt and lens.txt. It will write 12,420 unzipped fitsfiles inside the 13 folders `jedisim_out/out0/scaled_bulge/distorted_0/ to 12`. Then we combine these distroted images into a large file called `jedisim_out/out0/scaled_bulge/trial1_HST.fits` using the routine `jedipaste`.

We use the routine **convolve** to convolve this `HST.fits` file with the `psfb.fits`. First we will get 6 convolved bands (e.g. `jedisim_out/out0/scaled_bulge/convolved/convolved_band_0.fi` and we will combine them using `jedipaste` and get `jedisim_out/out0/scaled_bulge/trial1_HST_convolved` In case of bulge we call this file g__cb. Similarly we get g__cd and g__cbd.

We use the routine **rescale** to change the PIXSCALE of HST (0.06) to the pixscale of LSST (0.2) to get the convolved-scaled fitsfiles. (e.g. gcsb = `jedisim_out/out0/scaled_bulge/trial1_lsst_bulge.fits`) and so on.

$$
\begin{aligned}
g_{csb} &= g_{cb} \otimes p_b && (26)\\
g_{csd} &= g_{cd} \otimes p_d && (27)\\
g_{csm} &= g_{cbd} \otimes p_m && (28)\\
& && (29)
\end{aligned}
$$

Now we have three convolved-scaled images. We add the Poisson noise to the g__cbdm and call it lsst__monochromatic file.

This means for the folder `simdatabase/scaled_bulge_disk` after running `lsst_TDCR` with `configm.sh` we will get `jedisim_out/out0/scaled_bulge_disk/trial1_lsst_bulge_disk.f` and we add Poisssion noise to this using `jedinoise` and we get `lsst_mono` file `jedisim_out/out0/scaled_bulge_disk/trial1_lsst_mono.fits`.

To get LSST chromatic image, we first add g__csb and g__csd, then add the noise to it and call it lsst.fits.

This means first we combine two convolved rescaled files `jedisim_out/out0/scaled_bulge/trial1_lsst_bulge` and `jedisim_out/out0/scaled_disk/trial1_lsst_disk.fits` to get the unnoised file `jedisim_out/out0/scaled_bulge_disk/trial1_lsst_unnoised.fits`. We add the noise to this file using `jedinoise` and get the main output lsst monochromatic file as `jedisim_out/out0/scaled_bulge_disk/trial1_lsst.fits`

Schematically we can write:

$$
\begin{aligned}
monochromatic &= g_{csm} + Noise && (30)\\
chromatic &= (g_{csb} + g_{csd}) + Noise && (31)\\
& && (32)
\end{aligned}
$$

Outputs of Jedisim Program: For a given redshift we have three scaled psf files p_b, p_d, and p_m. In a single run of `jedisim` we will have four galaxies and two text files:

- g_{csb}, g_{csd} (output of lsst__TDCR for bulge and disk config file).
- g_{chro}, g_{mono} (Main outputs 1 and 2).
- catalog.txt, dislist.txt (catalog files).

We also have 90 degree rotated version of these 6 files.

Trivia: To convert markdown to pdf

```
# First replace  by whitespace in markdown file, then run pandoc
pandoc -o README.pdf README.md; open README.pdf
```