This file documents the details of the simulation.

List of functions:

a. 1D Compton-y profile

```
def battaglia_profile(r, Mvir, z)
```

Input: radius (Mpc), virial mass (m_{solar}), redshift, ect.
Output: Electron pressure profile (KeV/cm^3)
Library: colossus
Paper: https://arxiv.org/abs/1109.3711 (Battaglia, 2012)
https://arxiv.org/abs/1608.04160 (Vikram, 2016)
https://ui.adsabs.harvard.edu/abs/1986MNRAS.222..323K/abstract (Kaiser, 1986)
https://arxiv.org/abs/astro-ph/0410173 (Voit, 2005)
Steps:

1. Calculate ρ_{cri} using Battaglia Eq.1
2. Convert M_{vir} to M_{200} & R_{200} using colossus (Concentration: Ishiyama, 2021)
3. Calculate P_{200} (Kaiser, 1986; Voit, 2005)
4. Calculate p_{th} using Battaglia Eq.10
5. Calculate p_e from p_{th} assuming primordial helium mass fraction of 0.24 (Vikram, 2016)

```
def epp_to_y(profile)
```

Input: Electron pressure profile (p_e)
Output: Compton-y profile
Paper: https://arxiv.org/abs/1006.1945 (Shaw, 2010)
Steps:

1. Convert from p_e to y using Shaw Eq. 4

b. 2D Compton-y submap

```
def make_proj_image_new(radius, profile,maxRadius=0.5,pixel_scale=0.005,extrapolate=False)
```

Input: radius, profile (numpy arrays)
Output: image (default to range of 0.5 Mpc)
Steps:

1. Meshgrid and calculate distance from center
2. Assign amplitude using profile

```
def Mpc_to_arcmin(r, z)
```

Input: Distance, redshift
Output: angular scale ($arcmin$)
Library: astropy

Steps:

1. Set cosmology and use conversion from astropy

c. Convolution with gaussian kernel

```
def gaussian_kernal(pix_size,beam_size_fwhp)
```

Input: pixel_size, beam_size, submap
Output: convolved submap
Paper: https://arxiv.org/abs/1106.5065 (Aniano, 2011)
Source: CMB summer school
Steps:

1. Meshgrid and calculate distance from center
2. Construct Gaussian using Aniano Eq.13

```
def convolve_map_with_gaussian_beam(pix_size, beam_size_fwhp, Map)
```

Input: pixel_size, beam_size, submap
Output: convolved submap
Library: scipy.signal
Steps:

1. Make gaussian beam
2. Convolution using scipy (FT & IFT)

d. Conversion between Compton-y and ΔT submap

```
def f_sz(f, T_CMB)
```

Input: Electron pressure profile (p_e)
Output: Compton-y profile
Paper: https://arxiv.org/abs/2101.08373 (Vavagiakis, 2021)
Steps:

1. Calculate radiation frequency using Vavagiakis Eq.2
2. Calculate temperature decrement using Vavagiakis Eq.1

e. Image output

```
def generate_img(radius, profile, f, noise_level, beam_size, z, option, s = False, p = None):
```

Options:

1. 1D Compton-y Profile
2. 2D Compton-y Submap (unconvolved)
3. Gaussian beam
4. 2D Compton-y Submap (convolved)
5. 2D ΔT submap
6. Gaussian Noise
7. 2D ΔT submap with noise
8. 2D Compton-y Submap with noise
9. 2D Compton-y Submap with noise, ring and disk aperture

Note: If AP = True: generate_img returns tSZ signal (disk-ring)

f. Calculation of tSZ signal

```
def tSZ_signal(z, Map):
```

Input: redshift, submap

Output: tSZ signal ($arcmin$)

Library: astropy

Paper: https://arxiv.org/abs/1607.02139 (Calafut et. al., 2021)

https://arxiv.org/abs/2101.08373 (Vavagiakis, 2021)

Steps:

1. Calculate average distance diameter for given z and for z = 0.5
2. Calculate disk and ring aperture using Calafut Eq. from page 12
3. Meshgrid and calculate distance from center
4. Calculate tSZ signal using Vavagiakis

The simulation contains the following 3 files:

1. generator.ipynb: imports function from the script and data from the yaml file, stores the images in a subdirectory
2. simulation.py: critical functions used to construct the profile and output image
3. config.yaml: specifies the cosmology, telescope

Main inputs: Mass (unit of $10^{13}M_{solar}$), Redshift

Other parameters:

a. Cosmology (options of Battaglia, Planck 2015, Planck 2018)
b. Telescope (options of ACT_DR4, ACT_DR5, SPT)
c. Observation frequency (unit of GHz)

Output:

a. Final submap: ΔT with noise (option 7)
b. Other intermediate images (provide sanity check)

Output size: 18 $arcmin$ in diameter (also labeled with Mpc)

Pixel size: 0.5 $arcmin$ (default to Atacama Cosmology Telescope)

Comparison with real data:

ACT 2008:
Data Release: https://lambda.gsfc.nasa.gov/product/act/act_sz_clusters_2008/
Paper: Marriage et al. (2010)

ACT ID	$M_{500c}\ (10^{14} m_{solar})$	Redshift	$\delta T(uk)$ (real)	$\delta T(uk)$ (sim)
ACT-CL J0102-4915	9.9 ± 3.5	0.75	-520 ± 58	-601
ACT-CL J0235-5121	6.2 ± 1.4	0.43	-320 ± 51	-355
ACT-CL J0245-5302	6.6 ± 1.0	0.30	-198 ± 22	-412
ACT-CL J0330-5227	9.7 ± 1.7	0.71	-155 ± 26	-579
ACT-CL J0438-5419	12.1 ± 2.2	0.54	-214 ± 27	-695
ACT-CL J0658-5557	14.0 ± 1.2	0.296	-483 ± 42	-889

Comparison between varying apertures for redshift:

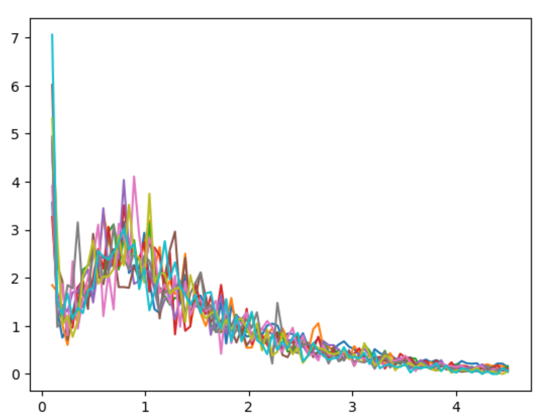
Equation 1:
$\Theta(z) = 2.1' D_A(z)/D_A(z=0.5)$ (in the paper)

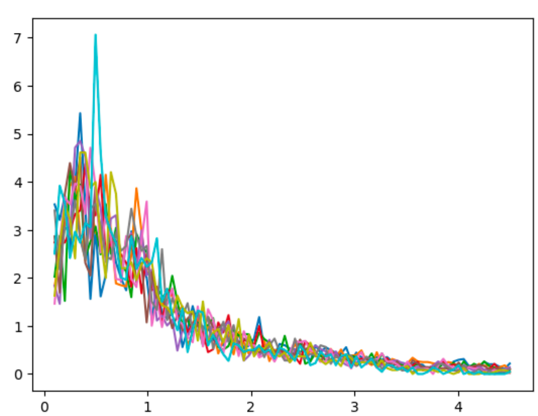
Equation 2:
$\Theta(z) = 2.1' D_A(z=0.5)/D_A(z)$

The following is based on SPT 150GHz, $5 < M_{vir} < 50$ ($10^{13} M_{solar}$), $0.1 < z < 4.5$:
For each mass and redshift, the $\Delta_{retrieve}$ is averaged over 10 instance of random noise

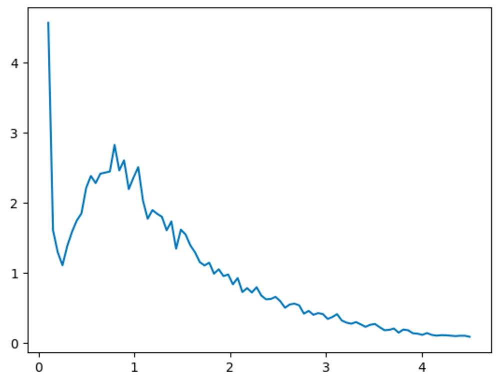
M_{vir} / $10^{13} M_{solar}$	Equation 1	Equation 2
5	54.86	36.28
10	24.33	15.70
15	15.07	10.32
20	10.68	6.61
25	8.00	5.75
30	6.58	4.19
35	4.99	3.48
40	4.60	2.84
45	3.89	2.45
50	3.40	2.19

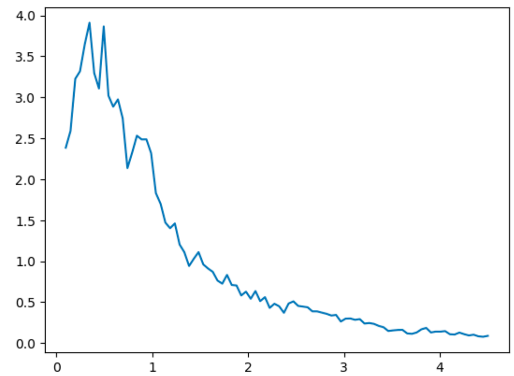
Visualization:
Each line scaled using the mean over all redshift for a specific mass



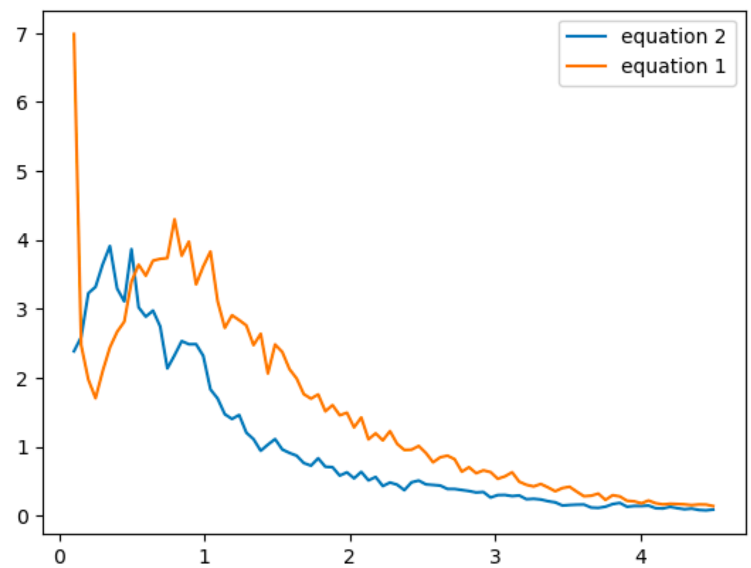


Average over all lines (overall trend of $\Delta_{retrieve}$ over redshift)





Both using scaling for Eq.2



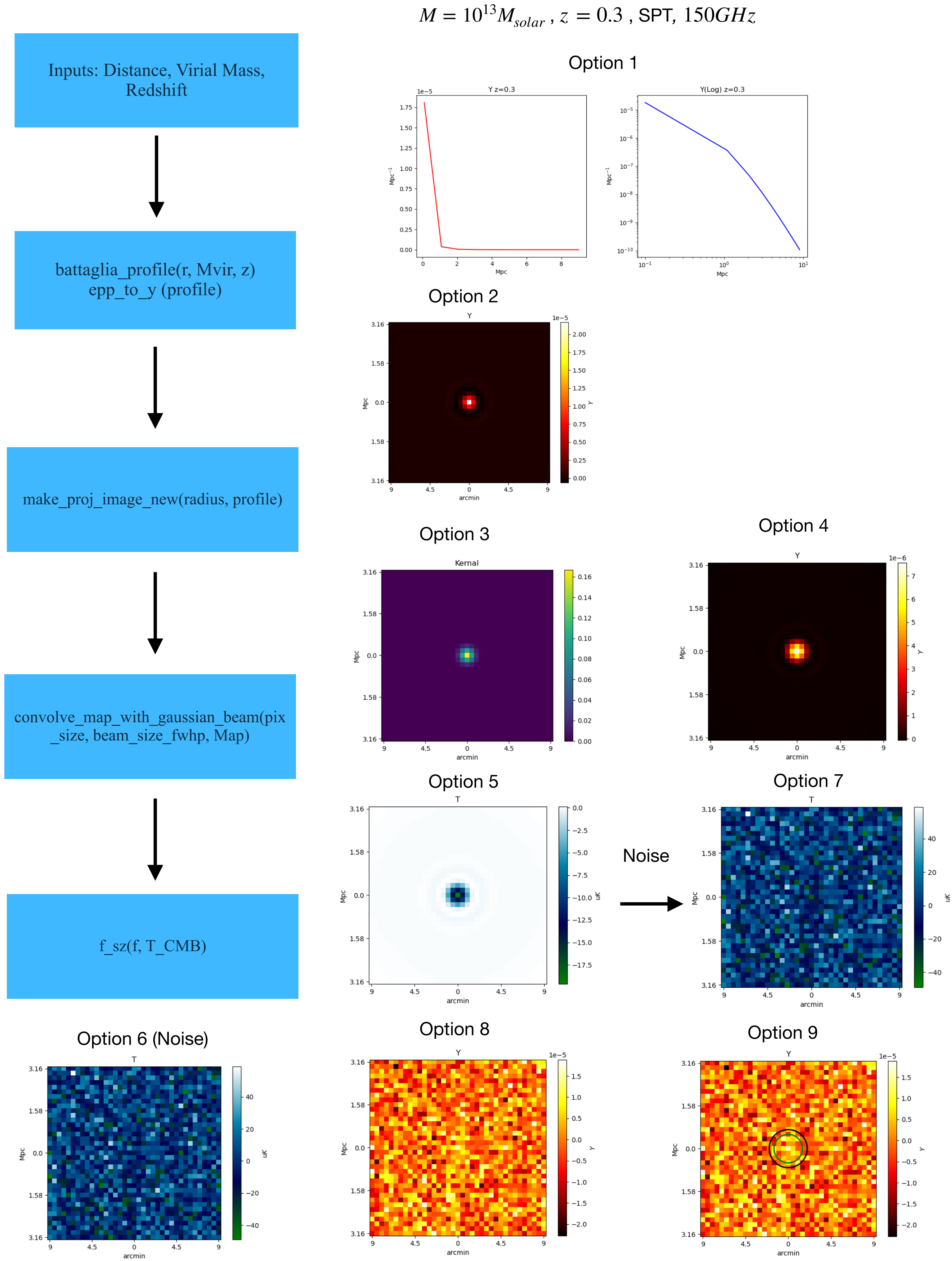
Points of intersection: z = ~0.149, ~0.52

Conclusion:
Eq.1 is preferred for range $0.149 < z < 0.52$
Eq.2 is preferred for range $z < 0.149$, $z > 0.52$

2.1' is an empirical value. Relationship between aperture size and redshift should have an general form of

$$\Theta(z) = mx + b \text{ ,where } x = D_A(z = 0.5)/D_A(z) \,|\, D_A(z)/D_A(z = 0.5)$$

Experiments should be done to determine the value of m and b empirically.


$M = 10^{13} M_{solar}$, $z = 0.3$, SPT, 150GHz
Inputs: Distance, Virial Mass, Redshift
battaglia_profile(r, Mvir, z)
epp_to_y (profile)
make_proj_image_new(radius, profile)
convolve_map_with_gaussian_beam(pix_size, beam_size_fwhp, Map)
f_sz(f, T_CMB)
Option 1
Y z=0.3
Y(Log) z=0.3
Mpc
Mpc⁻¹
Option 2
Y
Option 3
Kernal
Option 4
Y
Option 5
T
Noise
Option 7
T
Option 6 (Noise)
T
Option 8
Y
Option 9
Y
arcmin
Mpc
uK