

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

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

Input: pixel_size, beam_size, submap

Output: convolved submap

Source: [CMB summer school](#)

Steps:

1. Make gaussian beam
2. Fourier transform the submap and beam
3. Convolution & IFT

c. 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

d. 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 (Range 5 Mpc)
3. 2D Compton-y Submap (Range 0.5 Mpc) (unconvolved)
4. Gaussian beam
5. 2D Compton-y Submap (Range 0.5 Mpc) (convolved)
6. 2D ΔT submap
7. Gaussian Noise
8. 2D ΔT submap with noise
9. 2D Compton-y Submap with noise

Inputs: Distance, Virial Mass,
Redshift



battaglia_profile(r, Mvir, z)
epp_to_y (profile)



make_proj_image_new(radius, profile)

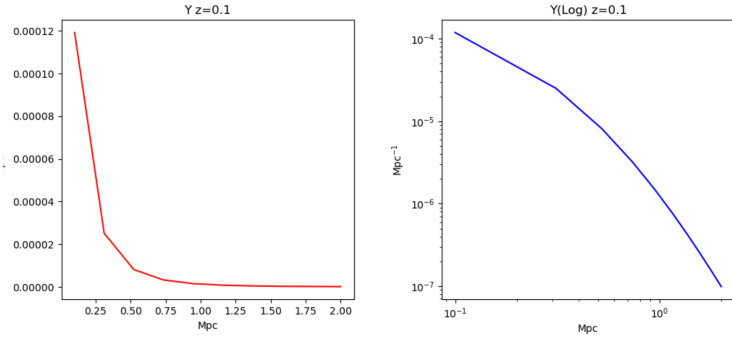


convolve_map_with_gaussian_beam(pix
_size, beam_size_fwhp, Map)

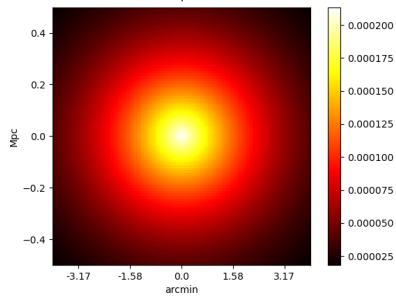


f_sz(f, T_CMB)

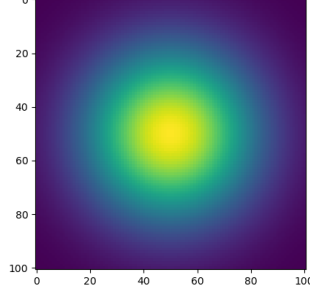
Option 1



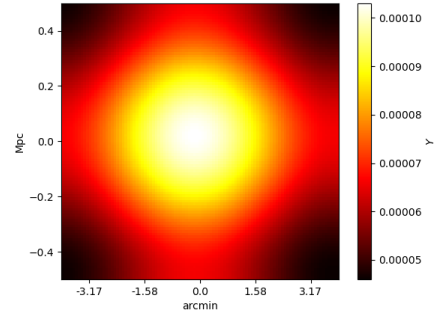
Option 3



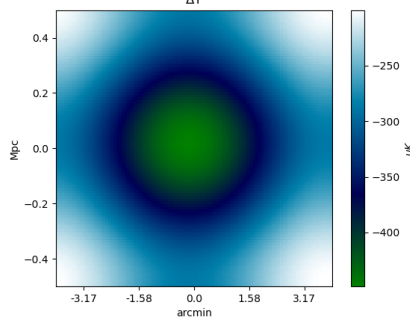
Option 4



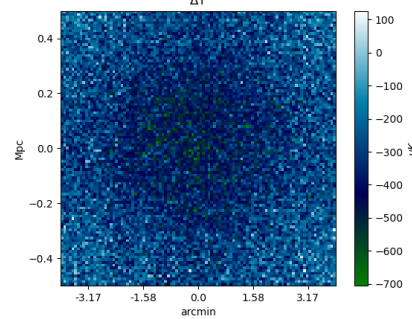
Option 5



Option 6



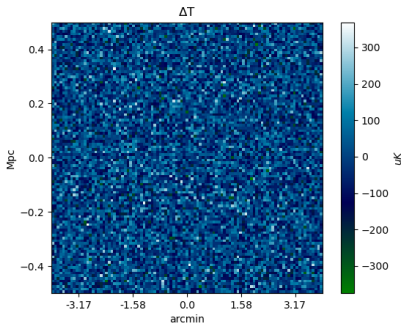
Option 8



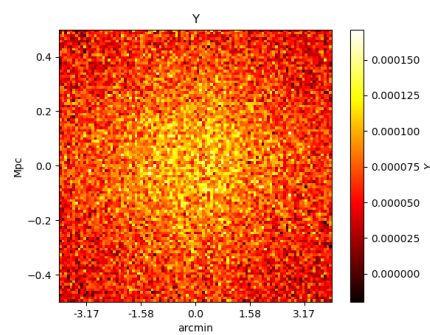
Noise



Option 8 (Noise)



Option 9



Final Submap