Chapter 4

Temperature profile modeling

Now we start using the clusteranalysis-software for the spectral analysis. To enable this, copy first a file **/data/jnevalai/clusteranalysis/repository/kurssi/idl_spectral_setup.pro** into directory **/wrk/***yourname***/clusteranalysis/data/OBJNAME/INSTRUMENT/OBSID**. Edit the keywords to match your data.

!!! DO ".run idl_spectral_setup" EACH TIME YOU START A NEW IDL SESSION!!!

4.1 Data

Extract spectra in several concentric annuli around the cluster center, and apply the spectral modeling. The best-fit value and the statistical uncertainties of the temperature at 1σ level in the different region constitute the data for the radial temperature profile.

Write the temperature profile results into a file called "t.data". In this file, the two first lines constitute the header and should not be changed (copy an example file **/data/jnevalai/clusteranalysis/repository/spectral/t_example.data** into your directory). After the header, each line corresponds to a different region of the cluster. Each line has 5 elements. 1. and 2. element are the inner and outer radius of the region in arcmin. 3., 4. and 5. element are the best-fit temperature, the lower and the upper 1 σ statistical limit.

4.2 Models

Here we model the above derived temperature profile. In the current software installation, there are 5 possibilities for the functional form of the 3D temperature profile, namely CONST, LINEAR, BETA , CONSTPOW and COOLCONSTPOW:

$$CONST \qquad T(r) = T_0 \tag{4.1}$$

$$LINEAR \quad T(r) = T_0 \times \left[1.0 + Tslope \times \frac{r}{rs}\right],\ where\ rs \equiv 60' \tag{4.2}$$

$$BETA \qquad T(r) = T_0 \times \left[1 + \left(\frac{r}{r_s}\right)^2\right]^{-c} \tag{4.3}$$

$$CONSTPOW \qquad T(r) = T_0 \times \left[1 + \left(\frac{r}{r_s}\right)^{-c}\right] \tag{4.4}$$

$$COOLCONSTPOW \quad T(r) = \left[T_{min} + \left(dT \times \left(\frac{r}{r_c}\right)^{0.4}\right)\right] \times \left[1 + \left(\frac{r}{r_s}\right)^{-c}\right] \tag{4.5}$$

$$\tag{4.6}$$

In CONST model, the temperature is constant at all radii. In LINEAR model, the temperature decreases linearly with radius. Tslope determines, how fast the temperature decreases with radius. The scaling is set so that at r=0, the temperature is T_0, and at 60' the temperature is 0 for Tslope = -1.

In BETA and CONSTPOW models the slope of the profile outside the scale radius r_s approaches a power-law with index 2c and c, respectively. In the central $r << r_s$ regions the temperature is quite constant. In the BETA model the radial decrease of the temperature in the inner regions is faster. The central cooling in most relaxed cluster can be taken into account by choosing the COOLCONSTPOW model. This is otherwise like the CONSTPOW model, except that the temperature increases from T_{min} at r=0 to $dT + T_{min}$ at $r = r_c$. The radial dependence of the temperature within r_s is fixed to $T \propto r^{0.4}$.

4.3 Fit

The basic fitting algorithm is the same as used for the surface brightness profile analysis. The temperature profile fit is complicated because the 3D model must be projected in the line-of-sight in order to compare the model prediction with the data. Using the geometry of the observation (basically the sizes of the spectrum extraction annuli) the software computes analytically the volumes of thin spherical shells cut by the extraction annuli in each line-of-sight. Using the gas density profile determined earlier the software attributes each shell a gas density value. Then, the given trial temperature profile model is used to attribute each shell a temperature value. Then, for a given temperature profile model the software weights each of the shell temperature with the emission measure or gas mass.

Edit first the parameters in a file **fit_tprof_par.pro** You can control the choice of the model by setting the parameter **tprofmod** into **const, linear, beta, constpow,** or **coolconstpow**, corresponding to the models described in Eqs. 4.1–4.5 above. You can choose the weighting method by setting the parameter **weighmeth** to **EM**. The resulting weighted temperatures in the extraction annuli are then compared with the data, and the best fit is found by minimizing the χ^2.

To run the fit, execute a script command **@tproffit**. This command reads the necessary parameters from **fit_tprof_par.pro**, compiles necessary procedures and executes the program **fit_tprof.pro**. The projected best-

fit model is plotted together with the data in the file defined as **T_plotfile** in **fit_tprof_par.pro**. The results are saved in a file defined as **Tfitfile** in **fit_tprof_par.pro**.

4.4 Uncertainties

In addition to the best-fit parameter values, we need to estimate the statistical uncertainties of the model parameters. The idea is to generate a large number of temperature profile "data" sets where the temperature value in each radius is given by T_{best} + G $\times\sigma_T$ where T_{best} and σ_T are the best-fit temperature value and its statistical uncertainty in a given radius. G is a random variable which follows normal Gaussian distribution (average value = 0.0, standard deviation = $\pm$1.0). You then repeat the temperature profile model fit to such generated data sets. As a consequence, you obtain a distribution of temperature profile parameter values which you can use to determine the 1 σ interval (i.e. remove 16% of the smallest and biggest values and use the remaining range as the statistical uncertainty for a given parameter.

The uncertainty estimation can be done in practice by running a script command **@tproferr**. For this, edit the parameters in file **err_tprof_par.pro**. The parameters controlling the fit (**niterhot_Terr, niter0_Terr, nsimu**) most likely work with the default setting. Change the parameters **Tfitfile, Terrfile, T_errplotfile** to correspond to the desired file names of the input and output data and the plot. The plot **T_errplotfile** shows the best-fit model to each of the generated "data" set. This can be used to verify that the fits make sense.

Chapter 5

Hydrostatic mass profiles

5.1 Theory

We will use the profiles of the gas density and the temperature (T(r) and ρ_{gas}(r) derived before to derive the integrated total mass profile assuming hydrostatic equilibrium (see lecture notes). The hydrostatic equilibrium equation can be written as

$$M_{tot}(\leq r) = 3.70 \times 10^{13} M_{\odot} \frac{T(r)}{keV} \frac{r}{Mpc} \left(-\frac{d \ln \rho_{gas}(r)}{d \ln r} - \frac{d \ln T(r)}{d \ln r} \right) \tag{5.1}$$

The density profile of the total mass is then obtained by

$$\rho_{tot}(r) = \frac{M_{tot}(r)}{V(r)} = \frac{M_{tot}(r+dr) - M_{tot}(r)}{\frac{4}{3}\pi \left[(r+dr)^3 - r^3\right]}, \tag{5.2}$$

where dr is the radius element. The dark matter mass and density profiles are then obtained by

$$M_{dark}(r) = M_{tot}(r) - M_{gas}(r) \tag{5.3}$$

and

$$\rho_{dark}(r) = \rho_{tot}(r) - \rho_{gas}(r), \tag{5.4}$$

and the gas mass fraction by

$$f_{gas}(\leq r) = \frac{M_{gas}(\leq r)}{M_{tot}(\leq r)}. \tag{5.5}$$

The overdensity δ at a given radius is defined as the average total density within that radius, divided by the critical density, i.e.,

$$\delta(r) = \frac{M_{tot}(\leq r)}{\frac{4}{3}\pi r^3} \times \rho_{crit}^{-1}, \tag{5.6}$$

where

$$\rho_{crit} = \frac{3}{8\pi} \frac{H(z, \Omega_m, \Omega_\Lambda)^2}{G} \tag{5.7}$$

In Chapter 4 we derived the distribution of temperature profile parameters to evaluate the statistical uncertainty level of the temperature profile. Here we will use these distributions to evaluate the statistical uncertainties of the masses and densities of the total and dark matter. For each set of temperature profile parameters, we use Eq. 5.1 to compute the total mass and then Eq:s 5.2, 5.3, 5.4 and 5.5 to obtain the corresponding profiles of mass, density and gas mass fraction. From the obtained distribution of mass and density profiles, we determine 1 σ interval for each quantity at each radius.

5.2 Practice

The computations are run by editing first the parameters in a file **totalmass_par.pro**. Then, in IDL prompt, execute **.run totalmass_par.pro** and **.run totalmass.pro**. The program prints mass and density values at radii defined as **rcalc_arcmin** and **rcalc_Mpc** on the screen. The resulting mass and density profiles are plotted in a file defined as **massplotfile**. The total mass, dark matter and gas are plotted with green, red and black lines, respectively. The results are saved in a file defined as **massfile**. Write down the gas mas fraction and its statistical uncertainties at the largest possible radius (or at 1 Mpc).

Chapter 6

$\mathbf{f}_{gas} \Rightarrow \Omega_M$

We will use here the gas mass fraction derived above to constrain the value of the universal matter density parameter Ω_m using the formula

$$\Omega_m = \Upsilon(r_{max}) \times \Omega_b \times f_{gas}(r_{max})^{-1}, \tag{6.1}$$

(see the lecture notes).

The computations are carried out by editing first the parameters in a file **fgas_omegam_par.pro**. You should insert the above derived gas mass fraction value at r_{max} and its uncertainty into keywords **fgas** and **sig_fgas** and the value of the Hubble constant you used for evaluating the masses into **H0_fgas** (defined as H in **gasmass_par.pro** above). You should choose a value for Ω_b and its uncertainty. One possibility are the WMAP 3 yr results, i.e. setting **Omega_b**, **sig_Omega_b** and **H0_Omega_b** into 0.0223, 0.0008 and 100., respectively. The local baryon enhancement parameter **Upsilon** at large radii is ∼0.9, based on cosmological structure formation simulations. Choose H_0 where you want to evaluate Ω_m. A good choice is to use the WMAP 3yr results, i.e. to set **H_best**, **H_best_lo**, **H_best_up** into 74., 71., 77. , respectively.

In IDL prompt, execute **.run fgas_omegam_par.pro** and **.run fgas_omegam.pro**. This will produce a Ω_m - H_0 plot and print the Ω_m value and its uncertainties for the above choices on the screen.