**Notes on Code**

Arnaud:

Task: To calculate the “Arnaud profile” of a cluster of given mass and redshift

Strategy: We’ll go through each section of the code commenting on what it is exactly doing and at the end we will provide a summary of inputs and outputs.

Part One: Here the function EFACT is defined. It is parameterized by the redshift, z, and the matter density of the universe, omm (we assume here that omm refers to OmegaM defined as 0.27. However, this might not be correct.

By way of explanation for myself: the redshift, z, of a cluster is the shift in the spectrum of the emitted light from the cluster towards the red side of the spectrum. This is caused by the fact that the universe is expanding (specifically that the cluster is moving away from us). We can determine the speed of this motion away from us by determining how great the red shift has been. For distant clusters one can use z to determine the distance to that cluster by plugging it into Hubble’s law which relates the velocity of a cluster to its distance from us. (v = H0\*d)

Also by way of explanation to myself: the matter density, omega in this code is the mass density of ordinary mass (baryonic mass) plus dark matter. So to be clear the total density parameter, lets call it omegaT, is equal to the matter density used in this code plus the effective mass density of relativistic particles plus the effective mass density of dark energy. OmegaT is the ratio of the actual measured density of the universe to the critical density of the universe. The critical density of the universe is the density necessary to cause the universe to expand forever. It has been found that OmegaT is actually very close to 1 (around 1.02 +/- 0.02). This means that the actual density of the universe is close to a value that would mean that it (the universe) would expand indefinitely. Still omega (just the matter density used in this code) is only .27 which suggest that most of the density of the universe is contained in the mass density of dark energy, since the mass density of relativistic particles is fairly low.

So looks like the function EFACT performs the following. Where the matter density (~0.27) and z is is of course the redshift:

So not at the moment sure what this means. In Part 2 it seems that EFACT modifies eh rho critical in some way. But it is not at all clear in what way. The second term seems to just be the reminder of the density of the universe (relativistic and dark energy). Also seems like EFACT is only really a function of z since it is the only parameter changing. But I am a little unclear on what this means at the moment, perhaps it is an effective redshift?

Part Two: Here another function, RHO\_CRIT is defined as a function of the redshift, z, the Hubble constant H0 and the matter density, omm.

Again by way of explanation to myself: the Hubble Constant H0 is a unit of measurement that quantities the expansion of the universe. Simply put, the H0 is the slope of the velocity of the object (say a cluster) as a function of its distance from the observer. So it is of great importance to have a very accurate measure of this rate or slope, since it parameterizes the known universe.

So the function RHO\_CRIT performs the following operation. Where G is the universal gravitation constant in units of km^2 Mpc/s ^2:

So as an output we get the critical value of rho in units of Msun/Mpc^3. For clarity rho critical is the critical density discussed above that would mean that the universe will expand indefinitely. This equation in streamlined form is simply:

So now it seems that EFACT modifies the rho critical in some way, since it is the factor by which equations (2.) and (3.) differ but again I’m a little at a lost for how. Additionally, it seems important to mention that (2) expresses rho\_critical as a function of z, the redshift. This is because H0 and Ω0 are both defined. Of course this is assuming that by ‘omm’ the code is referring to Ω0, the matter density

Part 3: Here a function PROJ\_ARNAUD\_PROFILE is defined as function of x, which at the moment I am assuming is the radial distance outward from the center of the cluster. Also a common block is created. For my own notes again, a common block definition statement creates a common block with the designated name and places the variables that follow into that block. The block along with its variables can be later reference by any program unit that declares that common block. So the variables defined in this common block are *y* and *params*. The common block itself is named *cblock,* The variable y is again unclear to me as far a meaning, but most likely will fit in somewhere with defining some distance. The variable params is an array defined as [alpha, beta, gamma, c500, R500]. There are 3 other additional variables that are defined outside of the block: index = (beta-gamma)/alpha, x1 = x \* (c500/R500), and y1= y \* (c500/R500).

The function produces the following:

So this equation is quite close to that found in Arnaud et al, 2010, namely the generalized NFW (GNFW) model equation. So it seems reasonable to assume that *proj* is an equation which gives the pressure profile of a given cluster as a function of radial distance from the center.

However, I note here some differences from the equation printed in Arnaud et al, 2010 (which I have reprinted below for clarity):

So the two differences are listed:

1. Seems that P0 has been set to 1 in the code.
2. Seems that c500x seems to be set equal to x12 + y12 = [r]2. Which when raised to γ/2 yields simply c500x = r. Moreover, seems that is just a scaled radius. A similar simplification is made in the second instance of c500x raised to α.

It is useful to state the meaning of some of the variables here that occur in both the formulation used in the code and the Arnaud paper:

* + - * α = the intermediate slope r ~ rs
      * β = the outer slope r >> rs
      * γ = the central slope r << rs
      * rs =R500/c500 = may be some sort of characteristic inner radius, but unclear
      * c500 = a concentration/density measurement at some characteristic radius
      * R500 = a characteristic radius for the cluster
      * X1= the x value of rs
      * Y1= the y value of rs

So the code as written returns a value, *proj*, which is a (scaled? Since P0 has been set to 1?) pressure given a certain input, namely an x, y and params[0-4]. Presumably, it does this for some large set of x & y using a constant set of values for params[0-4]. Or perhaps put a different way, a set of x & y over which the constant values of params[0-4] are valid.

To give more details again for my own sake: GNFW stands for the generalized Nagai-Frenk-White model. This model gives the average scaled pressure profile of a cluster as a function of radial distance from the center of the cluster. By scaled pressure, we mean that is really equal to P(r) / P500. More work needs

Why is knowing the average scaled pressure profile of a cluster all that meaningful? Well one reason, for example, is that we now use the SZE to determine cluster mass. However the SZE is a measure of the integrated pressure of the ICM. Thus, accurate pressure profiles of the ICM may come in handy as we go about doing these measurements. Moreover, the dynamics of the ICM are quite interesting in their own right, thus understanding them is quite important.

Part 4: The final part of the Arnaud code is a IDL procedure, given the name ARNAUD and parameterized by m500, redshift, r\_arcmin, dT\_uK. There is a last command: stopit=stopit which I am unclear about the meaning of. Next the previously defined common block, *cblock*, is invoked. Before going on let’s define the parameters of ARNAUD:

* + - * M500 = M500 🡪 characteristic mass of the cluster
      * Redshift 🡪 defined in Part 1. Usually given by variable *z*
      * R\_arcmin 🡪 ?????
      * dT\_uK 🡪 a temperature declination in Kelvin

So next some defaults are defined, presumably just in case there are some oddities in the data sets:

* if a value in the data set (array) of “n\_elements(m500)” is equal to 0 then take m500 to be equal to 6e14 in units of Msun
* if a value in the data set (array) of “n\_elements(redshift)” is equal to 0 then take redshift to be equal to 0.4
* if a value in the data set (array) of “n\_elements(stopit) is equal to 0 then take stopit to be equal to 0

Defaults are defined:

* cutoff = 6 (units of characteristic radius R500) 🡪 not really sure what this one means really
* H0 = 72 🡪 Hubble constant in km/s/Mpc
* OmegaM = 0.27 🡪 density of ordinary baryonic matter ‘
* hubble = H0/100 🡪 reduced Hubble constant 🡪 interesting note: physical baryon density is equal to baryon density multiple by square of reduced Hubble constant
* hubble70 = H0/70 🡪 Hubble constant parameterized/normalized
* mmw = 0.59 🡪 overall mean molecular weight of the gas
* mu\_e = 1.143 🡪 mean molecular weight of electrons in the cluster
* m\_p = 1.6726w-27 🡪 Mass of proton in kg
* mpc = 3.08568025e22 🡪 Mega parsec in meters
* Msol = 1.98892e30 🡪 Solar masses in kg
* sigma\_T = 6.652e-25/ (1002) 🡪 Thompson cross section in m2
* m\_e = 9.11e-31 🡪 mass of electron in kg
* k\_b = 1.3806503e-23 – Boltzmann constant in m2kg/s2K
* c = 3e8 🡪 speed of light in m/s
* q= 1.60217646e-19 🡪 electron volt in units of joules
* me\_csq = m\_e\*c2 / (1000\*q) 🡪 Energy of electron using its rest mass and in kilo electron volts
* R500 = 🡪 ????
* press\_to\_sz = sigma\_T/me\_csq 🡪 This is the constant term to the Compton parameter *y*. Does not include the temp.

Arnaud parameters set:

* P0 = P0 = 8.403/(hubble70)3/2 🡪 The pressure
* c500 =1.177 🡪 characteristic density/concentration
* alpha = 1.0510 🡪 the intermediate slope r ~ rs
* beta = 5.4905 🡪 the outer slope r >> rs
* gamma = 0.3081 🡪 the central slope r << rs
* params = [alpha, beta, gamma, c500, R500] 🡪 defined in Part 3
* E= Efact(redshift, OmegaM) 🡪 ??????
* Alphap = 0.12 🡪 a correction that accounts for M500 errors created by deriving them from the M500 to YX relation
* Pnorm = 🡪 ?????
  + Pnorm = (Pnorm)(P0)(1e6)
* x 🡪 is a double precision array of 100\*6\* R500 elements running from 0 to 6\* R500 -0.01 in increments of 0.01
* y2 🡪 is a double precision array with dimensions n\_elements(x) by 1. Each value is initially set to 0
* y2b 🡪 is a double precision array with dimensions n\_elements(x) by 1. Each value is initially set to 0

Now we come to what the code is actually doing with the data. A “for” loop is begun. We try to go step by step:

1. So we have an array x of 100\*6\* R500 elements running in values from 0/100 to (100\*6\* R500 -1)/100.
2. For i ranging from 0 to n\_elements(x)-1 we are defining R as x[i]. So, for example, if i = 54 then R = x[54], which is the 55th element in the array x.
3. We then set y equal to this R value
4. If the R value is less than cutoff\*R500 (that is 6\*R500) then the code performs the following integration using the Romberg method:
   1. The function it uses is the one created in part 3, the proj\_arnaud\_profile.
   2. The limits of integration are 0.001 to the upperlim defined as sqrt((cutoff\*R500)^2 – R\*R).
   3. Additional features are set to specify the operation
      1. /double: specifies double precision arithmetic
      2. EPS = 1de-6: specifies the fractional accuracy
      3. JMAX = 20: specifies the number of allowed steps in the integration
      4. k=5: specifies use of the Rombus method of 2k.
   4. The value of this integration for each R = x[i] value will populate the empty array y2[i]
5. If the R value is more than the cutoff\*R500 then the element of the array y2 at that value i,is zero.
6. Thus an array y2 is generated, which is populated by these integrations.

So all in all the average scaled pressure profile created in Part 3 is being integrated from just about 0 (0.001) up to a value which runs from 1/100 up to n/100.

Part 5: After the for loop there is some other stuff. Mostly definitions. Also a new function, ang\_diam\_dist, appears although it has not be previously defined in the code thus far. It’s a function of OmegaM, H0/100 and redshift. Here are the other values that are defined, some are based on the for loop in Part 4:

Running list of Questions:

1. Part 1
2. Part 2
3. Part 3
   1. Why is PROJ\_ARNAUD\_PROFILE a function of x? Or put another way, what exactly is meant by x1 and y1 and how are they related to x and y? Are they a characteristic concentration at a given scaled radial distance from the center of the cluster?
   2. Still a little hazy on what is meant by c500 & the ratio c500/R500
      1. And rs
4. Part 4
   1. We have at one point n\_ elements (m500), n\_elements(redshift), n\_ elements (stopit) and then a for loop using n\_elements(x). Does x run over m500, redshift, & stopit?
   2. Meaning of constants:
      1. Likewise, what does the Thompson cross section mean physically?
      2. me\_csq = me\*c2/1000\*q 🡪 looks like an energy but what is meant be the ‘csq’?
      3. R500, I’m pretty sure I know what R500 means, however I want to understand the way it is defined in the code, specifically as:
         1. It is the radius at which the average cluster density is 500 times the critical density.
      4. E is defined as Efact, how is it different from EFACT of Part 1?
      5. Pnorm, both the initial definition and then the subsequent one:
         1. Pnorm = 1.65e-3\*E^(8./3.)\*(hubble70\*M500/3.0e14)^(0.6667 + alphap)\*(hubble70^2); % units are keV/cm^3
         2. Pnorm = Pnorm\*P0\*1.0e6; % keV/m^3
5. Part 5
   1. Seems like a lot of variables are not being double/redefined. Is this now a separate matter? Or are the previous definitions being overridden? Examples given:
      1. y2 is now defined as “press\_to\_sz\*y2\*Pnorm\*2.0\*mpc”
         1. Unclear what “mpc” stands for
      2. absy\_150ghz = y2
   2. Also there is a new function ang\_diam\_dist (in bold), what might this be referring to?
      1. r\_arcmin = r\_over\_r500/**ang\_diam\_dist(OmegaM, 1d - OmegaM, H0/100., redshift)**\*180./!dpi\*60.
6. Others
   1. What is meant by a “self-similar” model?
   2. What does % mean in IDL?

Construct Simulation Map:

Task: To produce a simulated map of the ICM pressure density (in units of Kelvin

Strategy: We’ll go through each section of the code commenting on what it is exactly doing and at the end we will provide a summary of inputs and outputs.