Instructions Numerics Lab - An Example of Monte Carlo Markov Chain Methods: Estimating the Mass of a Galaxy Cluster with Weak Lensing

Stella Seitz and Jochen Weller
edited by Steffen Hagstotza and Kerstin Paech

University Observatory
Ludwig-Maximilians-University Munich

January 6, 2020

1 The Weak Lensing Signal of a Galaxy Cluster

The propagation of light is affected by the gravitational field it passes through along its way from the observer. This effect is called gravitational lensing. This leads to the distortion of the image of an object compared to its true intrinsic shape. For small gravitational fields this effect is called weak lensing. In the following we present, the quantities required from the *thin lens approximation*. If the lens is thin compared to the total length of the light path, the lens mass distribution can be projected along the line-of-sight and replaced by an orthogonal mass plane, called lens plane. Analogously, the source is assumed to lie in the so-called source plane. We first define the projected (or surface) mass density

$$\Sigma(\vec{\xi}) = \int \, dz \rho(\vec{\xi}, z) \, ,$$

where ρ is the density, $\vec{\xi}$ a vector in the plane of the sky and z refers to the line of sight. The deflection $\vec{\hat{\alpha}}$ is then given by

$$\vec{\hat{\alpha}} = \frac{4G}{c^2} \int \frac{(\vec{\xi} - \vec{\xi'})\Sigma(\vec{\xi'})}{|\vec{\xi} - \vec{\xi'}|^2} \, d^2\xi'$$

The geometry of a typical gravitational lens system is shown in fig. 1. A light ray is emitted from a source at transverse distance $\vec{\eta}$ from the optical

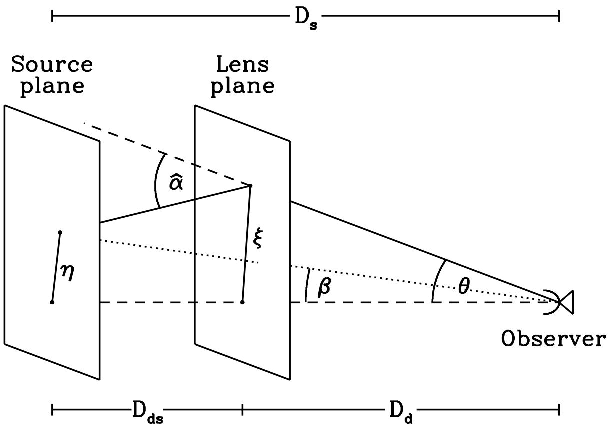

Figure 1: The geometry of the gravitational lensing configuration.

axis to the observer, crossing the lens plane at transverse distance $\vec{\xi}$, deflected by an angle $\vec{\hat{\alpha}}$. The angle between the optical axis and the image is $\vec{\theta}$, while the one between the optical axis and the true source is $\vec{\beta}$. The distances observer-lens, lens-source, observer-source are respectively D_d, D_{ds}, and D_s. From this geometry, one can obtain a relation between the unlensed and lensed position of the object

$$\vec{\beta} = \vec{\theta} - \frac{D_{ds}}{D_s}\vec{\hat{\alpha}}(D_d\vec{\theta}) \,,$$

which is usually called the lens equation. The solutions $\vec{\theta}$ of this equation yield the angular positions of the images of source at $\vec{\beta}$. It is convenient to define the convergence by

$$\kappa(\vec{\theta}) = \frac{\Sigma(D_d\vec{\theta})}{\Sigma_{cr}} \,,$$

with Σ_{cr} the critical surface density

$$\Sigma_{cr} = \frac{c^2 D_s}{4\pi G D_{ds} D_d} \,.$$

The critical surface density is the density, which will cause the refocusing of the light beam (in the absence of shear, see below). The deflection angle is then given by

$$\vec{\alpha}(\vec{\theta}) = \frac{1}{\pi}\int d^2\theta' \frac{(\vec{\theta}-\vec{\theta}')\kappa(\vec{\theta}')}{|\vec{\theta}-\vec{\theta}'|^2}$$

In order to study how the lensed image changes from the (unlensed) source, we have to calculate the Jacobian of the imaging map, i.e.

$$A = \frac{\partial\vec{\beta}}{\partial\vec{\theta}}.$$

This matrix can be decomposed in a diagonal part and a trace-free part

$$A = (1-\kappa)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \gamma\begin{pmatrix} \cos 2\phi & \sin 2\phi \\ \sin 2\phi & -\cos 2\phi \end{pmatrix},$$

with κ the convergence describing the magnification of the image and γ the tangential shearing of the image around the lens. We later exploit the shearing of galaxies due to a mass distribution of a galaxy cluster.

In order to estimate the mass of galaxy cluster we have to employ a model for its density distribution. Here we use a universal radially symmetric Navarro-Frenk-White profile, given by

$$\rho_{\mathrm{NFW}}(r) = \frac{\delta_c\rho_c}{(r/r_s)(1+r/r_s)^2},$$

where $\rho_c = 3H^2(z)/8\pi G$ is the critical density of the universe that depends on the Hubble rate $H(z)$ that depends on the cosmological parameters (which we will keep fixed for this analysis). The other two free parameters of the density profile are the scale radius r_s and the density contrast

$$\delta_c = \frac{200}{3}\frac{c^3}{\ln(1+c) - c/(1+c)}.$$

c is the concentration parameter defined as $c = r_{200}/r_s$, which gives the ratio of the "virial" radius r_{200} and the scale radius r_s. With a dimensionless radial distance $x = r/r_s = \theta/\theta_s$ we can write the convergence of a NFW profile as

$$\kappa(x) = \begin{cases} \frac{2r_s\delta_{\mathrm{c}}\rho_{\mathrm{c}}}{\Sigma_{\mathrm{c}}(x^2-1)}\left(1 - \frac{2}{\sqrt{1-x}}\mathrm{artanh}\sqrt{\frac{1-\mathrm{x}}{1+\mathrm{x}}}\right) & x < 1, \\ \frac{2r_s\delta_{\mathrm{c}}\rho_{\mathrm{c}}}{3} & x = 1, \\ \frac{2r_s\delta_{\mathrm{c}}\rho_{\mathrm{c}}}{\Sigma_{\mathrm{c}}(x^2-1)}\left(1 - \frac{2}{\sqrt{1-x}}\mathrm{artan}\sqrt{\frac{\mathrm{x}-1}{\mathrm{x}+1}}\right) & x > 1, \end{cases}$$

The gravitational shear is given by

$$\gamma(x) = \begin{cases} \frac{r_s\delta_{\mathrm{c}}\rho_{\mathrm{c}}}{\Sigma_{\mathrm{c}}} g_<(x) & x < 1, \\ \frac{r_s\delta_{\mathrm{c}}\rho_{\mathrm{c}}}{\Sigma_{\mathrm{c}}} \left[\frac{10}{3} + 4\,\ln\left(\frac{1}{2}\right)\right] & x = 1, \\ \frac{r_s\delta_{\mathrm{c}}\rho_{\mathrm{c}}}{\Sigma_{\mathrm{c}}} g_>(x) & x < 1. \end{cases}$$

The functions $g < (x)$ and $g > (x)$ are given by

$$\begin{aligned} g_<(x) &= \frac{8\ \mathrm{artanh}\sqrt{(1-x)/(1+x)}}{x^2\sqrt{1-x^2}} + \frac{4}{x^2}\ \ln(\tfrac{\mathrm{x}}{2}) - \frac{2}{(\mathrm{x}^2-1)} + \frac{4\ \mathrm{artanh}\sqrt{(1-\mathrm{x})/(1+\mathrm{x})}}{(\mathrm{x}^2-1)(1-\mathrm{x}^2)^{1/2}} \\ g_>(x) &= \frac{8\ \mathrm{arctan}\sqrt{(x-1)/(1+x)}}{x^2\sqrt{x^2-1}} + \frac{4}{x^2}\ \ln(\tfrac{\mathrm{x}}{2}) - \frac{2}{(\mathrm{x}^2-1)} + \frac{4\ \mathrm{arctan}\sqrt{(\mathrm{x}-1)/(1+\mathrm{x})}}{(\mathrm{x}^2-1)^{3/2}}\ . \end{aligned}$$

The “virial” mass within r_{200} is given by

$$M_{200} \equiv M_{\mathrm{NFW}}(< r_{200}) = 200 \cdot \frac{4\pi}{3}\rho_c r_{200}^3\ .$$

We give a file called `halo5.tab`, which in column 4 contains the angular separation θ of the galaxy from the cluster center in units of arcminutes, in column 6 the shear and in column 14 the redshift z_s of the galaxy. To convert the angular separation into a radius we need to multiply the angle by the angular diameter distance d_A in the lens plane, i.e.

$$r = d_A(z_{ls})\theta\ .$$

Note that the data is given for a flat cosmological model with $\Omega_m = 0.27$. The lens is located at redshift $z_d = 0.245$ and the uncertainty of the shear measurement is $\sigma_\gamma = 0.3$ for all data points.

2 Parameter Estimation

In order to estimate the mass of the galaxy cluster statistically we fit the theoretical shear profile $\gamma(x)$ to the data. For a given cosmological model, this means fitting for the mass M_{200} and the concentration c of the profile. In the following section we will first talk about general parameter fitting procedures and we define the parameter vector $\boldsymbol{\theta} = (M_{200}, c)$.

We are interested in the posterior distribution, i.e. the probability of the parameters given the data: $p(\boldsymbol{\theta}|D)$. With Bayes' theorem this can be written as

$$p(\boldsymbol{\theta}|D) = \frac{p(\boldsymbol{\theta})p(D|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta})p(D|\boldsymbol{\theta})\, d\boldsymbol{\theta}} \tag{1}$$

In order to analyse the posterior distribution we require to calculate quantities, like moments, quantiles, highest posterior density etc., which in general is the expectation of a function of the parameters:

$$E\left[f(\boldsymbol{\theta})|D\right] = \frac{\int f(\boldsymbol{\theta})p(\boldsymbol{\theta})p(D|\boldsymbol{\theta})\,d\boldsymbol{\theta}}{\int p(\boldsymbol{\theta})p(D|\boldsymbol{\theta})\,d\boldsymbol{\theta}}$$

However this can be a tricky task, particularly for a larger number of parameters. Further notice, that for most applications it is only necessary to calculate

$$E\left[f(\boldsymbol{\theta})|D\right] \propto \int f(\boldsymbol{\theta})p(\boldsymbol{\theta})p(D|\boldsymbol{\theta})\,d\boldsymbol{\theta}$$

The majority of the numerical work has do be done by estimating the likelihood $p(D|\boldsymbol{\theta})$. This is given in the Gaussian case by

$$p(D|\boldsymbol{\theta}) \propto \exp\left[-\frac{1}{2}\chi^2(\boldsymbol{\theta})\right] \;.$$

The core of this numerics lab is now to apply this to the fitting of the weak lensing shear data of a galaxy cluster, with the parameters $\boldsymbol{\theta} = (M_{200}, c)$, where $\pi(\boldsymbol{\theta}) \propto \exp[-0.5\chi^2(\boldsymbol{\theta})]$. The χ^2 is given by

$$\chi^2(\boldsymbol{\theta}) = \sum_{i=1}^{N_{dat}} \frac{\left(\gamma(r_{dat,i}, \boldsymbol{\theta}) - \gamma_{dat,i}\right)^2}{\sigma^2_{\gamma,i}}$$

Exercise 1: Have a look at the IPython notebook. Implement the χ^2 calculation outlined above and complete the `Cluster` and `Grid_Search` classes. You can find more information and most of the functions you need in the notebook. Test your new function for speed and aim for execution times of ~ 0.1 s. If you try several methods of calculating χ^2 and find performance differences, document them briefly. Can you think of reasons for the difference in execution times? Keep in mind that Python is not a compiled language.
Once you have a fast routine, implement a loop over M and c to find an estimate for the minimal χ^2 and make a 2d-plot of the likelihood (Hint: The mass of the cluster is above $10^{14}M_\odot$ and below $10^{16}M_\odot$, the concentration is above 2 and below 10).

The general problem of posterior estimation is to calculate an integral. This can be done efficiently by a *Monte Carlo* integration. In order to simplify our notation we consider the following problem:

$$E\left[f(X)\right] = \int f(x)\pi(x)\,dx\ . \tag{2}$$

where X comprise of N continuous real variables and $\pi(x)$ is the distribution. The Monte Carlo method works by drawing samples $\{X_t, t = 1, ..., N_s\}$ from $\pi(\cdot)$ and then approximating

$$E\left[f(X)\right] \approx \frac{1}{N_s}\sum_{t=1}^{N_s} f(X_t)\ .$$

Note that if N_s is chosen large enough this is an adequate approximation. In general it is not possible to draw the X_t independently and directly from $\pi(\cdot)$, since $\pi(\cdot)$ can be any distribution. However, the X_t need not be independent, as long as they are generated in the correct proportions according to $\pi(\cdot)$.

This is given if one can create a Markov chain which has $\pi(\cdot)$ as stationary (limiting) distribution. Suppose we generate a sequence of random variables, $\{X_0, X_1, X_2, ...\}$, such that the next state is sampled from a distribution $p(X_{t+1}|X_t)$, i.e. the next state only depends on the current state of the chain and not on the entire history. This sequence is called a *Markov chain* with transition kernel (probability) $p(\cdot|\cdot)$. If the probability is well behaved (regular), the chain will gradually forget about the initial state X_0 and approach a stationary (or invariant) distribution after a sufficient sequence size. In Figure 2 we see an example of a sequence which approaches a stationary distribution. That means as t increases the sample points X_t, will look more and more like they are drawn from a stationary distribution $\phi(\cdot)$. After a sufficient long *burn-in* the points $\{X_t; t = m + 1, ..., n\}$ will be dependent samples approximately from $\phi(\cdot)$. We can now estimate $E[f(x)]$, where X has the distribution $\phi(\cdot)$

$$\bar{f} = \frac{1}{n-m}\sum_{t=m+1}^{n} f(X_t)\ . \tag{3}$$

The next step is to construct a Markov chain where $\phi(\cdot)$ is $\pi(\cdot)$. One possibility is the *Metropolis-Hastings* algorithm. At each time step a *candidate* point Y is chosen from a proposal distribution $q(\cdot|X_t)$, for example a multivariate Gaussian, with mean X_t and fixed covariance. The candidate

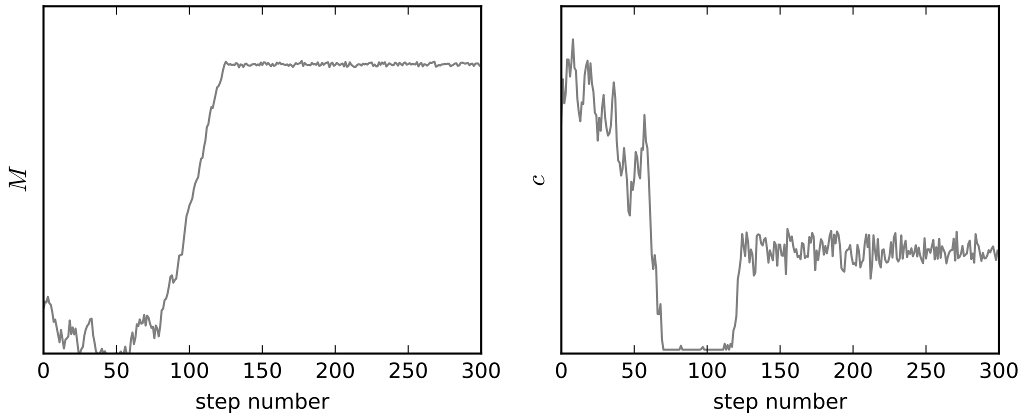

Figure 2: Markov chain sequence approaching stationary distribution.

point is then accepted with probability $\alpha(X_t, Y)$, where

$$\alpha(X, Y) = \min\left(1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)}\right) \ .$$

If the candidate point is accepted the next state becomes $X_{t+1} = Y$, if the candidate point is rejected the chain does not move, i.e. $X_{t+1} = X_t$. The Metropolis-Hastings algorithm is then:

1. Initialize to a random X_0.
2. Sample point from Y from $q(\cdot|X_t)$.
3. Sample a Uniform $(0, 1)$ variable U.
4. If $U \leq \alpha(X_t, Y)$ set $X_{t+1} = Y$, otherwise set $X_{t+1} = X_t$.
5. increment t and start again at 2.

Interestingly the proposal distribution $q(\cdot|\cdot)$ can have any form and the stationary distribution of the chain will be $\pi(\cdot)$. The Metropolis algorithm itself (which we will exploit later) considers only symmetric proposals $q(Y|X) = q(X|Y)$. For examples $q(\cdot|X)$ a multivariate Gaussian with mean X and covariance matrix C. Hence we obtain

$$\alpha(X, Y) = \min\left(1, \frac{\pi(Y)}{\pi(X)}\right) \ .$$

One has to be careful how to choose C. If C is too small there will be a high acceptance rate and slow mixing, while a wide distribution will result in low acceptance and no movement of the chain, hence resulting in slow mixing as well. We will later discuss how to improve this.

For the problem of the mass in concentration fitting we recommend as allowed regions: $5 \times 10^{13} h^{-1} M_\odot \leq M_{200} \leq 10^{16} h^{-1} M_\odot$ and $1 \leq c_{200} \leq 12$. We recommend to test of the order of a total (accepted and unaccepted) 100,000 sample points. Figure 2 shows the beginning of the chain in the M_{200} variable. In Figure 3 we show the number of samples of the Markov chain sequence, within bins of the parameters.

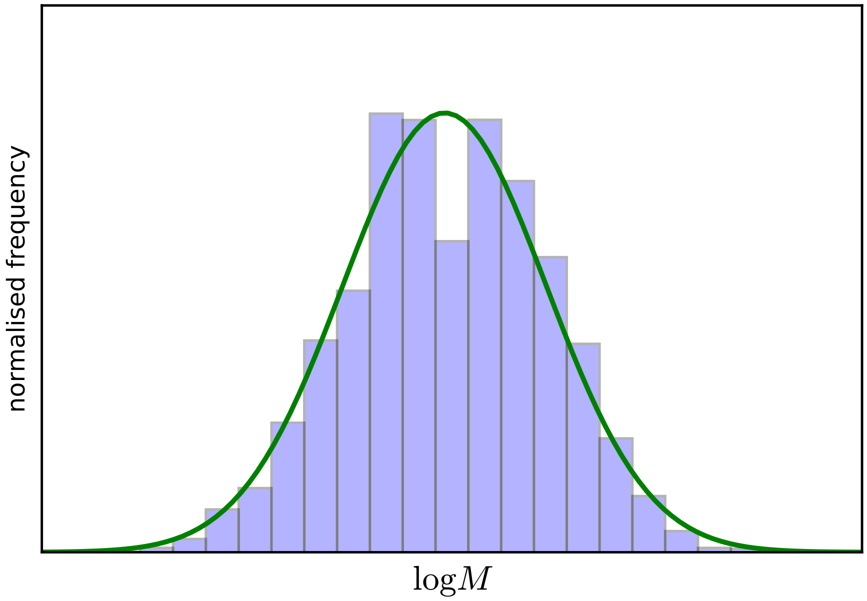

Figure 3: Histogram of the MCMC sampled distribution. The solid line is the Gaussian distribution with the variance of the chain as the width.

In Figure 4 we show the two dimensional joint likelihoods.

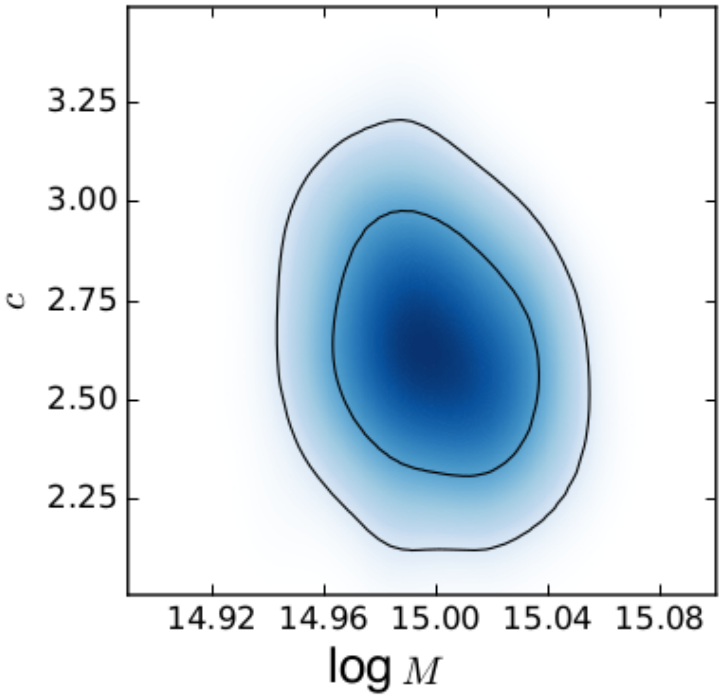

Figure 4: Joint likelihood. The color scheme correspond to the density of the sample over the parameter space, while the solid line are the 68% and 95% marginalized joint likelihoods.

Exercise 2: In the notebook you can find a `Sampler` class. Implement the Metropolis-Hastings algorithm there. Write the chain to a file as outlined in the notebook. Run some short chains and observe the convergence behaviour. What happens when you change the covariance?
Run some longer chains (depending on how fast your χ^2 calculation is) and plot a histogram of the chain values and the two-dimensional distribution.

The next question to address how we know for how long to run the chain? This is the question if the sequence has converged and is truly stationary. One of the simplest methods to address this question, is by running several chains in parallel, with *over-dispersed* starting values and compare estimates

$\bar{f}$.

The fundamental problem of inference from a Markov chain simulation is that there will always be areas of the target distribution that have not been covered by the finite chain. First we have to set up multiple chains (maybe run in parallel), with over-dispersed initial points. This is essential for a successful diagnostic. Over-dispersion can be achieved after running a single chain initially and get an idea about the distribution of this chain. One can then use the variance of this chain to achieve over-dispersion: we want to start additional chains separated from the starting point of the first chain by more than a standard deviation for each parameter to prevent convergence towards a local minimum.

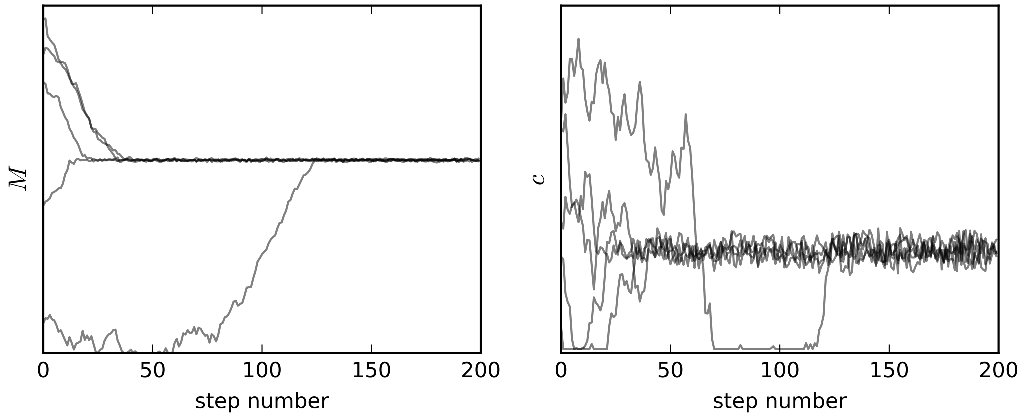

Figure 5: Start of five chains with over-dispersed initial conditions.

Figure 5 shows the results for five over-dispersed chains for our profile fitting procedure with over-dispersed initial points. It is evident from the Figure that the chains begin to converge already after 150 steps. Let us assume we are interested in a quantity ψ from the chain. These can be the parameters or any function of the parameters. Let us further assume that we run m parallel sequences of length n and label the quantities (ψ_{ij}), $j = 1, ..., n$ and $i = 1, ..., m$.

We hence compute two quantities: The between sequence variance B and the within-sequence variances W.

$$B = \frac{n}{m-1} \sum_{i=1}^{m} \left(\bar{\psi}_i - \bar{\psi}\right)^2 \ ,$$

where

$$\bar{\psi}_i = \frac{1}{n}\sum_{j=1}^{n}\psi_{ij} \qquad \bar{\psi} = \frac{1}{m}\sum_{i=1}^{m}\bar{\psi}_i \ .$$

Further we define

$$W = \frac{1}{m}\sum_{i=1}^{m} s_i^2 \ ,$$

where

$$s_i^2 = \frac{1}{n-1}\sum_{j=1}^{n}\left(\psi_{ij} - \bar{\psi}_i\right)^2 \ .$$

So W is just the *average variance* of all the chains, while B measures the *variance of the averages* of the chains. Note that the between-sequence variance B contains a factor n because it is based on the variance of the within-sequence means, $\bar{\psi}_i$, each of which is an average of n values ψ_{ij}.

One estimate of the variance of ψ in the target distribution is

$$\widehat{\mathrm{var}}(\psi) = \frac{n-1}{n}W + \frac{1}{n}B \ ,$$

which is an overestimate. Further W is an underestimate of the target variance, because individual chains had not have time to cover the target distribution. For $n \to \infty$ both estimates approach the target variance $\mathrm{var}(\psi)$. Convergence can now be established by monitoring

$$\sqrt{\hat{R}} = \sqrt{\frac{\widehat{\mathrm{var}}(\psi)}{W}} \ , \tag{4}$$

which approaches 1 at convergence. You should run at least 3 different chains. Note that there are many other convergence criteria some of which are discussed in Gilks et al. (1996).

> Exercise 3: Generate at least 5 different chains with the code written in Exercise 2 and analyse the chains. Discuss your results and compare to the result of Exercise 1. Use at least a number of samples that $R - 1 < 0.05$.

3 Questions for preparation

The following questions can serve as a basis for your preparation:

1. What is the physical observable allowing us to determine the cluster mass? How is it measured?

2. What is the difference between a χ^2, a likelihood and a posterior?

3. Why doesn't the Metropolis-Hastings algorithm accept every step with a better likelihood?