Draft version May 13, 2019
Typeset using LaTeX default style in AASTeX62

Senior Thesis: Reconstructing Log-normal Density Fields using Hamiltonian Monte Carlo Techniques

Abigail Lee[1]

[1] *University of Pennsylvania, Department of Physics and Astronomy*

1. INTRODUCTION

How can we make inferences about how dark matter is distributed given that we cannot observe it directly? One such method is to use observations from galaxies and use Bayesian analyses to make inferences about the cosmic density field. Next-generation surveys such as LSST will rapidly produce large amounts of galaxy and lensing data which can provide information about the cosmological parameters, which includes the percentage of dark matter in the universe. Currently, in order to constrain cosmological parameters, cosmologists use methods such as measuring two-point correlation functions and the power spectrum of shear from lensing data, which is directly proportional to the power spectrum of dark matter. However, the power spectrum only completely specifies the statistical properties of the cosmic density field if the density field is gaussian. It is known that the initial density field was once well described by a Gaussian density field in standard cosmological pictures, but as the density field evolved, that is no longer the case. The cosmic density field $\rho(x)$ is now non-Gaussian, particularly on scales below $\sim$10 Mpc. One large reason for this is that if the fully evolved density field was a Gaussian density distribution, then this would allow for negative densities and would therefore cause negative mass, i.e the Gaussian description must fail when the fluctuation amplitude is O(1), because the density field must be ≥ 0 . Therefore, the power spectrum no longer completely specifies the statistical properties of the cosmic density field, and a new technique for better constraining the cosmological parameters is necessary in order to fully measure the statistical properties of the cosmic density field than just measuring the power spectrum of shear from lensing data.

According to observations and theory, the fully evolved non-linear matter fields statistical behavior is better described with a lognormal probability distribuution than a Gaussian distribution (Jones). However the log-normal field is not a perfect description of the cosmic density field because there are no filaments in a log-normal field, and we do not really know that the galaxies occupy the field by Poisson process. However for our purposes, a log-normal distribution is a good approximation for the cosmic density field.

Additionally, if the non-linear non-gaussian density field is to be used for scientific purposes, then it would be useful to completely reconstruct the density field instead of just inferring information about the mean or maximum from the power spectrum. For this reason, the Hamiltonian Monte Carlo sampling algorithm is a useful method for completely reconstructing non-linear density fields because the HMC method provides a sampled representation of the density posterior.

In this senior thesis, I demonstrate that the HMC can efficiently sample the lognormal Poissonian posterior of the cosmic density field even in high-dimensional spaces. Additionally, because I successfully reconstructed the entire density field, any statistical information such as mean or variance can be easily calculated from all of the HMC samples.

2. MARKOV CHAIN MONTE CARLO METHODS

In statistics, MCMC methods comprise a class of algorithms for sampling from a probability distribution $P(x)$, from which directly sampling is difficult.

For our purposes, we employ Metropolis Hastings Algorithm by calculating the posterior and comparing it to the posterior of the step before it in order to determine whether to reject or accept that step. In our case for the sampling of cosmic density fields, we are calculating the posterior $P(s_{i,j}|galaxies_{i,j})$, where s_i is the lognormal cosmic density field, and $galaxies$ is an $l \times l$ galaxy field where there is a galaxy count in each cell.

In conventional MCMC algorithms, the Markov chain is generated by using a proposal distribution for new steps. The Conventional MCMC Metropolis-Hastings algorithm is:

Let $P(x)$ be the target distribution which you are trying to sample.

1. Initialization: choose an arbitrary x_0 to be the first sample

2. For each iteration t:
   (a) Generate a candidate x' for the next sample by picking from a symmetric distribution $g(x'|x_t)$, usually chosen to be a Gaussian distribution.
   (b) Calculate an acceptance ratio $\alpha = P(x')/P(x_t)$
   (c) Generate a uniform random uniform number u on [0,1]
      i. If u $\leq \alpha$, accept candidate and $x_{t+1} = x'$
      ii. If u $> \alpha$, reject the candidate and set $x_{t+1} = x_t$ instead

3. HAMILTONIAN SAMPLING

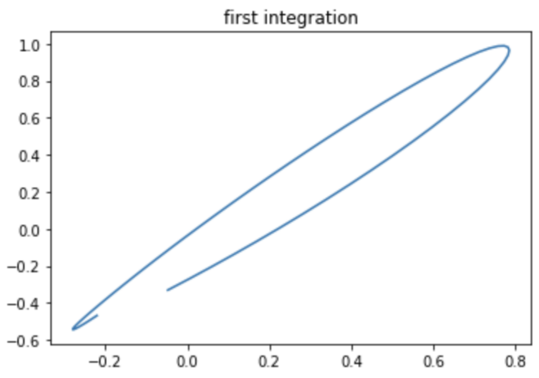
In the HMC sampling algorithm, we employ a Metropolis-Hastings algorithm, which in literature is often referred to as the Hybrid Monte Carlo method (MCMC Handbook). This HMC method exploits techniques that follow classical dynamical particle motion as analogized in section III.

If we want to draw samples from a posterior probability distribution $P(x)$, we can adopt the potential energy $U = -\ln(p(q))$. Furthermore, we artificially introduce a momentum variable p and a mass matrix M. Here, M is a symmetric, positive-definite matrix, which I will discuss later. M is typically diagonal (in order to save computation time) and is often just a scalar multiple of the identity matrix. This allows us to formulate a Hamiltonian describing the dynamics of a system, where:

$$H = \frac{1}{2}p^T M^{-1} p + U \tag{1}$$

We can then produce a Markov chain in which each iteration resamples the momentum and then does a Metropolis update with a proposal using Hamiltonian dynamics. Each iteration of the HMC algorithm has two steps. The first step is a Gibbs sampling step which draws from $P(p|q)$ and draws a momentum from a Gaussian distribution (which uses the mass matrix as its covariance matrix and the zero vector as its mean), independent of the current values of the position variables. Here $P(p,q) = e^{-H}$. In the second step, Hamiltonian dynamics is simulated for L steps using the leapfrog approximation method with a step size of ϵ. $L\epsilon$ can be thought of as an integration time, which must be discretized. Here, we start with a half step for the momentum variables, then do a full step for the position variables, using the new values of the momentum variables, and then finally do another half step for the momentum variables, using the new values for the position variables.

$$\begin{aligned} p_i(t+\epsilon/2) &= p_i(t) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t)), \\ q_i(t+\epsilon) &= q_i(t) + \epsilon\frac{p_i(t+\epsilon/2)}{m_i}, \\ p_i(t+\epsilon) &= p_i(t+\epsilon/2) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t+\epsilon)) \end{aligned}$$



In the plot above, one can see an example of one integration. The proposed state q_i is accepted as the next state of the Markov chain with probability:

$$min[1, exp(-H(q', p') + H(q, p))] = min[1, exp(-U(q') + U(q) - K(p') + K(p)] \tag{2}$$

If the proposed state is not accepted, the next state q_{i+1} is the same as the current state q_i (and is counted again in the sampling). This is the Metropolis-Hastings sampling mechanism. The two criteria that the Metropolis-Hastings algorithm places on the proposal distribution $p(q_{i+1}|q_i)$ are:

1. That it is ergodic, meaning that any place in the parameter space can eventually be reached. The potential problem of nonergodicity is usually solved by randomly choosing L or ϵ from a fairly small interval in order to prevent exact periodicities from occuring.

2. That detailed balance is satisfied, i.e that the Metropolis update is reversible in that the leapfrog step algorithm above is reversible

4. HAMILTONIAN DYNAMICS ANALOGY

Here, I will explain the Hamiltonian Monte Carlo algorithm using a classical Hamiltonian dynamics analogy, as a physical interpretation can provide useful intuitions. In two dimensions, we can visualize the dynamics of that as a ball being shot from a cannon with a random direction and then rolling back around a potential well. After we mark a stopping place after a pre-determined integration time, we shoot the ball out of the cannon again and let it roll again, and mark its place, and then repeat.

The state of this system consists of the position of the ball, q, and the momentum of the ball (its mass time velocity), given by p. The potential energy, $U(q)$, of the ball is proportional to the height of the ball and its kinetic energy $K(p)$, is equal to $|p|^2/(2m)$. On a rising slope of the potential well, the ball will continue to roll up with its kinetic energy decreasing until it is zero, at which point it will slide down again. The integration time is a tuning parameter in the HMC algorithm that determines when to stop the rolling and shoot the ball with a new initial kinetic energy.

For our applications, the position of the ball will correspond to a single estimate of the density field; if I choose to have 100 samples, then the vector q will have a size of 100 $\times$ the number of pixels in the density map.. The potential energy is $-\log(p(q))$ where $p(q)$ is the posterior probability distribution we are trying to sample (not to be confused with momentum p). The initial momentum (i.e. the initial direction and magnitude of the ball being shot from the cannon) is introduced artificially and drawn randomly from a normal distribution.

5. HMC VS. CONVENTIONAL MCMC

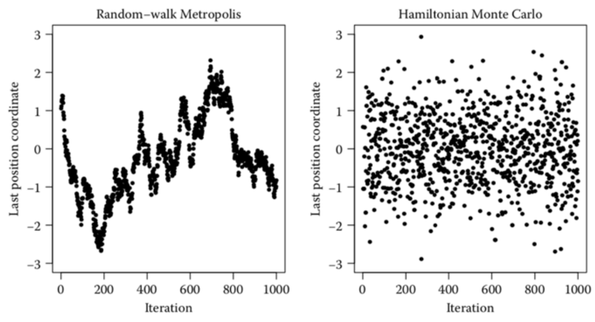
In this section, I will demonstrate the benefits of using the Hamiltonian Monte Carlo (HMC) algorithm over the more widely-known standard Monte Carlo Markov Chain (MCMC) algorithm, in which proposals are chosen from a fixed local distribution, meaning $p(q_{i+1}|q_i)$ is symmetrical around a mean of 0. The largest problem with this is that it often leads to random walking because each proposal step is calculated from a symmetric distribution centered at the previous step, meaning it is difficult to efficiently sample high-dimensional spaces quickly because samples are highly correlated.

Therefore, a new Monte Carlo method called the Hamiltonian Monte Carlo method has proved to be more efficient in sampling higher-dimensional posteriors because it suppresses random walk (MCMC Handbook), as the HMC introduces a persistent motion of the Markov chain through parameter space by introducing the momentum vector, meaning samples are weakly correlated as we allow for long-range changes between samples. This means that HMC allows you to travel a long way across the space of variables without changing the probability, thus leading to acceptance nearly all the time because in Hamiltonian Dynamics the Hamiltonian is conserved.

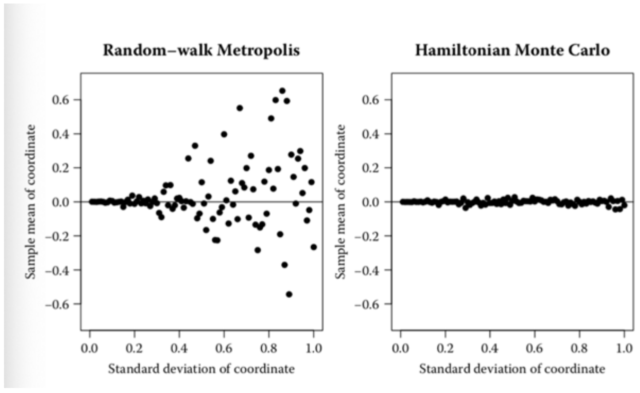
A reminder that $H = -\log(p(q)))$ so probability is conserved ($\frac{\partial H}{\partial t} = 0$). For Metropolis updates using a proposal found by Hamiltonian dynamics, the acceptance probability is one if H is kept invariant. However, for our purposes we are using the leapfrog method (discussed in section 3), which is only an approximation to exact Hamiltonian dynamics, so H is not exactly invariant.

Below are some plots showing the obvious benefits of the HMC algorithm over the more widely used MCMC algorithm taken from the MCMC Handbook Chapter 5. Simple Gaussian distributions were used as the posterior distribution, where the Gaussian being sampled is a 100-dimensional multivariate Gaussian distribution with independent variables, means of zero, and standard deviations of $0.01, 0.02, ..., 0.99, 1.00$.

First, the figure below shows the trajectory of one position coordinate over 1000 samples. This shows that the autocorrelation is clearly much higher for random-walk Metropolis than for HMC. It is bad to have a high correlation length for an MCMC because this means the samples are unrepresentative in the short run of the true underlying posterior distribution.



In the second figure, the sample means for a 100-dimensional sample after 1000 samples were taken are shown. One can see here that except for the first few variables (which had the smallest standard deviations), the error in the mean estimates from HMC is roughly 10 times less than the error in the mean estimates from random-walk Metropolis. Therefore, the HMC outperformed the MCMC algorithm as it made more accurate estimates and had less correlated samples.



6. LOGNORMAL POISSONIAN POSTERIOR

Sampling the density field of the Universe requires that there is some observable tracer, which in our case is the galaxies, which tends to follow the density of matter. It is important to note that we are adopting this model for the mass and galaxies as an approximation of the real universe, so while it is a good approximation, this is not the only possible model. Therefore, the likelihood in our sampling algorithm is a Poissonian likelihood and the prior is a lognormal prior, which results in a lognormal Poissonian posterior.

The lognormal prior is given by a multivariate lognormal distribution, where s_k is the lognormal density field, Q is the covariance matrix of the lognormal distribution, and k is an index into the discrete pixels that we are using to model the density field:

$$P(s_k|Q) = \frac{1}{\sqrt{2\pi \det(Q)}} e^{(-(1/2)\sum_{ij} \ln(1+s_i)+\mu_i] Q_{ij}^{-1} [\ln(1+s_j)+\mu_j])} \prod_k \frac{1}{1+s_k} \tag{3}$$

Here, μ_i is the constant mean field.

The corresponding probability distribution for the likelihood for the galaxy distribution can be described as a specific realization drawn from a Poission process. The corresponding probability distribution is given by the following,where N_k is the number of galaxies in pixel k. λ_k is the expectation value of N_k.

$$P(N_k|\lambda_k) = \prod_k \frac{(\lambda_k)^{N_k} e^{-\lambda_k}}{N_k!} \tag{4}$$

Note that the galaxies are being drawn from a lognormal density field, where $\bar{N}$ is the mean number of galaxies per pixel:

$$\lambda_k = \bar{N}(1 + s_k) \tag{5}$$

These together yield the the lognormal poissonian posterior

$$P(s_k|N_k) = \frac{1}{\sqrt{2\pi \det(Q)}} e^{(-(1/2)\sum_{ij}[\ln(1+s_i)+\mu_i]Q_{ij}^{-1}[\ln(1+s_j)+\mu_j])} \prod_k \frac{1}{1+s_k} \prod_k \frac{(\lambda_k)^{N_k} e^{-\lambda_k}}{N_k!} \tag{6}$$

However, this posterior is greatly simplified (and speeds up later calculations as I will show) if we introduce a change of variables $r_k = \ln(1 + s_k)$, as this changes the field to a Gaussian Random Field.

7. EQUATIONS OF MOTION FOR A LOGNORMAL POISSONIAN SYSTEM

It is important to note that since the potential energy and kinetic energy must be re-calculated for each sample, it is beneficial to find a way to speed up this calculation. We found that the gaussian prior can be calculated much faster in Fourier space because it contains a term involving the covariance matrix, which in Fourier space is a diagonal matrix for a gaussian distribution, instead of a 2-dimensional covariance matrix in real-space. This changes our calculations of terms involving the power spectrum S from $O(N^2)$ to $O(N)$ where $N = l^2$. If we want to transform our real-space variables r into their Fourier transform coefficients a, there is a simple linear transformation: $a = Fr$, and has a diagonal covariance matrix S (where F is a special FFT matrix), so our gaussian prior in Fourier space can be written as:

$$\log(P(a)) = \frac{1}{2}\sum \log(S) + \frac{a^2}{S} \tag{7}$$

If we are taking the power spectrum as a known, then the log S terms are constant and can be dropped. This means we are multiplying an $l \times 1$ matrix by an $l \times l$ matrix $(a^2 S^{-1})$ instead of multiplying an $l \times l$ matrix by an $l \times l$ covariance $(ln(1 + s_i)^2 * Q^{-1})$ matrix in real-space.

Therefore, our potential energy can be written as (dropping constant terms and setting bias $b = 1$ for now):

$$U = N_{tot} * log \sum e_k^r - \sum N_k * r_k + \frac{1}{2}\sum \frac{a_k^2}{S} \tag{8}$$

Where N_{tot} is the sum of all the galaxies in the galaxy field.

The gradient of this potential is the derivative with respect to r of U, $\frac{\partial U}{\partial r}$. Again, if we take the gaussian prior in Fourier space, then we end up with an $l \times 1$ times an $l \times l$ matrix multiplication instead of an $l \times l$ times an $l \times l$ matrix. That is, we are multiplying by an $l^2 \times l^2$ matrix in the general case, or just l^2 multiplications for the diagonal case. Therefore, the gradient $\frac{\partial U}{\partial r}$ can be written as:

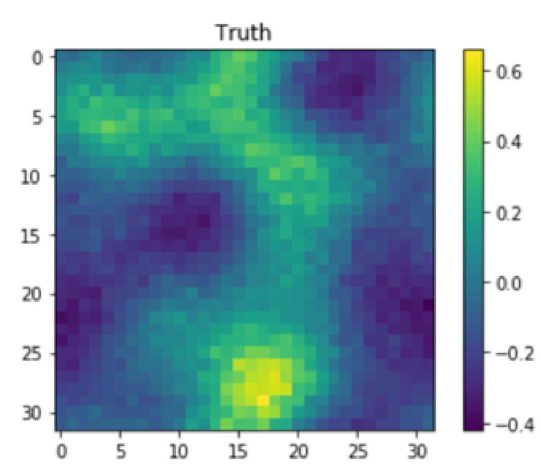
$$\frac{\partial U}{\partial r} = \frac{N_{tot}}{\sum e^r} * e^r - N_k + F^{-1}(\frac{a}{S}) \tag{9}$$

Therefore the Hamiltonian equations of motion $dq/dt = dH/dp$ and $dp/dt = dH/dq$ can be easily obtained and integrated numerically to simulate dynamical particle motion. The integration of these equations of motion yields the new position q_{i+1} and p_{i+1}. This new point is accepted or rejected according to the Metropolis Hastings algorithm described in part 3.

8. SETTING UP MOCK OBSERVATIONS

For the purpose of this thesis, I generated lognormal random fields and poissonian lognormally galaxy tracers by using the following schema:

1. **Starting with a gaussian power spectrum:** $P(k, c, A) = A * e^{-k^2 c^2/2}$, where A is the amplitude and the k-values go from -$l/2$, $l/2$, where l is the length of the grid size. c can be thought of as a parameter in the power spectrum corresponding to how many large-scale structure there are. A lower c means a greater number structures. The plot on the left for a 32×32 pixel grid shows c=1. And the plot on the right shows c=0.5. The third plot shows the 1-dimensional power spectrum I used.



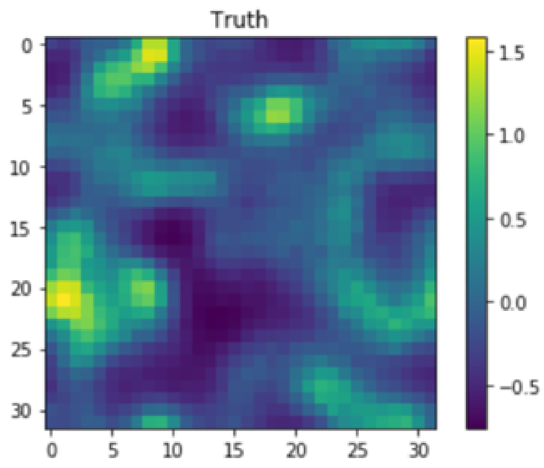
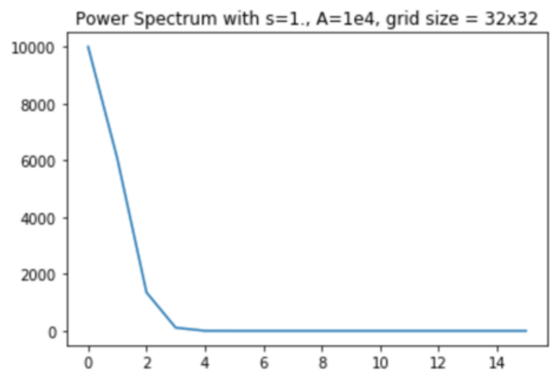


Figure 1. Plot on the left shows c=1, and the plot on the right shows c=0.5.



2. **The Gaussian Random Field** $r(x)$ is the Fourier Transform of a field whose coefficients are drawn at random from a multivariate normal with the power spectrum which is the diagonal of the covariance matrix.

3. **The Log-normal random field** $\delta(x) = e^{(r(x)-\sigma^2/2)} - 1$, where σ^2 is the variance of the Gaussian Random Field.

4. **The Lognormal Poissonian Galaxy Catalog** is generated by each specific realization drawn from a Poisson process where the number of galaxies per cell is randomly drawn from a Poisson Distribution (see equation 4).

8.1. *Normalizing log-normal mass distributions*

We will want to normalize the log-normal distribution, where $\rho(x)$ is the density field. If we want to have a zero-mean Gaussian random field such that the density $\rho(x)$ is: $\rho(x) = \rho_0 * e^{r(x)}$, we must normalize the lognormal field.

For our Gaussian random field, we know the variance σ^2 is: $\sigma^2 = \langle r^2 \rangle - \langle r \rangle^2$. But $\langle r \rangle^2 = 0$ because the Gaussian random field is zero-mean, so therefore $\sigma^2 = \langle r^2 \rangle = C(0) = \int P(k) d^2k$, where $C(0)$ is the 2-point correlation function at zero lag. Note that $\langle e^r \rangle = e^{\langle r^2 \rangle}$ for gaussian r. This means that:

$$\frac{\rho(x)}{\bar{\rho}} = 1 + \delta(x) = \frac{\rho_0 e^{r(x)}}{\rho_0 e^{\sigma^2/2}} = e^{r-\sigma^2/2}. \qquad (10)$$

Thus, we should subtract the value $\sigma^2/2$ from our r function in order to normalize $\delta(x)$.

However, if $\sigma^2 = \int P(k) d^2k$, then here $\sigma^2 = 2\pi A/s^2/l^4$.

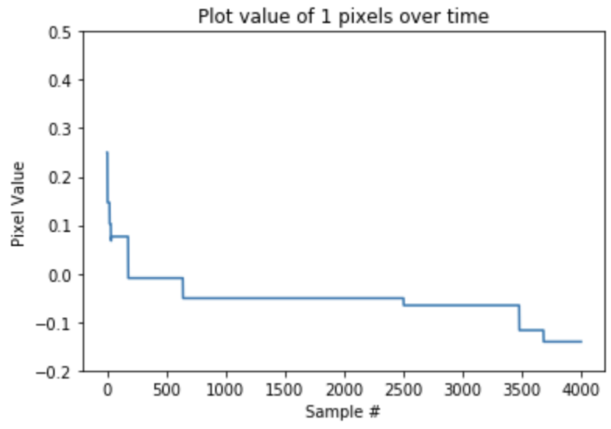


Figure 2. An example where ϵ is too large

8.2. *Mass Matrix*

The Hamiltonian sampler uses the mass matrix M, which can greatly influence the sampling efficiency by making certain directions more probable, as M determines the shape of the multivariate Gaussian determining the cannon shots from p. In our case, we can choose $M = C^{-1}$ where C is the covariance matrix of the posterior. However, in order to save computation time, we choose C to be diagonal, and we let it be a scalar multiple of the identity matrix, σ^2*Id. This changes the matrix multiplication of $p^2 * M^{-1}$ which involves a matrix multiplication of $l \times l$ times $l \times l$ to a matrix multiplication of $l \times l$ times $l \times 1$.

9. TUNING HMC

There are two main things that need to be tuned in the use of the HMC, which are the leapfrog stepsize ϵ, and the number of leapfrog steps L, which together determine the length of the trajectory ϵL

Tuning the HMC sampler requires preliminary runs with trial values for ϵ and L. Doing several runs is advisable as selecting proper ϵ and L are crucial.

9.1. *Choosing ϵ*

Too large a stepsize ϵ will result in a low acceptance rate because this will raise (or lower) H (see equation 2), resulting in a negative likelihood and a very small chance of acceptance. Thus, we never move away from the initial guess, and too small a stepsize will waste computation time and lead to an unnecessarily slow convergence. In order to choose an adequate ϵ, one can plot the samples of a single pixel and look at the acceptance ratio (how many samples were accepted/total number of samples). A good acceptance rate is above .5 for the HMC algorithm.

Fig. 2 is an example of an ϵ that is too large in the 32x32 example. An ϵ of 0.001 for 100 samples results in an acceptance rate of .02 and the following plot which tracks the most upper left pixel of the grid. One can see that the pixel barely moves and when it does move, the movements are choppy.

The next plot Fig. 3 is an example of when the choice of ϵ is too low. Here, lowering the value of ϵ to 0.000001 raises the acceptance rate to 1.0, but the pixel barely moves over 100 samples:

Finally, Fig. 4 is a good example of a good choice of an ϵ =0.0001, where the pixel is moving considerable yet has a reasonable acceptance rate of 0.88.

9.2. *Choosing the trajectory length L*

Additionally, choosing a suitable trajectory length L is crucial if HMC is to adequately explore the phase space. If the trajectory length ϵL is too short, then the samples will be correlated because the sample does not have a long enough time to explore the state space, and then we are back to random-walking. In order to select an adequate L, one can observe the full integration length of the first sample and the trajectory of a single pixel after choosing ϵ. Randomly varying the length of the trajectory over a fairly small interval is desirable to avoid choosing a trajectory length that happens to produce a near-periodicity for certain pixels (i.e. the pixel ends up in the same place each time). That is, trajectories for a pixel will return to the same place each time when ϵL is approximately 2π.

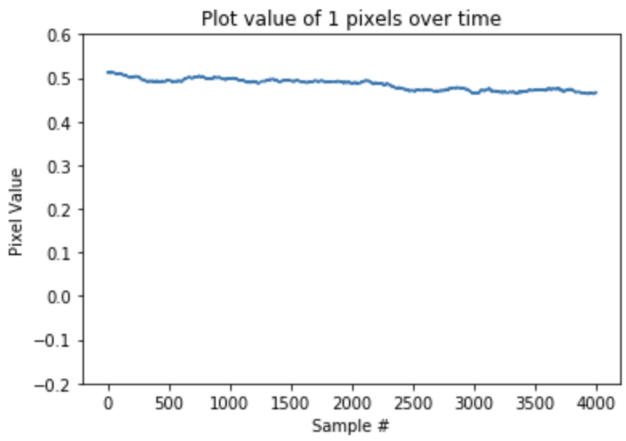


Figure 3. An example where ϵ is too small

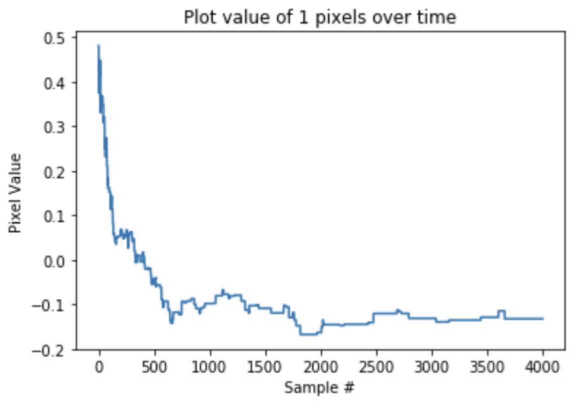


Figure 4. An example where ϵ is well chosen

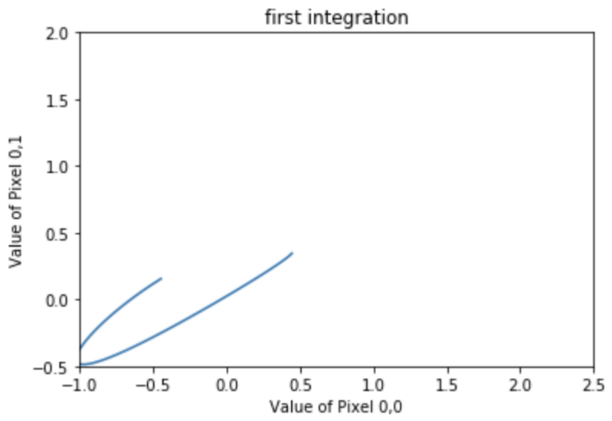


Figure 5. An example where L is well chosen

When looking at the trajectory of the first sample of one pixel, it is ideal that the trajectory looks approximately periodic and makes one half rotation, since we dont want to end up where we started and we want to move across the posterior. For example, Fig.5 is a good example of a well-chosen combination of ϵL:

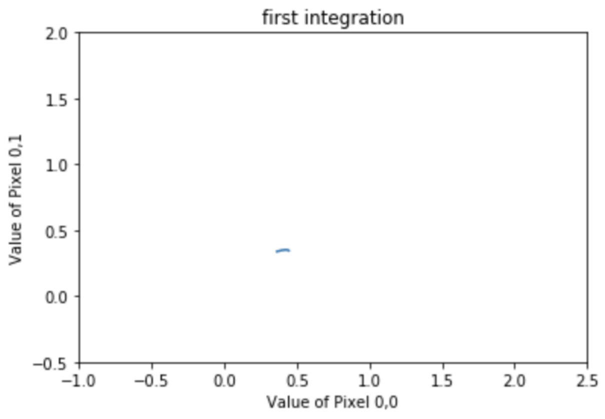


Figure 6. An example where L is too small

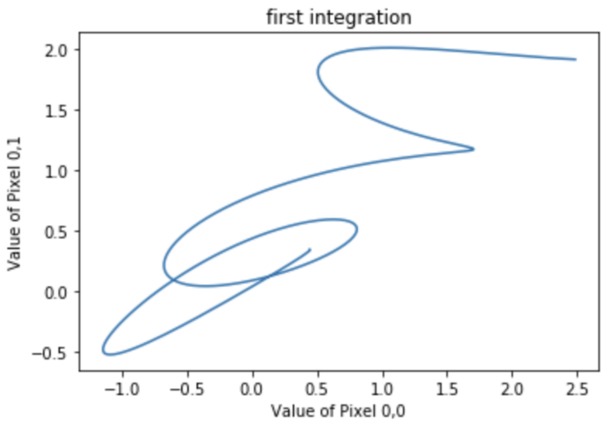


Figure 7. An example where L is too long

Fig. 6 is bad choice of ϵL, because the sample has not reached one full integration:

Fig. 7 is also a bad choice of ϵL, because the sample has completed many full periods and is therefore wasting computing time.

One can also observe the value of a single pixel, to observe whether the samples look uncorrelated or not. For example, in Fig. 9, one can see that the samples look very correlated:

Therefore, selecting ϵ and L should be done by trial and error, and looking at the preliminary statistics of observing a single pixel over time and the full first integration of one step.

10. NUMERICAL IMPLEMENTATION

In this section, I demonstrate that this HMC algorithm was able to successfully reconstruct the true non-linear density field, which we have as known here because we created the galaxy catalog from the truth. The HMC sampler is given a Gaussian power spectrum, a lognormal-poissonian galaxy catalog (which was generated from the true density field).

Both the truth and the simulated galaxy catalog are l=512 grids. The HMC sampler took about 5.6 minutes to run on a PC. Here, we chose a stepsize of $\epsilon = 0.001$, and an integration of length L was chosen from a uniform distribution U(500,510). 100 samples were taken, and there was an acceptance rate of 0.40.

In Fig. 9, one can see that the truth is almost identical to the reconstruction of the cosmic density field done by the HMC algorithm.

Figure 10 is an example of the l=32 case wih s=1., and an acceptance rate of .68.

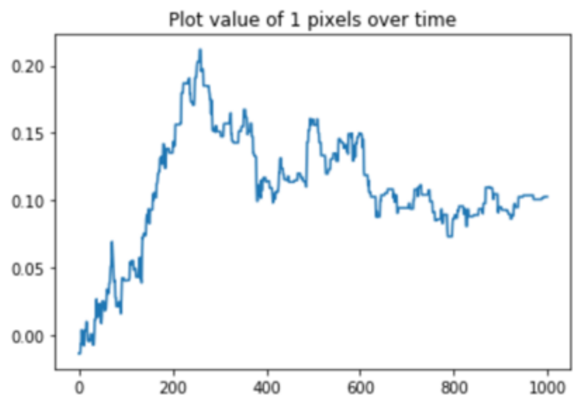


Figure 8. The samples look very correlated, therefore one should increase L

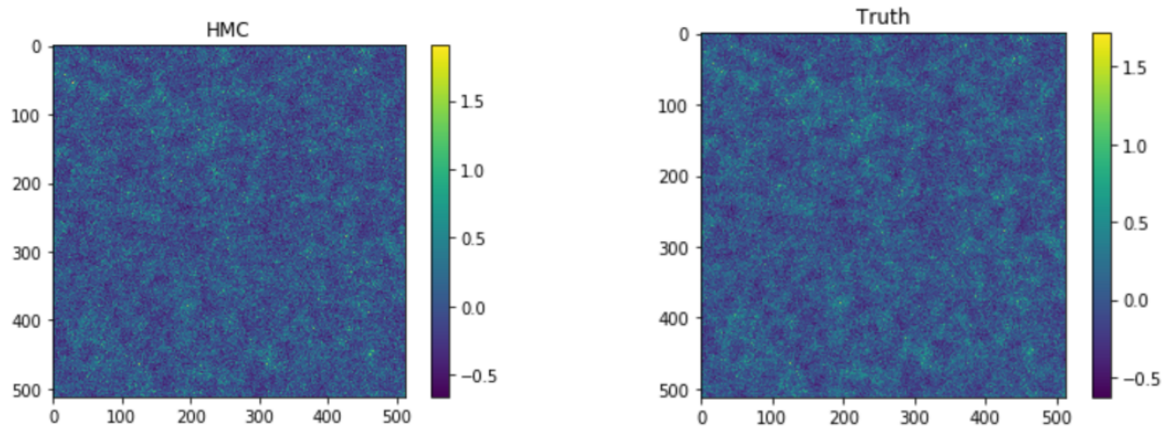


Figure 9. A 512x512 grid where the last sample of the HMC algorithm is compared to the lognormal truth from which the galaxy field was generated from

Figure 11 shows the galaxy catalog that the HMC sampler was given, as well as the standard deviation of the last 200 samples in the chain.

10.1. *Burn-in behavior*

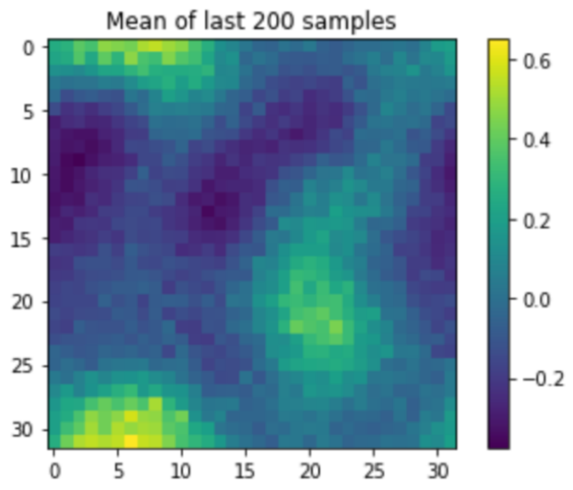
The Hamiltonian sampler provides samples from the correct probability distribution, but because the sampler must start somewhere at a randomly chosen initial step chosen from a random lognormal field, there will be an initial burn-in phase. In order to test whether the Markov chain has successfully burned-in, we can look at the value of the likelihood and a value of a single pixel to test for convergence. For example, Fig. 11 shows a chain that has burn-in at around 40,000 samples.

One can also look at whether the likelihood (the total change in kinetic energy + potential energy) has converged as well, like the plot in Fig. 12.

11. CONCLUSION

In this paper I present a numerically efficient Bayesian method for large-scale structure inference which provides a sampled representation of a very high-dimensional non-Gaussian large-scale structure posterior. This permits me to easily calculate any statistical summary such as mean or variance of the cosmic density field samples.

I am going to continue to work on this project later on in the summer. Future work includes having bias and the power spectrum as free parameters. I would also like to implement the Gelman-Rubin test for convergence as another way of testing whether the Markov chain has converged or not.



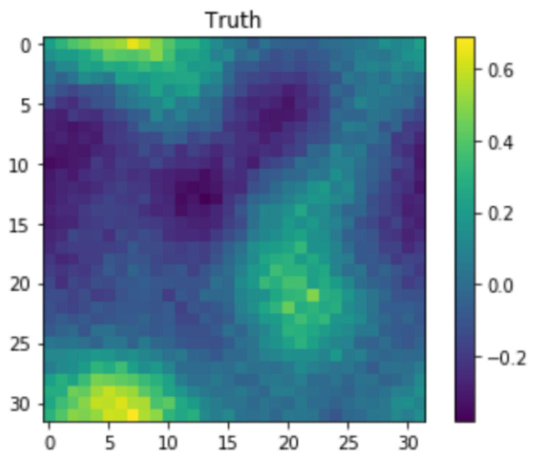
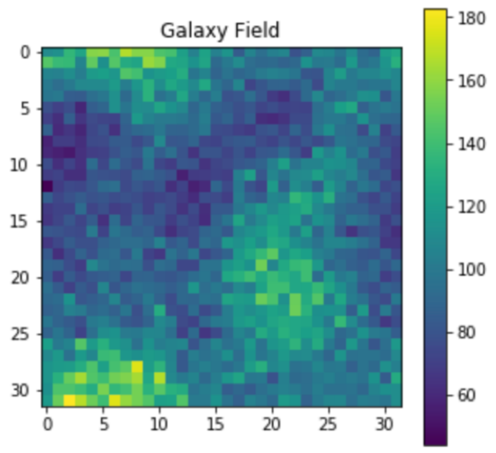


Figure 10. A 32x32 grid where the truth is compared to the mean of the last 200 samples generated from the HMC sampler



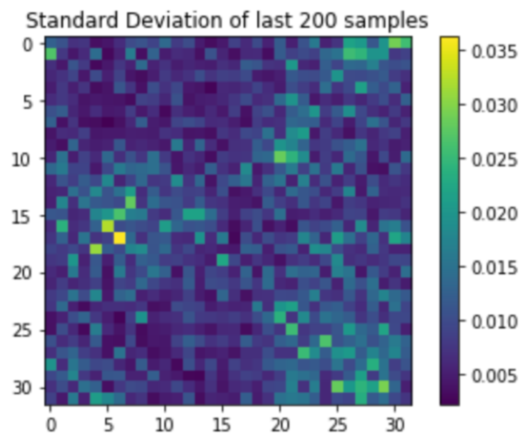


Figure 11. The lognormal Poissonian galaxy field from which the HMC sampled from, and the standard deviation of the last 200 samples

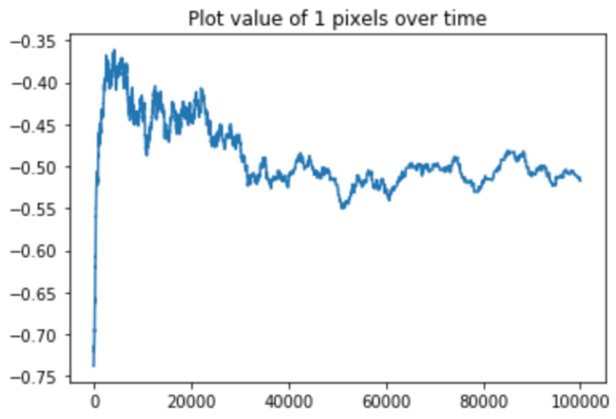


Figure 12. By tracking one pixel for all the samples, one can observe the burn-in period of this chain, which lasts until about the 40,000th sample

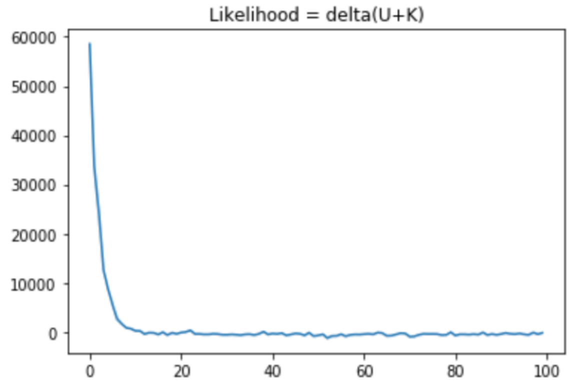


Figure 13. By tracking the likelihood, one can also observe the burn-in period

Finally, the over-arching final goal of this project is to apply the HMC method to the problem of estimating the redshift distribution $n(z)$ of a population of galaxies in an imaging survey by observing their clustering patterns in comparison to a second, smaller tracer population of galaxies that have accurately known redshifts. Sanchez and Bernstein in their paper (Sanchez and Bernstein) write down a joint posterior probability distribution for the $n(z)$ of the unknowns and the density field s_i that both the tracers and the unknowns live in. The HMC will make it possible to sample from this joint distribution. I would also like to test whether this HMC sampler works on an N-body simulation where the density field is not exactly lognormally distributed. Then, the final goal would be to test this on real data, such as galaxy data from DES or LSST.

12. ACKNOWLEDGEMENTS

I would like to thank Dr. Gary Bernstein for his mentorship and support over the course of this senior thesis.

REFERENCES

Jasche, J., Kituara, F. Fast Hamiltonian sampling for large-scale structure inference, arXiv:0911.2496.

Carles Sanchez and Gary Bernstein, Redshift inference from the combi- nation of galaxy colors and clustering in a hierarchical Bayesian model. arXiv:1807.11873.

Brooks, Steve, Andrew Gelman, Galin L. Jones, Xiao-Li Meng, Handbook of Markov Chain Monte Carlo.

Peter Coles and Bernard Jones, A lognormal model for the cosmological mass distribution, 1991MNRAS.248....1C.