Fast inference for Cosmology

Raul Soutelo Quintela

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2017

Abstract

Bayesian methods have been commonly used to infer the parameters of cosmological models. They allow us to asses how plausible a set of parameters are, based on how well these parameters justify the observed data. Due to the high complexity of cosmological models, Markov Chain Monte Carlo (MCMC) methods are needed to explore the probability distribution of the model parameters.

The computational cost involved in evaluating a set of parameters of a cosmological model depends on the parameters that have been modified with respect to the previous evaluation. If most computation can be reused, the parameters are considered fast; whereas if we need to perform all the computation again, the parameters are considered slow. Additionally, these models have some nuisance parameters that are not of interest. Although nuisance parameters are usually fast, it is not always the case. Previous work has focused on creating methods that allow us to explore the slow parameters as if the fast parameters are analytically marginalizable.

In this project we have devised a Pseudo-Marginal method to ideally marginalize the fast parameters and it also allows to marginalize the nuisance parameters that are slow. Furthermore, our method provides a way to parallelize some computation for machines with multiple cores. The proposed method was tested on a synthetic posterior distribution, intended to be representative of the posterior distribution of a model realizable by the CosmoSIS package. The results show that the proposed method works better than the existing methods for the model selected, and it is expected to have larger margins in more complex models.

Acknowledgements

I would like to thank my supervisor Iain Murray for introducing me to this field and providing me with great support through all the dissertation.

I would also like to thank my family for all the support, without them this year wouldn't have been possible.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Raul Soutelo Quintela*)

Table of Contents

Chapter 1

Introduction

Bayesian methods are used to compare physical models to data. In cosmology each model corresponds to a different theory. Cosmological models have many parameters. In order to test a model, we need to estimate the probability distribution of its parameters. Estimating these parameters is becoming more complicated due to the sophisticated models needed to face the incoming challenges (Zuntz et al., 2015).

A cosmological model expresses the probability of observed data D for each configuration of its parameters $p(D|\theta)$. In the Bayesian approach, the probability of the parameters given the observed data D is proportional to how well the parameters justify the data reweighted with our prior beliefs on the model parameters: $p(\theta|D) \propto p(D|\theta)p(\theta)$. For complex models, it is not possible to compute $p(\theta|D)$ analytically. Sampling methods are used to obtain independent samples from the posterior distribution $p(\theta|D)$, allowing to extract any necessary information about $p(\theta|D)$. Markov Chain Monte Carlo (MCMC) methods are used to sample from $p(\theta|D)$ in cosmology. A more detailed description is found in Section 2.1.

In this project we will explore how to efficiently generate independent samples from $p(\theta|D)$. MCMC methods produce a Markov chain of settings of θ, with $p(\theta|D)$ as its equilibrium distribution. Consecutive settings of θ may be highly correlated. We need to obtain independent samples from this distribution, so we may have to discard some values of θ between two valid samples. All methods considered in this project will use the Metropolis-Hastings updates to generate the elements θ of the Markov chain. This algorithm, explained in Section 2.2, proposes a new configuration of θ and accepts it or not depending on how plausible these parameters are. Evaluating $p(\theta|D)$ is computational expensive. The objective is then to propose plausible and independent configurations of θ.

Different tools have been developed for estimating the parameters of cosmological models. In this project, we will consider CosmoSIS that encourages modularity and allows to easily integrate different approaches for the processes involve: physics, likelihood and sampler libraries. Due to the existing multi-core machines and multi-processing platforms, the samplers that allow to calculate $p(\theta|D)$ in parallel are of special interest (Zuntz et al., 2015). This project will identify a model realizable by this package, run different MCMC methods to explore the probability distribution of the parameters $p(\theta|D)$, and make recommendations for its future development.

Cosmological models have two properties that are fairly generic:

- Computing $p(\theta|D)$ depends on the parameters in θ that have been modified with respect to the previous evaluation. Lewis and Bridle (2002) introduced the idea of fast and slow parameters in cosmological models. Evaluating $p(\theta|D)$ when fast parameters are modified allow to reuse some of the computation, whereas calculating $p(\theta|D)$ when slow parameters are modified implies to do all the computation again.

- Cosmological models have cosmological and nuisance parameters. The nuisance parameters are not of interest. Ideally, we would like to analytically marginalize over the nuisance parameters and generate independent samples from $p(\theta_{cosmo}|D) = \int p(\theta_{cosmo}, \theta_{nuisance}|D) d\theta_{nuisance}$. Unfortunately, this is usually not possible and we need a method that efficiently samples from the joint distribution $p(\theta_{cosmo}, \theta_{nuisance}|D)$ and discard the values of $\theta_{nuisance}$.

Lewis and Bridle (2002) and Lewis (2013) have developed methods to exploit the different speed of the parameters by performing extra updates of the fast variables. Neal (2011) proposed to have an ensemble of states with different settings of the fast parameters but only one setting of the slow variables. Andrieu and Roberts (2009) introduced the idea of Pseudo-Marginal MCMC that allows to ideally marginalize the parameters that are not of interest. In this project we propose a Pseudo-Marginal method to marginalize the parameters that are not of interest and exploits the different speed of the parameters by creating an ensemble of states. Moreover some computation of $p(\theta|D)$ can be parallelized. The method proposed has been compared with the existing methods in the synthetic posterior distribution selected. The experimental results show that the method proposed performs considerably better and it is expected to have larger margins in more complex cosmological models.

This project is broken down as follows. First we will evaluate in Background other methods addressing the exploration of cosmological models. We will focus on the two ways of exploiting the fast and slow parameters: performing extra updates of the fast parameters and having an ensemble of states with many configurations of the fast parameters. We will also analyse the case when we want to explore a subset of variables contained in θ. Secondly, we will present the proposed method. It is a Pseudo-Marginal approach that allows to update the slow and fast variable separately allowing to use any standard MCMC update. Then, we will proceed in Evaluation to the comparison of the existing methods with the approach proposed and we will justify the results obtained. Finally, more general conclusions will be drawn motivating the comparison in more complex models.

Chapter 2

Background

In this chapter we will explain different approaches that have been used to estimate the probability distribution of the parameters of cosmological models $p(\theta|D)$. First, we will give a more detailed explanation of the problem and justify the use of MCMC methods to sample from $p(\theta|D)$. Secondly, we will then present the Metropolis-Hastings algorithm (Hastings, 1970). It will be the MCMC update used in all other methods. Then, we will explain the idea of fast and slow parameters and introduce three methods that use this information to more efficiently sample form the target distribution: Extra update Metropolis (Lewis and Bridle, 2002), Fast-slow decorrelation (Lewis, 2013) and MCMC ensemble (Neal, 2011). Finally, we will evaluate different alternatives when we want to obtain the probability distribution over some of the parameters of a model but not all of them. We will present the idea of Pseudo-Marginal methods and introduce an approach to approximately sample from $p(\theta_{slow}|D)$ when marginalizing analytically over θ_{fast} is not possible (Neal, 2005).

2.1 Problem description

Cosmological models assign a probability of observed data D for each configuration of the model parameters $p(D|\theta, M)$. Each model M corresponds to a different cosmological theory. In order to test a theory, we need to estimate the probability distribution of the model's parameters (Zuntz et al., 2015).

The likelihood of the parameters given the data $p(\theta|D, M)$ indicates how plausible a set of parameters of a model M are given the observed data D. A prior knowledge $p(\theta|M)$ is assumed over the parameters of a cosmological model. This knowledge may come from previous experiments and tries to summarize the expected values of

the parameters before seeing any data. The term $p(D|\theta,M)$ is called 'likelihood' and indicates how well a set of parameters θ justifies the data D. Prior and likelihood are combined with the Bayes rule to estimate the posterior belief over the model parameters:

$$p(\theta|D,M) = \frac{p(D|\theta,M)p(\theta|M)}{p(D|M)} = \frac{f(\theta)}{C} \tag{2.1}$$

We usually cannot evaluate $f(\theta)/C$ because we cannot compute C. By multiplying the likelihood $p(D|\theta,M)$ and the prior $p(\theta|M)$ we can obtain $f(\theta)$. This function expresses the goodness of the parameters, how likely they are given the observed data D. If we can evaluate $f(\theta)$ up to a constant, we can generate independent samples from the posterior distribution $p(\theta|D,M)$ using a sampler.

Standard methods such as rejection sampling or importance sampling directly sample from a target distribution. However these methods don't scale well with the numbers of dimensions. Cosmological models are usually high-dimensional. In order to obtain independent samples for a high dimensional probability distribution Markov Chain Monte Carlo (MCMC) methods are used (Zuntz et al., 2015).

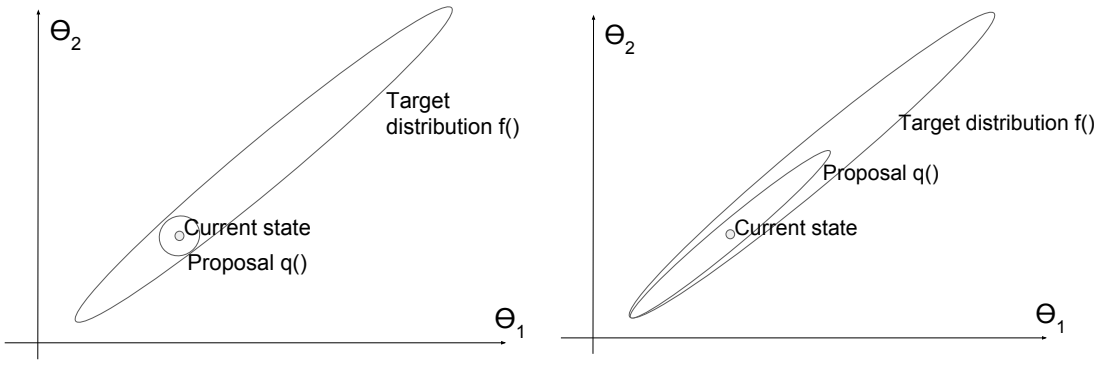
A MCMC method constructs a Markov Chain on the parameters θ that has as equilibrium distribution the target distribution $f(\theta)$. We can obtain independent samples from the probability distribution $f(\theta)$ by taking some of the states θ from the Markov Chain. Consecutive states may be highly correlated, so that we may need to discard some of them between two independent states. Obtaining these samples is computational expensive. The goal of this project is to obtain a MCMC method that provides independent samples from $f(\theta)$ with the lower computational cost possible.

2.2 Metropolis-Hastings

The Metropolis-Hastings algorithm (Hastings, 1970) explores a probability distribution $f(\theta)$ by perturbing θ and accepting or rejecting the new configuration θ' depending on the new value of $f(\theta')$. The new θ' is drawn from the distribution $q(\theta';\theta)$. This function $q()$ is called proposal distribution and it is the single free parameter of the method to be decided by the user. The new state θ' is accepted with probability

$$P(\text{accept}) = \min\left(1, \frac{f(\theta')q(\theta;\theta')}{f(\theta)q(\theta';\theta)}\right) \tag{2.2}$$

The efficiency of the Metropolis-Hastings algorithm may depend a lot of on the



(a) Spherical proposal $q()$.

(b) Alternative proposal $q()$.

Figure 2.1: Comparison of the standard spherical proposals with an alternative that sets $q()$ to a multivariate Gaussian where the covariance matrix is an approximate of the covariance of the target distribution.

proposal distribution $q()$. A standard choice, when we do not have any prior knowledge about the target distribution, is to set $q()$ to a spherical multivariate Gaussian. If we have an approximate covariance Σ of the target distribution $f(\theta)$, we can set $q()$ to a multivariate Gaussian with Σ as its covariance matrix. We show in Figure 2.1 how using an approximate of the covariance Σ of the target distribution $f(\theta)$ to tune $q()$ allows to make longer proposals while being θ still plausible (Haario et al., 1999).

Haario et al. (1999) proposes to use an adaptive method and modify $q()$ continuously depending on the Σ obtained from the previous states. The Markov chain obtained does not have as equilibrium distribution the target distribution but one that is closed. In order to obtain the exact equilibrium distribution, he proposes to run a preliminary Markov chain with the adaptive method and use this chain to obtain Σ. Once this matrix is obtained, we can run the definitive Markov chain with a fixed proposal distribution.

We will use Metropolis-Hastings to update the parameters θ in all methods explored in this project. None of the proposal distributions will be modified during the chain. The proposal $q()$ will be tuned to different distributions depending on the variables perturbed and the methods used. We will use the idea of tuning the proposal distribution $q()$ to multivariate Gaussians where the covariance matrix is an approximate of the covariance matrix of the target distribution.

2.3 Fast and slow parameters

Fortunately, when evaluating the quantity $f(\theta)$ for different values of θ some computation could be reused. The quantity depends on the parameters θ that are modified. Lewis and Bridle (2002) proposed to divide the parameters into fast and slow depending on the computational cost associated to evaluate $f(\theta)$ when these parameters are changed. If a lot of computation could be reused to evaluate $f(\theta)$ for some θ' with respect to a previous configuration of θ, the parameters changed are considered fast. Otherwise, if computing $f(\theta)$ requires to do a lot of computation again, these parameters are considered slow.

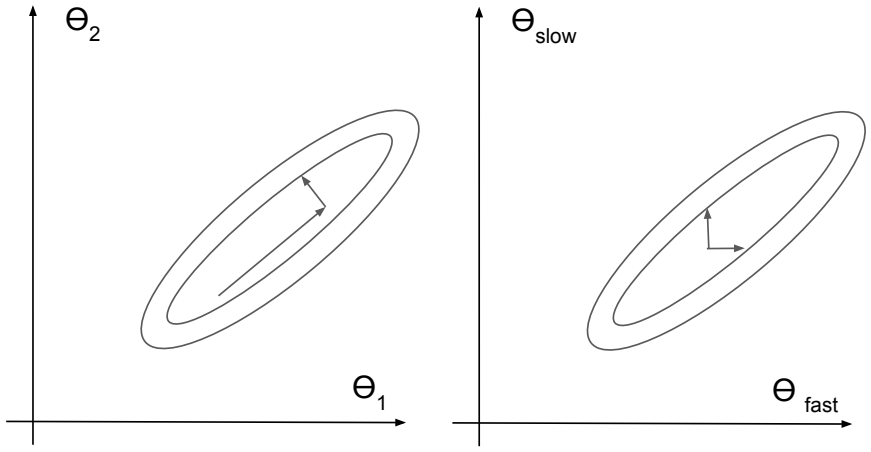
In cosmology there are parameters with different speeds. For instance, evaluating $f(\theta)$ for different values of a matter density parameter requires a lot of computation, so this parameter is considered slow, whereas computing $f(\theta)$ for different values of a power spectrum parameter allows to reuse some of the computation, so this parameter is considered fast (Lewis, 2013).

In this section we will explore different methods that divide the parameters θ in these two groups and update them in a clever way to more efficiently get independent samples from $f(\theta)$. We will first explain the extra update Metropolis (Lewis and Bridle, 2002) and Fast-slow decorrelation (Lewis, 2013) methods. They exploit the fast-slow feature by performing extra update of the fast parameters. Then we will explain the method proposed by Neal (2005) that exploits the fast-slow feature by constructing an ensemble of states.

2.3.1 Extra update Metropolis

In order to more efficiently explore the probability distribution, Lewis and Bridle (2002) proposed to update alternatively θ_{slow} and θ_{fast}, performing extra updates of the fast variables. These extra updates are cheap since a lot of computation could be reused from the previous evaluation. The extra updates are intended to fully explore the fast subspace between each update of θ_{slow}. Alternatively, perturbations of both fast and slow parameters can be combined with extra updates of the fast variables. The notation extra update Metropolis was taken from Neal (2011).

In this project, we will use both approaches: there will be methods that update both variables separately; and methods that update either slow and fast variables or just fast variables. Although this is a general idea that allows to use different updates, we will only consider Metropolis-Hastings for both updates.



(a) Proposals for θ of same speed. (b) Proposals for θ of different speed.

Figure 2.2: Different proposals depending on the relative speed of its parameters (Lewis, 2013).

2.3.2 Fast-slow decorrelation

The performance of Metropolis-Hastings methods depends on the proposal distribution $q()$. We have seen in Section 2.2 how to tune the proposal distribution $q()$ when the variables are correlated and all of them are updated at once. When dealing with fast and slow parameters we may want to update them separately, so that the update of the fast variables remains fast. We can easily use the same principle that in Section 2.2 to exploit the correlations among the fast and among the slow variables separately. However, we will be ignoring the correlation between the slow and the fast parameters. Figure 2.2 shows two possible updates depending on whether the variables have the same speed or not.

Lewis (2013) proposes to redefine the parameters to be as uncorrelated as possible but in such a way that the new fast parameters just depend on the original fast parameters. The new slow parameters will depend on the original slow and fast parameters, while the new fast parameters only on the original fast ones. The new uncorrelated parameters will be denoted ω. First, the original parameters are sorted by speed in an increasing order: θ_i is slower than θ_j if $i < j$. Then, the Cholesky decomposition of the approximate of the covariance matrix Σ is done:

$$\Sigma = \langle \theta\theta^T \rangle = LL^T \tag{2.3}$$

Lewis (2013) suggests that Σ could be obtained by running a preliminary chain with a standard method or taking the first steps of an adaptive method as Haario et al.

(1999) proposed. The new decorrelated parameters are $\omega = L^{-1}\theta$. A new move in the new space is $\omega \to \omega + \Delta\omega$, therefore in the original space

$$\theta \to \theta + L\Delta\omega \tag{2.4}$$

Being L a lower triangular matrix implies that for changing the $i-th$ decorrelated parameter is only necessary to calculate parameters that are faster than itself. In our problem, we just consider two groups: fast and slow variables. Therefore, updating one subset of variables will involve just perturbing the fast variables and updating the other subset of variables will perturb both slow and fast parameters.

Similarly to the extra update Metropolis methods, Lewis (2013) suggests to perform more frequent updates of the fast variables although he does not specify how much. This is then a free parameter to be tuned by the user. In this project, we will spend the same CPU time in both updates. It means that if the evaluation of $f(\theta)$ is f times more expensive when the slow variables are modified, we will perform on average f updates of the fast parameters for each update of the slow parameters.

It is important to notice that $L\Delta\omega$ from equation 2.4 could come from any MCMC update. Lewis (2013) proposes to perform Metropolis-Hastings updates. In this project we will only consider these updates for the Fast-slow decorrelation method.

2.3.2.1 Updates explanation

We will now provide a more detailed explanation of the updates of the Fast-slow decorrelation method. For this purpose, we will use $g()$ to denote a zero mean multivariate Gaussian with Σ as covariance matrix. If we would perturb all new variables ω, $L\Delta\omega$ would be a sample from $g(\theta_{slow}, \theta_{fast})$. By perturbing a variable we mean that $\Delta\omega$ is drawn from $\mathcal{N}(0, 1)$. Now we will analyse from which distributions $L\Delta\omega$ is drawn when only some variables of ω are perturbed as Lewis (2013) proposes for the Fast-slow decorrelation method.

Since $g(\theta_{slow}, \theta_{fast})$ has zero mean, the joint distribution $g(\theta_{slow}, \theta_{fast})$ is given by

$$g(\theta_{slow}, \theta_{fast}) = \mathcal{N}\left(\begin{bmatrix} \theta_{slow} \\ \theta_{fast} \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix}\right) \tag{2.5}$$

The conditional distribution is

$$g(\theta_{fast}|\theta_{slow}) = \mathcal{N}(\theta_{fast};\, C^\top A^{-1}(\theta_{slow}),\, B - C^\top A^{-1} C) \tag{2.6}$$

The Cholesky decomposition of the covariance matrix of the joint distribution $g(\theta_{slow}, \theta_{fast})$, presented in equation 2.3, could by decomposed in:

- L1: Cholesky decomposition of the covariance of $g(\theta_{slow})$.
- L2: $C^T A^{-1}$.
- L3: Cholesky decomposition of the covariance of $g(\theta_{fast}|\theta_{slow})$.

$$L = \begin{bmatrix} L1 & 0 \\ L2 & L3 \end{bmatrix} \tag{2.7}$$

The update of θ given in equation 2.4 as $L\Delta\omega$ depends on the subset of variables that are modified in ω.

$$\begin{bmatrix} \Delta\theta_{slow} \\ \Delta\theta_{fast} \end{bmatrix} = \begin{bmatrix} L1 & 0 \\ L2 & L3 \end{bmatrix} \begin{bmatrix} \Delta\omega_{slow} \\ \Delta\omega_{fast} \end{bmatrix} \tag{2.8}$$

If the new fast variables are modified, the update of the original fast variables $\Delta\theta_{fast}$ would be a sample from $g(\theta_{fast}|\theta_{slow} = 0)$ since only L3 is involved. If the new slow variables are modified, the update of the original slow variables $\Delta\theta_{slow}$ would be a sample from $g(\theta_{slow})$ due to L1. Updating the new slow variables also modifies θ_{fast} due to L2, so $\Delta\theta_{fast} = C^\top A^{-1}(\theta^{(s)}_{slow})$, being $\theta^{(s)}_{slow}$ the sample obtained previously. Figure 2.3 shows these three movements. The slow updates would involve the steps 1 and 2, whereas the fast update would correspond to the third step.

2.3.3 Ensemble MCMC

Neal (2011) proposed an alternative way to exploit the fast-slow feature. Instead of performing extra updates of the fast variables, he suggested to run a MCMC method in an ensemble of K states. This procedure could be useful when the computational cost needed to update the K ensembles is smaller than K times the cost necessary to update a single θ. The idea is having an ensemble $(\theta^{(1)}, .., \theta^{(k)})$ where all the elements $\theta^{(i)}$ of the ensemble share all the slow parameters but not the fast ones. We can then run a MCMC in the ensemble and accept or reject the movement of the whole ensemble depending on all $f(\theta^{(i)})$. Since all $\theta^{(i)}$ share the same slow parameters, evaluating the function $f(\theta^{(i)})$ for the different elements of the ensemble is computationally cheap.

We will combine this idea with performing extra updates of the fast variables to exploit the fast-slow feature in the method proposed.

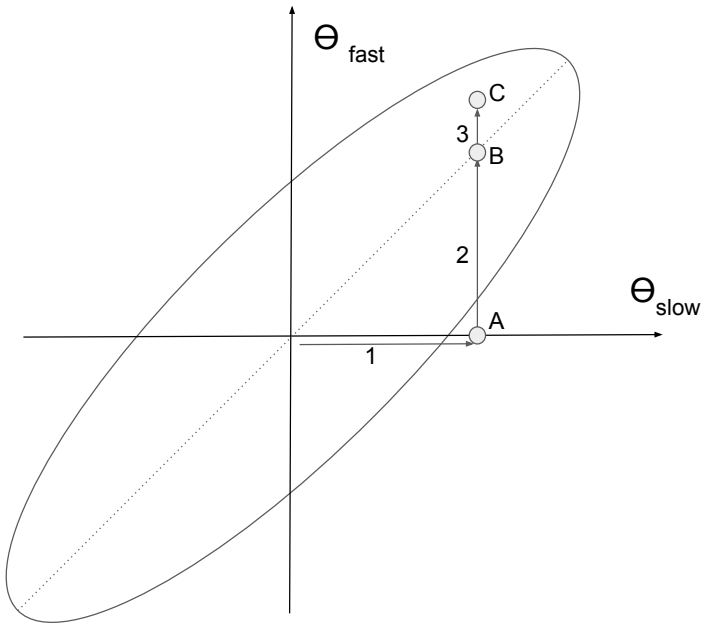


Figure 2.3: Process of sampling from $g(\theta_{slow}, \theta_{fast})$ given the Cholesky decomposition of its covariance matrix Σ. It could be divided into sample sample from $g(\theta_{slow})$; set θ_{fast} to $C^{\top}A^{-1}(\theta_{slow})$ and add to θ_{fast} a sample from $g(\theta_{fast}|\theta_{slow} = 0)$.

2.4 Cosmological and nuisance parameters

In the previous methods we described how to split the parameters into slow and fast parameters to explore the target distribution $f(\theta) = f(\theta_{slow}, \theta_{fast})$ efficiently. In cosmology the parameters not only could be divided into slow and fast but also into cosmological and nuisance. The cosmological parameters are the ones we want to explore, whereas the nuisance are not of interest. Cosmological parameters are usually slow and nuisance parameters are usually fast, although this is not always the case. Some nuisance parameters are related to the noise level or the calibration of some sensors (Lewis, 2013).

The objective is to sample from the probability distribution $f(\theta_{cosmo})$. In some cases the nuisance parameters can be analytically marginalized

$$f(\theta_{cosmo}) = \int f(\theta_{cosmo}, \theta_{nuisance}) d\theta_{nuisance} \tag{2.9}$$

Unfortunately, this is usually not possible. In this project we will consider the cases where we cannot marginalize analytically the parameters that are not of interest. The standard method to explore $f(\theta_{cosmo})$ is to run a Markov chain on the joint distribution

$f(\theta_{cosmo}, \theta_{nuisance})$ and discard the values of $\theta_{nuisance}$. More advanced methods have been developed to more efficiently run a Markov chain on the joint distribution when we want to explore a subset of the parameters of θ. We will refer to marginalize some parameters when we run a Markov chain on $(\theta_{cosmo}, \theta_{nuisance})$ and the acceptance of a new proposed θ only depends on a subset of variables $f(\theta_{cosmo})$. In these cases we will be ideally sampling from the marginal distribution $f(\theta_{cosmo})$.

In this Section we will first present a standard Metropolis-Hastings method intended to efficiently sample from a multivariate Gaussian when we want to explore a subset of variables θ_{cosmo}. Then we will present the method Dragging fast variables by Neal (2005). It describes a method to approximately marginalize over some fast variables and sample from $f(\theta_{slow})$. Finally, we will discuss the differences and motivate the proposed method.

2.4.1 Metropolis-Hastings with nuisance parameters

In multivariate Gaussians we can sample from the marginalized distribution $f(\theta_{cosmo})$ by running a Markov chain on $(\theta_{cosmo}, \theta_{nuisance})$ with the right proposal distribution $q()$. The target distribution $f(\theta_{cosmo}, \theta_{nuisance})$ could be decomposed using the chain rule in

$$f(\theta_{cosmo}, \theta_{nuisance}) = f(\theta_{cosmo}) f(\theta_{nuisance}|\theta_{cosmo}) \tag{2.10}$$

We can propose a value of θ_{cosmo} and set $\theta_{nuisance}$ to a sample from $f(\theta_{nuisance}|\theta_{cosmo})$ for the proposed θ_{cosmo} such as:

$$q(\theta'_{cosmo}, \theta'_{nuisance}; \theta_{cosmo}, \theta_{nuisance}) = q(\theta'_{cosmo}; \theta_{cosmo}) f(\theta'_{nuisance}|\theta'_{cosmo}) \tag{2.11}$$

Substituting 2.10 and 2.11 into the Metropolis-Hastings acceptance ratio we obtain

$$\begin{aligned} a &= \min\left(1, \frac{f(\theta'_{cosmo}) f(\theta'_{nuisance}|\theta'_{cosmo})}{f(\theta_{cosmo}) f(\theta_{nuisance}|\theta_{cosmo})} \frac{q(\theta_{cosmo}; \theta'_{cosmo}) f(\theta_{nuisance}|\theta_{cosmo})}{q(\theta'_{cosmo}; \theta_{cosmo}) f(\theta'_{nuisance}|\theta'_{cosmo})}\right) \\ &= \min\left(1, \frac{f(\theta'_{cosmo})}{f(\theta_{cosmo})} \frac{q(\theta_{cosmo}; \theta'_{cosmo})}{q(\theta'_{cosmo}; \theta_{cosmo})}\right) \end{aligned} \tag{2.12}$$

This is the acceptance ratio of the standard Metropolis-Hastings algorithm. Therefore, with the proposal distribution $q()$ from equation 2.11 we can sample from the marginalized distribution $f(\theta_{cosmo})$ in multivariate Gaussian distributions.

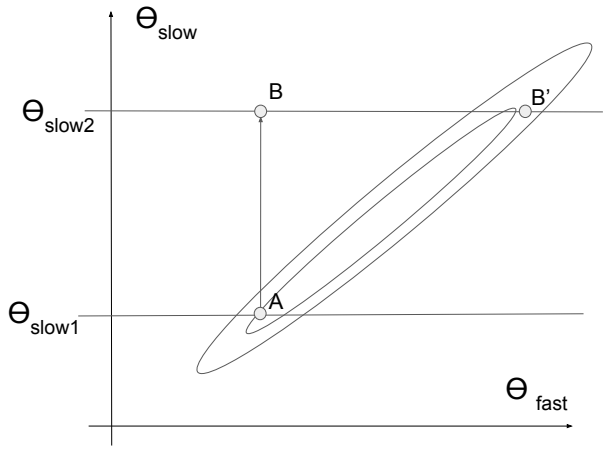


Figure 2.4: $A \rightarrow B'$ shows a much more likely proposal to be accepted than $A \rightarrow B$. Getting B' from B requires a cheap MCMC chain since only exploring the fast space is required (Neal, 2005).

2.4.2 Dragging fast variables

Neal (2005) devised a general scheme to make larger proposals when using Metropolis-Hastings by dragging the fast parameters with each slow proposal. It is shown in Figure 2.4 that for a large proposal $A \rightarrow B$ in the slow parameter space, the likelihood of accepting this move is very low. The method introduced by Neal works by making a proposal in the slow parameter space and running a MCMC chain in the fast parameter space. Then, by interpolating the two probability distributions $f(\theta_{fast}|\theta_{slow1})$ and $f(\theta_{fast}|\theta_{slow2})$, the fast parameters are dragged along the degeneracy direction. We can then take samples in the fast direction of the new function that interpolates between the two $f(\theta_{fast}|\theta_{slow})$. This new chain is likely to end at a position where the value of $f(\theta)$ is much larger. The new proposal is then $A \rightarrow B'$ that is much more likely to be accepted than $A \rightarrow B$. If the number of interpolations is big enough, the probability of accepting the whole move is equivalent of sampling from the marginalized distribution $f(\theta_{slow})$.

The method proposed in this project also proposes to marginalize over some of the variables. However it has two advantages with respect to the Dragging fast variables method: it allows to marginalize also slow variables, and it enables to parallelize some computation in case of marginalizing fast variables.

2.4.3 Discussion

We have seen in Section 2.4.1 a method that is able to sample from the marginalized distribution $f(\theta_{cosmo})$ in multivariate Gaussians. It has achieved that by approximating $f(\theta_{cosmo}) = \int f(\theta_{cosmo}, \theta_{nuisance}) d\theta_{nuisance}$ for a given setting of θ_{cosmo}. This method is an example of Pseudo-Marginal MCMC. This idea will be presented in more detail in the next Chapter.

The slow updates of the Fast-slow decorrelation method are also samples from the fully fast-marginalized probability distribution $f(\theta_{slow})$ if $f(\theta_{slow}, \theta_{fast})$ is a multivariate Gaussian. The slow updates set θ_{fast} to the same relative position for the new θ'_{slow} that has for the previous θ_{slow}. The marginal and conditional distribution are the same in fully orthonormalized parameter spaces (Lewis, 2013).

Both methods are sampling from the marginalized distributions because the target distribution $f(\theta)$ is a multivariate Gaussian. Unfortunately, the probability distributions to be explored are usually not Gaussian and therefore we need other methods to marginalize the variables we are not interested in. In the next Chapter we will propose a Pseudo-Marginal method that allows us to marginalize over the variables that are not of interest, either slow or fast.

Chapter 3

Method

This chapter explains the method that we propose for exploring a probability distribution $f(\theta_1, \theta_2)$ when we only want to explore a subset of variables θ_1. We will divide the variables in θ_1 and θ_2 to keep it generic. In θ_2 could be contained fast variables, slow variables or both. Like the Metropolis-Hastings method described in Section 2.4.1, it marginalizes over the variables that are not interest by approximating the marginal distribution $f(\theta_1)$. First we will introduce the idea of Pseudo-Marginal MCMC (Andrieu and Roberts, 2009). This approach is used when we cannot evaluate the function $f(\theta)$ but we can sample from a noisy unbiased estimator $\hat{f}(\theta)$. Secondly, we will move to the Auxiliary Pseudo-Marginal (APM) framework (Murray and Graham, 2016). It introduces auxiliary variables u into the Markov chain allowing to use new MCMC updates. Then, we will explain how to obtain the unbiased estimator $\hat{f}(\theta)$. Finally we will present two approaches, one to marginalize fast variables and other to marginalize slow and fast variables that are not of interest.

3.1 Pseudo-Marginal MCMC

Pseudo-Marginal MCMC methods are used to explore probability distributions that we cannot evaluate its value $f(\theta)$ up to a constant but we can get an approximate of its value. This approximation is sometimes too big, sometimes too small and on average it is correct. This noisy function is denoted $\hat{f}(\theta)$ and it is an unbiased estimator of $f(\theta)$. The unbiased estimator could be obtained from methods such as standard importance sampling estimates, particle methods (Andrieu et al., 2010) or randomized series truncation (Girolami et al., 2013).

Andrieu and Roberts (2009) proposed to explore the target distribution $f(\theta)$ by running a Markov chain on the state pair $(\theta, \hat{f})$. If the estimator is deterministic for all θ, we can replace $\hat{f}(\theta)$ with $f(\theta)$ and we have the same acceptance rate that the Metropolis-Hastings algorithm. If the estimator is noisy, we need to store the value of the noisy estimator for the last accepted pair $(\theta, \hat{f})$. This is explicitly accomplished by introducing the unbiased estimator into the Markov Chain state. A Pseudo-Marginal update is described in Algorithm 1.

Algorithm 1 Pseudo marginal update (Andrieu and Roberts, 2009)

1: $\theta' \sim q(\cdot;\theta)$.
2: $\hat{f}' \sim \in (\cdot;\theta')$.
3: $a = \min\left(1, \frac{\hat{f}(\theta')q(\theta;\theta')}{\hat{f}(\theta)q(\theta';\theta)}\right)$
4: **if** Uniform $[0, 1] < a$ **then return**
5: $\quad (\theta', \hat{f}')$
6: **else return**
7: $\quad (\theta, \hat{f})$

This method could be useful if we have a joint probability distribution $f(\theta_1, \theta_2)$ and we want to explore θ_1 but not θ_2. Ideally we would like to sample from the marginalized probability distribution $f(\theta_1)$. We may be able to obtain an unbiased estimator $\hat{f}(\theta_1)$ that allows us to explore θ_1 more efficiently than sampling from the joint distribution $f(\theta_1, \theta_2)$ and discarding the values θ_2. If this is the case, Pseudo-Marginal MCMC would improve the exploration of θ_1.

3.2 Auxiliary Pseudo-Marginal

In order to remove the noise from the unbiased estimator $\hat{f}(\theta)$, Murray and Graham (2016) proposed to assume that the noisy behaviour comes from some random numbers u and include them into the Markov Chain. The random numbers u are drawn from the probability distribution $q(u)$. The unbiased estimator $\hat{f}(\theta;u)$ would then be deterministic. The MCMC chain is then run on the joint auxiliary target distribution

$$p(\theta, u) = \hat{f}(\theta;u)q(u)/C \tag{3.1}$$

If $\hat{f}(\theta;u)$ is an unbiased estimator when u is drawn from $q(u)$ then

$$\mathbb{E}_{q(u)}[\hat{f}(\theta;u)] = \int \hat{f}(\theta;u)q(u)\,du = f(\theta) \tag{3.2}$$

Therefore, if we sample from the joint auxiliary target distribution $p(\theta,u)$ and we discard the values of u, we are sampling from the target marginal distribution $p(\theta) = \int p(\theta,u)\,du = f(\theta)/C$.

If using the Metropolis-Hastings algorithm, the proposals for θ and u can be drawn independently such as

$$q(\theta',u';\theta,u) = q(\theta',\theta)q(u') \tag{3.3}$$

Substituting 3.1 and 3.3 into the acceptance ratio we obtain

$$a = \min\left(1, \frac{p(\theta';u')q(\theta,u;\theta',u')}{p(\theta;u)q(\theta',u';\theta,u)}\right) = \min\left(1, \frac{\hat{f}(\theta';u')q(\theta;\theta')}{\hat{f}(\theta;u)q(\theta';\theta)}\right) \tag{3.4}$$

This is the same probability of acceptance that Pseudo-Marginal MCMC in Algorithm 1. Unfortunately, Pseudo-marginal algorithms are prone to get stuck at points where $\hat{f}(\theta;u)$ is high compared with its surroundings (Murray and Graham, 2016).

Incorporating, the random numbers u into the Markov chain allows us to use MCMC updates that were not available before. We can use now any MCMC method to update separately the variables θ and u. Murray and Graham (2016) called this new scheme Auxiliary Pseudo-Marginal (APM) framework, which proposes updating the parameters u and θ such as

$$p(u|\theta) \propto \hat{f}(\theta;u)q(u) \tag{3.5}$$

$$p(\theta|u) \propto \hat{f}(\theta;u) \tag{3.6}$$

The naming scheme for the APM framework consists of attaching the MCMC updates for the random numbers u and θ respectively to APM. For example, when using Metropolis Independence (MI) proposals for updating u and Metropolis-Hastings (MH) for θ, the method would be called APM MI+MH. Using MI for updating u is what the Pseudo-Marginal approach suggests to do (Andrieu and Roberts, 2009). Making independent proposals for u could make that the Markov chain gets stuck at points where $\hat{f}(\theta;u)$ for a given u is much larger than for any other u. When the Markov chain stops to update some of the variables, we say that is suffering from sticking behaviour. Using any other MCMC method to update u, which takes into account the

previous value of u to propose a new one such as Metropolis-Hastings, would solve the problem.

3.3 Unbiased estimator

In this section we will explain how to obtain the unbiased estimator, that in our case comes from importance sampling estimates. If we have the joint distribution $f(\theta_1, \theta_2)$ and we want to marginalize over θ_2, we have:

$$f(\theta_1) = \int f(\theta_1, \theta_2)\, d\theta_2 \tag{3.7}$$

We can then multiply and divide by $r(\theta_2)$ such as

$$f(\theta_1) = \int \frac{f(\theta_1, \theta_2) r(\theta_2)}{r(\theta_2)}\, d\theta_2 \tag{3.8}$$

Defining $b(\theta_1; \theta_2) = f(\theta_1, \theta_2)/r(\theta_2)$, we obtain

$$f(\theta_1) = \int b(\theta_1; \theta_2) r(\theta_2)\, d\theta_2 \tag{3.9}$$

We can get an approximation of $f(\theta_1)$ by drawing samples of θ_2 from $r(\theta_2)$ and computing the average $b(\theta_1; \theta_2)$ obtained. This approximation is denoted $\hat{f}(\theta_1)$ and it is an unbiased estimator of $f(\theta_1)$

$$\hat{f}(\theta_1) = \frac{1}{S} \sum_{s=1}^{S} b(\theta_1; \theta_2^{(s)}) \quad , \quad \theta_2^{(s)} \sim r(\theta_2) \tag{3.10}$$

The unbiased estimator $\hat{f}(\theta_1)$ is expected to be more accurate as the number of S increases. The variance of $\hat{f}(\theta_1)$ is proportional to $1/S$.

We have motivated previously the use of an approximate $g(\theta_1, \theta_2)$ of the target distribution $f(\theta_1, \theta_2)$ to make more efficient proposals when using Metropolis-Hastings methods. We have also described how to obtain $g(\theta_1, \theta_2)$ by running a preliminary chain. In our method we will use a new probability distribution $h(\theta_1, \theta_2) = g(\theta_1, \theta_2) + \mathbb{E}(\theta_1, \theta_2)$. This is an approximation of the target distribution $f(\theta_1, \theta_2)$ like $g(\theta_1, \theta_2)$ but with the same mean. We want to define $r(\theta_2)$ in such a way that each value of $b(\theta_1; \theta_2^{(s)})$ approximates as good as possible $f(\theta_1)$. Using the chain rule we can rewrite $b(\theta_1; \theta_2)$ as

$$b(\theta_1; \theta_2) = \frac{f(\theta_1, \theta_2)}{r(\theta_2)} = \frac{f(\theta_1) f(\theta_2|\theta_1)}{r(\theta_2)} \tag{3.11}$$

In order to be as accurate as possible, we want our function $r(\theta_2)$ to be close to $f(\theta_2|\theta_1)$. We will then set $r(\theta_2)$ to $h(\theta_2|\theta_1)$. If $f(\theta_1,\theta_2)$ would be a multivariate Gaussian and $h(\theta_1,\theta_2)$ would perfectly match $f(\theta_1,\theta_2)$, we would obtain an exact estimation of $f(\theta_1)$. In this case a single sample from θ_2 would be enough. However, this is not usually the case and obtaining an unbiased estimator with low variance is complicated.

3.4 Alternative approaches

In this Section two variants of the method will be proposed. First, we will present a method to marginalize the fast parameters and sample from $f(\theta_{slow})$ and then we will introduce a small modification to also marginalize the slow but nuisance parameters and sample from $f(\theta_{cosmo})$.

3.4.1 Marginalize fast parameters

In this approach we will describe a method to marginalize the fast variables. In the unbiased estimator from Section 3.3, $\theta_1 = \theta_{slow}$ and $\theta_2 = \theta_{fast}$.

In order to generate a sample from $r(\theta_{fast})$, we need an array of n random numbers u, being n the number of variables contained in θ_{fast}. If using S samples to estimate $\hat{f}(\theta_{slow})$, we need S arrays of random numbers u. Every single u is under the distribution $q(u)$. In our case $q(u)$ is a independent multivariate Gaussian where each variable comes from $\mathcal{N}(0,1)$. The Algorithm that obtains the unbiased estimator $\hat{f}(\theta_{slow};u)$ is described in the Algorithm 2.

Algorithm 2 Unbiased estimator from importance sampling

1: **function** UNBIASED ESTIMATOR($\theta_{slow}, u, h(\theta_{slow},\theta_{fast})$)
2: $output \leftarrow 0.$
3: **for** `each sample` **do**
4: $\Sigma \leftarrow Cov[h(\theta_{fast}|\theta_{slow})]$
5: $L \leftarrow Chol(\Sigma)$
6: $\theta_{fast} \leftarrow Lu[sample] + \mathbb{E}_{h(\theta_{fast}|\theta_{slow})}(\theta_{slow})$
7: We evaluate $r(\theta_2)$
8: $output \leftarrow output + f(\theta_{slow},\theta_{fast})/r(\theta_{fast})$
9: **return** output/number samples

Drawing more than one sample S is a way of exploiting the different speed of the parameters. All these samples S have the same value for θ_{slow} and different values for θ_{fast}. This means that, in theory, the cost associated to evaluate the unbiased estimator $\hat{f}(\theta_{slow};u)$ does not increase significantly when increasing S. The variance of $\hat{f}(\theta_{slow};u)$ decreases with the number of samples S as described in Section 3.3, so that it gets closer to sample from the marginal distribution $f(\theta_{slow})$. Therefore, increasing the number of samples S is expected to be worthwhile.

Additionally, updating separately θ_{slow} and u allow to perform extra updates of u. The random numbers u determine the value of θ_{fast}. This is the standard method to exploit the different speed of the parameters proposed by Lewis and Bridle (2002). We will also perform extra updates of u.

3.4.2 Marginalize nuisance parameters

The goal of the project is to obtain independent samples from $f(\theta_{cosmo})$. We have presented in the previous Section a method to marginalize the fast parameters. However, not all the slow variables are cosmological, some of them are nuisance: $\theta_{slow} = (\theta_{cosmo}, \theta_{slow,nuisance})$. In this second approach, we want to marginalize all the nuisance parameters, so that $\theta_1 = \theta_{cosmo}$ and $\theta_2 = \theta_{nuisance} = (\theta_{slow,nuisance}, \theta_{fast})$.

We will use the same random numbers u, described in the previous Section, to set the values θ_{fast}. Additionally, we will create another array of random numbers $u2$ to determine the values of $\theta_{slow,nuisance}$. This array contains $n2$ random numbers, being $n2$ the number of variables in $\theta_{slow,nuisance}$. Every single $u2$ is under the distribution $q2(u2)$. In our case $q2(u2)$ is also a independent multivariate Gaussian where each variable comes from $\mathcal{N}(0,1)$. We have decided to only have one configuration of the $\theta_{slow,nuisance}$ because it is expensive to evaluate the function $f(\theta_{slow}, \theta_{fast})$ when any variable from θ_{slow} is modified. The Algorithm that obtains the unbiased estimator $\hat{f}(\theta_{cosmo};u,u2)$ is described in the Algorithm 3.

Algorithm 3 Unbiased estimator from importance sampling

function UNBIASED ESTIMATOR($\theta_{cosmo}, u, u2, h(\theta_{slow}, \theta_{fast})$)
    $output \leftarrow 0.$
    for `each sample` **do**
        $\Sigma \leftarrow Cov[h(\theta_{slow,nuisance}, \theta_{fast}|\theta_{cosmo})]$
        $L \leftarrow Chol(\Sigma)$
        $\theta_{nuisance} \leftarrow L(u2, u[sample]) + \mathbb{E}_{h(\theta_{slow,nuisance}, \theta_{fast}|\theta_{cosmo})}(\theta_{nuisance})$
        We evaluate $r(\theta_{nuisance})$
        $output \leftarrow output + f(\theta_{cosmo}, \theta_{nuisance})/r(\theta_{nuisance})$
    return output/number samples

Chapter 4

Evaluation

In this chapter we present the experiments undertaken to evaluate the proposed method and compare it with other existing methods. Some of them have free parameters such as the proposal distribution in the Metropolis-Hastings algorithm. These free parameters will be also explored.

First, we will use standard methods to explore a simple test distribution (Toy problem). Afterwards, these methods will be applied to a synthetic distribution, provided by Joe Zuntz, intended to be more representative of a real posterior distribution. This distribution is a much more high dimensional function where the parameters are correlated. The differences in performance amongst the methods will be measured and discussed. Finally, we will evaluate the two alternatives of the Pseudo-Marginal approach proposed and compare them with the existing methods.

4.1 Toy problem

The simple test distribution consists of a truncated 4-dimensional multivariate Gaussian. A truncated distribution is a probability distribution whose values have been bounded below, above or both. In our case, the values of the simple test distribution are bounded above and below. Regarding the speed of the variables, 2 of them are considered slow variables and the other 2 remaining are fast variables. We created the parameters of this test distribution accordingly to:

- The elements $A_{i,j}$ of the matrix A, giving the covariance matrix $\Sigma = AA^T$, are drawn from $A_{i,j} \sim \mathcal{N}(0, 1)$.
- The mean of each variable is drawn from $\mu_i \sim \mathcal{N}(0, 1)$.

- The truncations are $\theta_{i,max} \sim \mu_i + 8\ \text{Uniform}(0,\ 1)$
- and $\theta_{i,min} \sim \mu_i - 8\ \text{Uniform}(0,\ 1)$

In order to compare the different methods, we need to assign an idealized computational cost to each iteration of the MCMC. As specified by Joe Zuntz, the cost of evaluating the likelihood of the data in the real problem is 0.01 seconds if only fast variables are modified with respect to the previous evaluation of the function and 20 seconds if any of the slow variables is perturbed. To be consistent in our experiments with the real problem, any time the posterior distribution is evaluated one of these idealized costs is assumed.

4.1.1 Experiments

The three standard MCMC methods selected to be run in the test distribution are:

- Metropolis-Hastings (Hastings, 1970). This is a standard method that does not assume any prior knowledge about the probability distribution to be explored, and doesn't take into consideration the cost associated to evaluate the function when each parameter is perturbed.
- Extra update Metropolis (Lewis and Bridle, 2002). This method performs extra updates of the fast parameters along with less frequent updates of the slow parameters.
- Fast-slow decorrelation (Lewis, 2013). This method is intended to, by means of an estimate of the covariance matrix, decorrelate the parameters as much as possible but keeping the new fast parameters fast.

All methods above mentioned have a free parameter to be tuned: the proposal size. The proposal distribution is spherical for all methods. For the Fast-slow decorrelation method, the proposal distribution is spherical in the new decorrelated space, but the proposals in the original space are drawn from the estimated covariance matrix. Before any method is run, a preliminary chain is run to tune the proposal size for each update in such a way that the acceptance ratio of each of the update is located in the interval $(0.2, 0.3)$. The optimal acceptance ratio for high-dimensional problems is 0.234, although the performance does not decrease considerably in the interval

$(0.15, 0.4)$ (Gelman et al., 1996). For all methods considered in this project, this preliminary chain sets a proposal size successfully so that the main Markov chain have an acceptance ratio within the interval $(0.2, 0.3)$.

In order to estimate the covariance, the emcee package is used, (Foreman-Mackey et al., 2013). It lets you run multiple chains to explore a probability distribution. It has two parameters: the number of chains to run and the number of iterations these chains are run for. The number of chains was set to four times the number of parameters the probability distribution has. The number of iterations was used to obtain different estimated covariances in terms of accuracy. The more iterations these preliminary chains are run for, the more accurate the estimate of the covariance is expected to be.

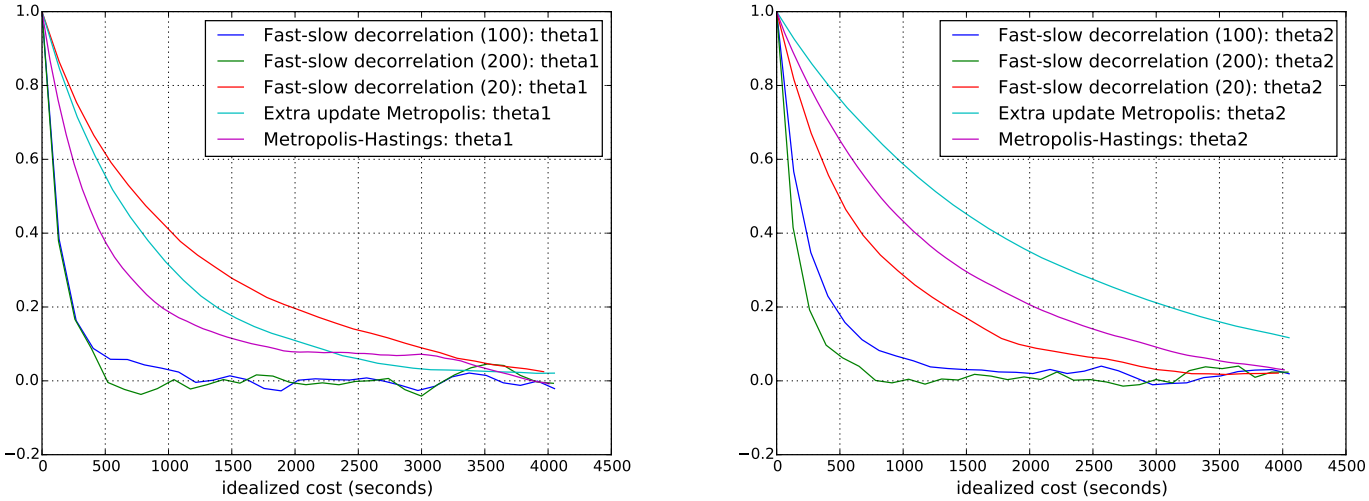
Extra update Metropolis and Fast-slow decorrelation methods propose to perform extra updates of the fast parameters. However, it is not specified how often these updates should be performed in comparison with the updates in the slow subspace. For the experiments undertaken in this project, it was decided to spend the same computational cost in both updates. Since the relative cost of both sets of parameters is 2000, on average 1 out of 2001 updates perturbs the slow parameters and the other ones perturb the fast parameters.

In figure 4.1 the autocorrelations obtained against the idealized cost are plotted for the three methods. For the Fast-slow decorrelation method, the performance with three different covariance matrices is evaluated. The more iterations the preliminary chains to estimate the covariance are run for, the more accurate these covariance matrices are. The number of iterations are: 20, 100 and 200.

We can conclude with these experiments that:

- The method that gives the best performance is the Fast-slow decorrelation. The more iterations the preliminary chain is run for the better performance is obtained.

- The Extra update Metropolis method was expected to work better than the Metropolis-Hastings method as it was intended to exploit the relative speed of the parameters. As pointed out by Lewis (2013), these extra updates of the fast parameters explore better the conditional distribution for each configuration of the slow parameters. He also said that this method may not work well when both fast and slow subspaces are not strongly correlated. In our case the correlations may not be important enough and having spent the same computational cost in the fast subspace (performing a huge number of extra updates) than in the slow subspace



(a) Autocorrelation of θ_1.

(b) Autocorrelation of θ_2.

Figure 4.1: Autocorrelation of the two slow parameters with respect to the idealized computational cost for the three methods proposed. For the Fast-slow decorrelation method, the performance is measured for different accuracies of the estimated covariance.

may simply not be worthwhile. However, the cost could be potentially reduced to the half by tuning the number of extra updates. Therefore, as long as it is not more than twice the cost of the Metropolis-Hastings method, it is considered a valid method susceptible to be optimized.

4.2 Real problem

We will now move to the synthetic distribution provided by Joe Zuntz. This probability distribution is intended to be representative of a real posterior distribution realizable by the CosmoSIS package (Zuntz et al., 2015). This probability distribution is a much more high dimensional function: 10 slow and 13 fast parameters. All fast parameters are nuisance parameters, therefore we are not interested in their values. Regarding the slow parameters, 6 out of them are cosmological, so that we want to explore the marginal distribution of these ones. As in the previous problem, the idealized cost of evaluating the likelihood of the data when fast parameters are modified is 0.01 seconds and when any slow parameters is perturbed 20 seconds.

We will now run different methods to explore this probability distribution. First, we

will start with the standard methods from the previous section and we will move then to the Pseudo Marginal approach proposed taking into consideration the performance of the different intermediate methods evaluated.

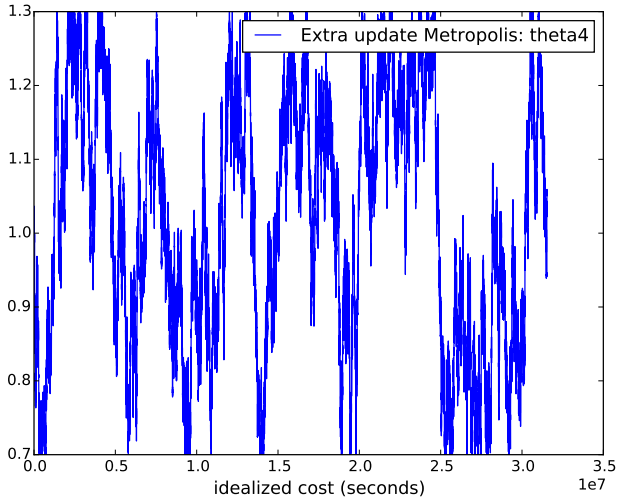
4.2.1 Standard methods

In this first setup, the three standard methods run in the toy problem will be run in the synthetic probability distribution: Metropolis-Hastings (Hastings, 1970), extra update Metropolis (Lewis and Bridle, 2002) and Fast-slow decorrelation (Lewis, 2013).

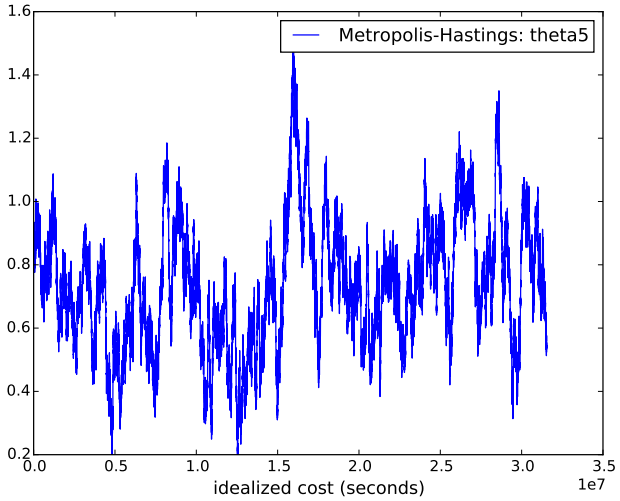
The Metropolis-Hastings and extra update Metropolis methods were not able to converge within an idealized cost of one year. A function that evaluates whether all cosmological parameters have mixed properly was run periodically inside the chain. This function goes through the already obtained samples of the chain and calculates their correlation. If the autocorrelation from a delay onwards is below a threshold it decides that the chain has mixed properly. In order to do that, the autocorrelation has to be smaller than a threshold for all the cosmological parameters. In our case, being len the length of the Markov chain, the autocorrelation for a delay within the interval $(0.25len, 0.5len)$ has to be smaller than 0.1 to say that the Markov chain has mixed.

Figure 4.2 shows the trace of two variables after an idealized cost of one year for the Metropolis-Hastings and extra update Metropolis methods. In order for a MCMC chain to mix properly, all the variables have to fluctuate covering all the domain of the variable. If this would be the case, the values would not be autocorrelated in time and would not depend on the previous values. As long as the values of both parameters strongly depend on the previous ones, we can say that these MCMC chains are not mixing properly for this number of iterations.

The reasons for not mixing are: higher correlations, more dimensions and different domains. Future experiments will evaluate the importance of the correlations between the variables compared to the standard deviation of each parameter, which determines the domain of the parameter. The number of dimensions always reduces the speed of convergence. These experiments also encourage the use of an estimate of the covariance matrix to speed up convergence.



(a) Trace of θ_4 for the extra update Metropolis method.



(b) Trace of θ_5 for the Fast-slow decorrelation method.

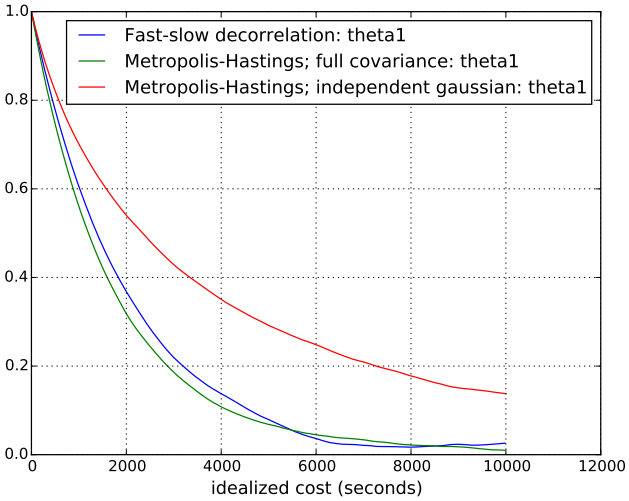
Figure 4.2: Trace of two different variables obtained with these standard methods after an idealized cost of one year.

4.2.2 Modification of the proposal distribution for Metropolis-Hastings methods

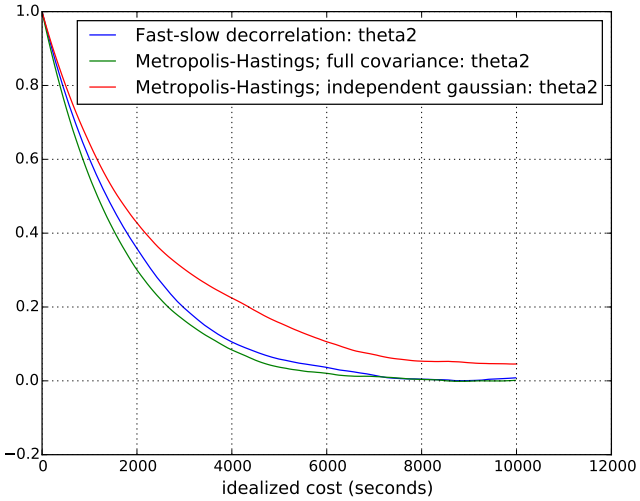
The performance of Metropolis-Hastings algorithms depends on the proposal chosen. If some of the parameters have bigger domains than others, we want to propose longer movements of these first parameters. In case some parameters are correlated, we want to move along the degeneracy directions (Kosowsky et al., 2002). When following these rules, we can make longer proposals while keeping a high acceptance ratio. This was explained in detail in Section 2.2. Fast-slow decorrelation method uses this principles to make better proposals. In this section we will adapt the proposals distribution for the Metropolis-Hastings and extra update Metropolis methods.

First, we will tune the proposal distribution of the Metropolis-Hastings method. We will consider two different proposals: scale the displacement of each variable with its standard deviation; and use the estimation of the covariance matrix as proposal. The first one is intended to exploit the different domains of the parameters and the second one also takes into consideration the correlation of the variables. The objective of this comparison is to determine to what extent each factor influences the performance.

From now on, once the importance of the accuracy of the estimated covariance was evaluated, we will just use one covariance matrix with a fixed accuracy. In order to be fair while comparing different methods, a preliminary chain was run beforehand and



(a) Autocorrelation of θ_1.



(b) Autocorrelation of θ_2.

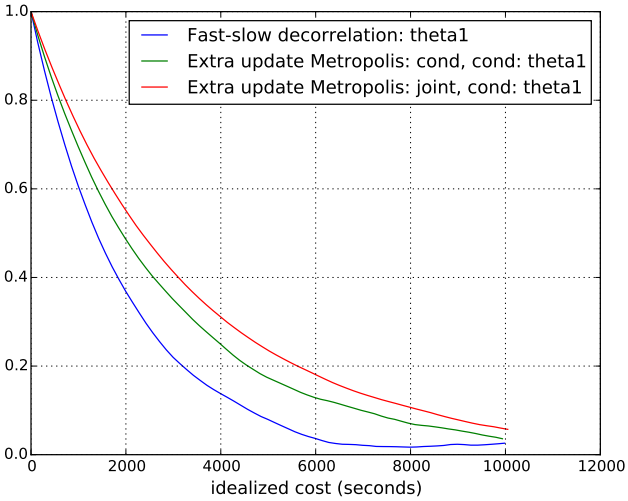
Figure 4.3: Autocorrelation of the two parameters that are more correlated with the Fast-slow decorrelation and the Metropolis-Hastings methods. For the last one, two different approaches are evaluated for the proposal: an estimate of the covariance and an independent proposal where the the step of each variable is scaled with its standard deviation.

the same covariance is used for all experiments.

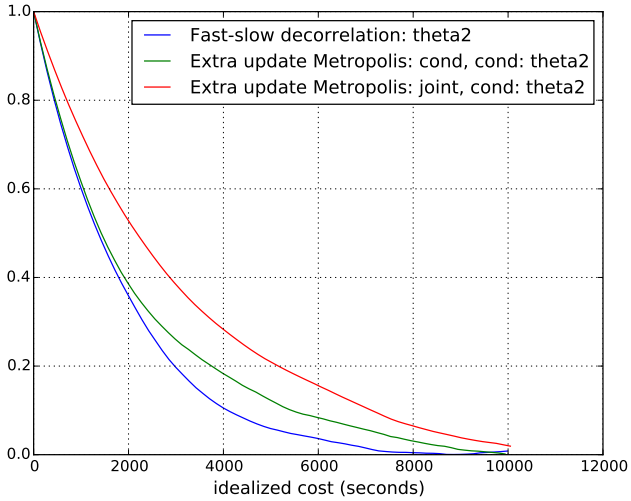
Figure 4.3 shows the performance of these two methods compared with the Fast-slow decorrelation that gave the best performance so far. The variables that have been plotted are the ones that were more autocorrelated and therefore the ones that determine how much computational cost is needed to obtain two independent samples from the probability distribution.

The Metropolis-Hastings methods that takes into account the correlation between parameters works better than the one that does not do it. Not only the standard deviation of each variable is important, but also the correlation between the different variables to be able to make more independent proposals. The Metropolis-Hastings performs roughly the same that the Fast-slow decorrelation method. As discussed before, any method that exploits the fast-slow feature is susceptible to be optimized by tuning the number of extra fast updates.

Secondly, we will use the estimate of the covariance to improve the proposal distribution of the extra update Metropolis method. All these approaches use Metropolis-Hastings for both updates where the proposal is drawn from a probability distribution



(a) Autocorrelation of θ_1.



(b) Autocorrelation of θ_2.

Figure 4.4: Autocorrelation of the two parameters that are more autocorrelated with the Fast-slow decorrelation and the extra update Metropolis method. Two different approaches are considered for Extra update Metropolis: draw the proposal from the conditional distributions (cond, cond) or from the joint distribution to update all variables and from the conditional to update the fast ones (joint, cond).

center at the current state. The three approaches considered use the following probability distributions for each update:

- Main update: $\Delta\theta_{slow} \sim g(\theta_{slow}|\theta_{fast})$; and fast update: $\Delta\theta_{fast} \sim g(\theta_{fast}|\theta_{slow})$.
- Main update: $\Delta\theta_{slow,fast} \sim g(\theta_{slow,fast})$; and fast update: $\Delta\theta_{fast} \sim g(\theta_{fast}|\theta_{slow})$.
- Main update: $\Delta\theta_{slow} \sim g(\theta_{slow})$ and $\Delta\theta_{fast}$; and fast update: $\Delta\theta_{fast} \sim g(\theta_{fast}|\theta_{slow})$.

The third approach corresponds to the Fast-slow decorrelation method. The $\Delta\theta_{fast}$ modifies the fast variables in such a way that the new fast variables θ_{fast} are located at the same position of the conditional distribution $g(\theta_{fast}|\theta_{slow})$ when modifying the value of θ_{slow}. Figure 4.4 shows the performance of the three methods.

We can conclude that:

- Fast-slow decorrelation is the method with the best performance.
- Using as proposal the joint distribution $g(\theta_{slow,fast})$ works considerably worse than the Fast-slow decorrelation method. This is because moving all variables at

once makes the step size, to get a similar acceptance ratio, smaller. As explained in Section 4.1.1, the step size for each method is tuned beforehand so that the acceptance ratio in the main chain is within the interval $(0.2, 0.3)$. A smaller step size makes the process of exploring the slow variables slower.

- The Fast-slow decorrelation works better than updating both parameters with a proposal from the conditional distribution. We can say that there exist a correlation between the fast and slow subspaces that needs to be taken into account.

4.2.3 Pseudo-Marginal first approach

In this section we will explore the first Pseudo-Marginal method proposed, intended to marginalize the fast parameters. Pseudo-Marginal methods are used to explore probability distributions that we cannot evaluate its value $f(\theta)$ up to a constant but we can get an approximate of the function $\hat{f}(\theta)$. We want to sample from the probability distribution $f(\theta_{slow})$ but we can not marginalize over the fast variables. The unbiased estimator $\hat{f}(\theta_{slow})$ could be seen as a noisy function that also depends on θ_{fast} (Andrieu and Roberts, 2009), or we can make the θ_{fast} dependent on some random numbers u which should be included in the MCMC. The unbiased estimator $\hat{f}(\theta_{slow}, u)$ would be now stochastic and we can run a MCMC chain on the pair (θ_{slow}, u) (Murray and Graham, 2016). The second approach allows to use any MCMC mechanism to update θ_{slow} and u independently.

The objective is to sample from the marginal distribution $f(\theta_{slow})$. The Pseudo-Marginal approach is useful when we have access to an unbiased estimator $\hat{f}(\theta_{slow}, u)$ that does not vary too much when modifying u. If this is the case, running a MCMC on the pair (θ_{slow}, u) could be a more efficient way to explore $f(\theta_{slow})$ than exploring the joint distribution $(\theta_{slow}, \theta_{fast})$ and discard the values of θ_{fast}. Therefore, the unbiased estimator $\hat{f}(\theta_{slow}, u)$ is intended to be a measure, as accurate as possible, of the marginal distribution of $f(\theta_{slow})$. In our case, the unbiased estimator comes from importance sampling, described in Algorithm 2.

In order to exploit the fast-slow feature we decided to perform additional updates of the fast variables (Lewis and Bridle, 2002) and create an ensemble of states sharing the same slow variables but not the fast ones (Neal, 2011). In order to compare with the other methods evaluated, we decided to spend the same idealized computational cost in both updates. Considering one cycle the updates involved on average between two updates of the slow variables, N the number of ensembles, C_1 the cost of modifying the

slow variables and C_2 the cost of modifying the fast variables, the costs in this period are:

- Cost A: C_1. This is the whole cost of the mean update if $N = 1$. In our problem, $C_1 = 20s$.
- Cost B: This is an extra cost of the mean update due to having more than one ensemble. Since all ensembles have the same slow variables, $B = (N-1)C_2$. In our case, $C_2 = 0.01s$.
- Cost C: This cost is associated to the fast updates. To spend the same computational cost in both updates: $C = A + B$. The number of extra updates of the fast variables per cycle is then $C_1/(NC_2) + B$.

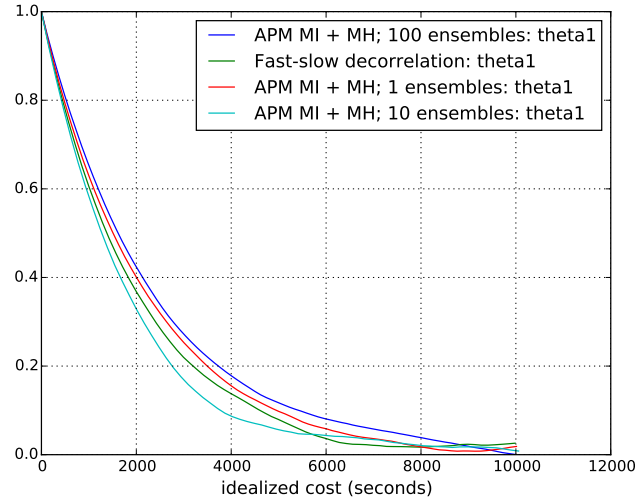
We will now compare the performance of the Pseudo Marginal approach with the Fast-slow decorrelation method. We will use the APM framework (Murray and Graham, 2016) and run a MCMC chain on the pair (θ_{slow}, u). This framework allows to use any MCMC update to update θ_{slow} and u according to equations 3.5 and 3.6. In the slow update, the proposal of the the Fast-slow decorrelation method is drawn from the estimated covariance, such as

$$\theta'_{slow} \sim \theta_{slow} + g(\theta_{slow}) \tag{4.1}$$

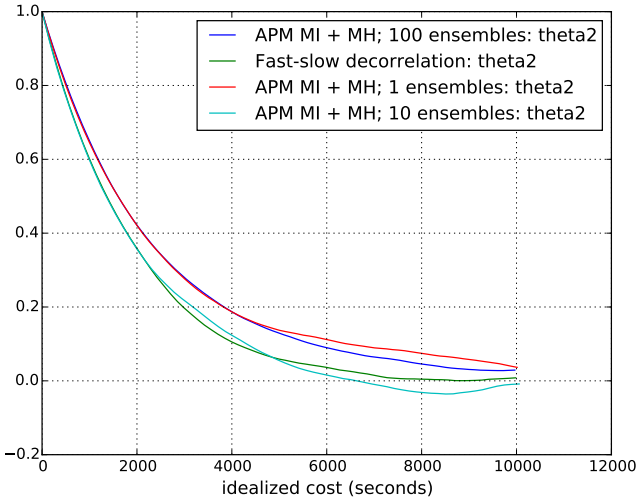
$$\theta'_{fast} = \theta_{fast} + \mathbb{E}_{g(\theta_{fast}|\theta'_{slow})}(\theta_{fast}) \tag{4.2}$$

In order to compare with the Fast-slow decorrelation method, the update selected for the variable θ_{slow} in the APM framework is the Metropolis-Hastings (MH) method with the marginal distribution $g(\theta_{slow})$ as proposal distribution. The values of θ_{fast} are also perturbed in the main update of the Fast-slow decorrelation method. In the Pseudo-Marginal approach, for a fixed value of the random numbers u, the unbiased estimator also modifies θ_{fast} depending on the current value of θ_{slow}. Regarding the update of the random numbers u, the Pseudo marginal literature suggests to perform Metropolis Independence (MI) proposals. In Figure 4.5 we compare the performance of the Pseudo-Marginal approach (APM MI+MH) with the Fast-slow decorrelation method for different number of ensembles.

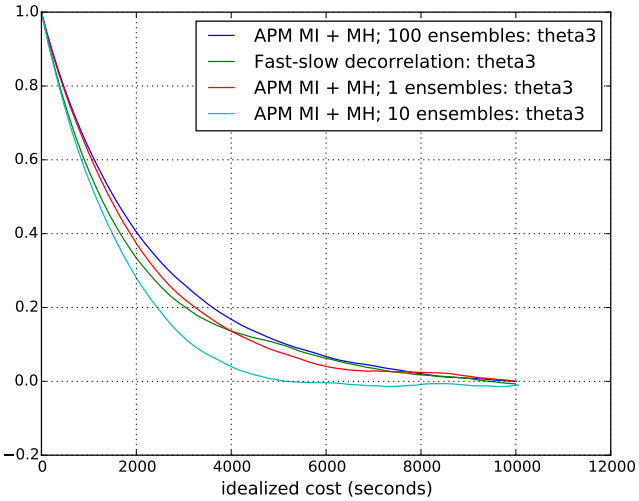
Using Metropolis Independence (MI) proposals for updating the random numbers u could trigger a sticking behaviour (Murray and Graham, 2016). This is likely to happen when the unbiased estimator $\hat{f}(\theta_{slow}|u)$ has high variance. When using MI for updating



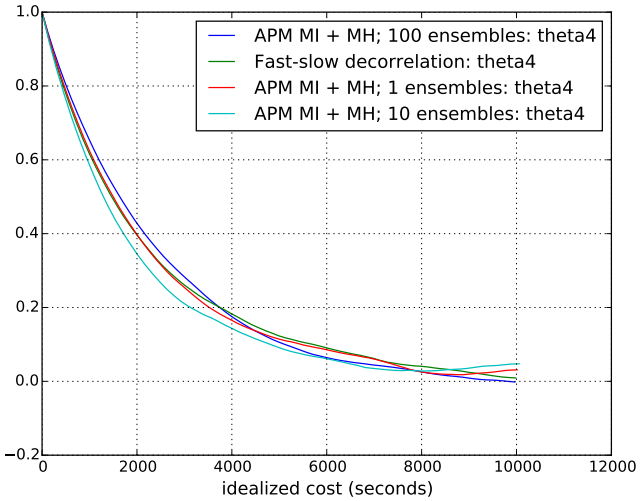
(a) Autocorrelation of θ_1.



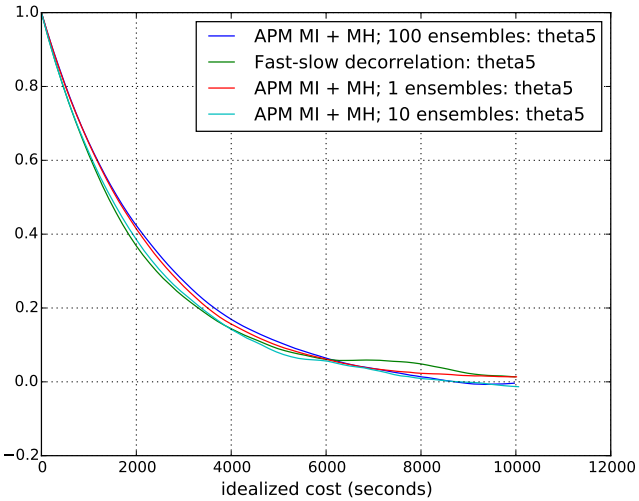
(b) Autocorrelation of θ_2.



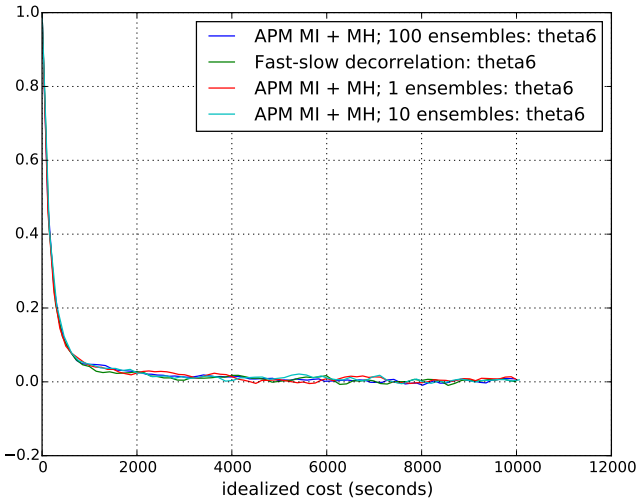
(c) Autocorrelation of θ_3.



(d) Autocorrelation of θ_4.



(e) Autocorrelation of θ_5.

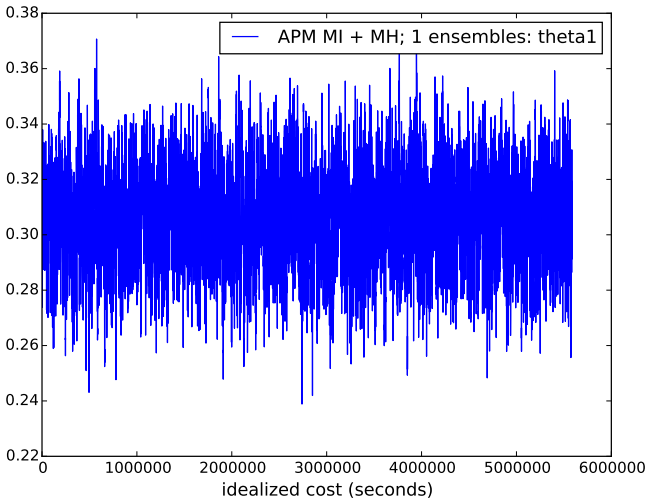


(f) Autocorrelation of θ_6.

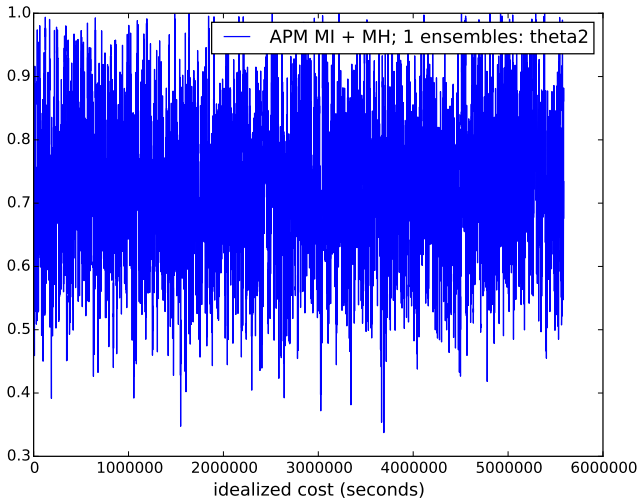
Figure 4.5: Autocorrelation of all cosmological variables for the Pseudo-marginal (APM MI+MH) and the Fast-slow decorrelation methods.

Number of ensembles	Acceptance ratio
1	0.727
10	0.901
100	0.966

Table 4.1: Acceptance ratio when perturbing the random numbers u with Metropolis Independence (MI) proposals in the APM MI+MH framework for different number of ensembles.



(a) Trace of θ_1 showing the lack of sticking behaviour.



(b) Trace of θ_2 showing the lack of sticking behaviour.

Figure 4.6: Trace of two different variables obtained with the method APM MI+MH and one ensemble.

the random numbers u, high variance unbiased estimators have low acceptance ratios and low variance estimators have acceptance ratios close to one. Table 4.1 shows the acceptance ratio for different number of ensembles.

The higher number of ensembles, the lower variance the unbiased estimator is expected to be and therefore higher acceptance ratio. It is shown in Table 4.1 that this holds in our experiment. However, a sticking behaviour is still quite possible. It could be the case that an u with a high value of the unbiased estimator is proposed and it takes a lot of time to accept a different u. Since the unbiased estimator has higher variance with one ensemble, this configuration is more susceptible of sticking behaviour. In Figure 4.6 we show the trace of two variables with the method APM MI+MH (one ensemble), showing that this behaviour is not present.

Once we have checked that the perturbations of u are working properly and the same update is performed for θ_{slow} in our APM MI+MH than in the Fast-slow decorrelation method, we will analyse the differences shown in Figure 4.5:

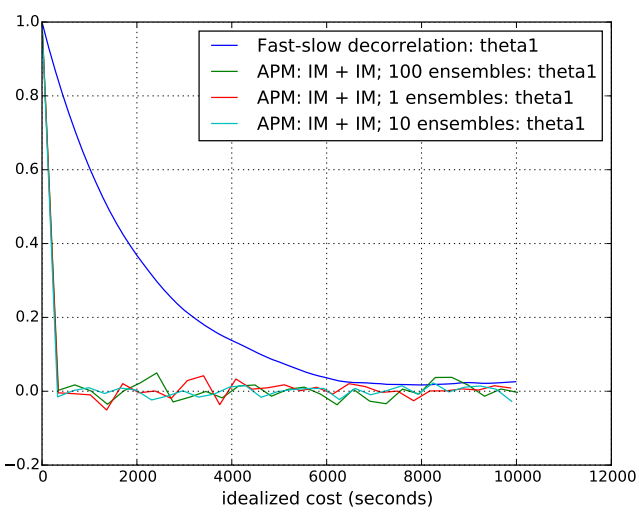
- APM MI+MH, with 1 ensemble, performs roughly the same than the Fast-slow decorrelation method. When having one ensemble in the Pseudo-Marginal approach this is equivalent to the approach described in Section 2.4.1. We have described in Section 2.4.3 that they are two different ways of sampling from the fast-marginalized distribution. They would be sampling from the marginal distribution $f(\theta_{slow})$ if the target distribution $f(\theta_{slow}, \theta_{fast})$ would be a multivariate Gaussian. The target distribution is a truncated multivariate Gaussian. Although, for the given target distribution both methods give the same performance, it is expected that for a more complex target distribution the Pseudo-Marginal approach works better.

- APM MI+MH, with 10 ensembles, performs better than the Fast-slow decorrelation method and APM MI+MH with 1 ensemble. The decrease of variance in the unbiased estimator $\hat{f}(\theta_{slow}, u)$ compensates the extra cost associated to evaluate this 9 additional states. This robustness in the unbiased estimator is theoretically expected and could be deduced from the acceptance ratios shown in Table 4.1.

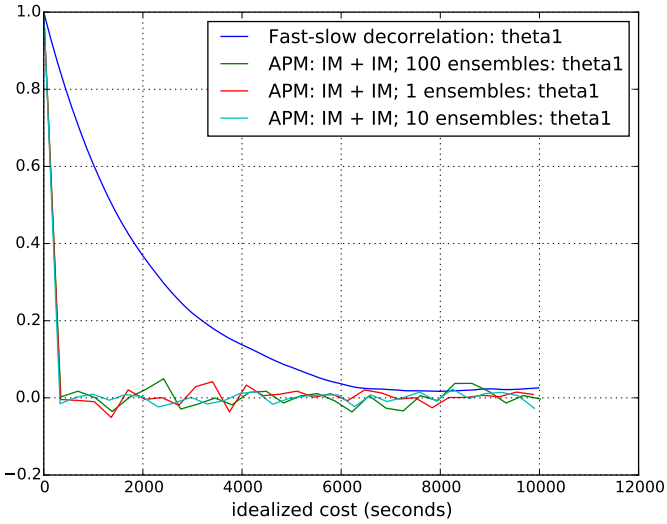
- When the number of ensembles is much larger, the cost of evaluating these additional ensembles in the main updates increases the idealized cost involved in one cycle. Therefore, the updates of the variables of interest occur less often, making that the decrease in variance of the unbiased estimator is not worthwhile.

- The computation of C and B could be potentially paralellized when the computation takes place in machines with multiple cores. The evaluation of the function $f(\theta_{slow}, \theta_{fast})$ for the different states in the ensemble is independent to each other, so that could be done in parallel. For the case of 10 ensembles, if having enough cores, the cost could be reduced to $A + B/10 + C/10$ per cycle.

Finally, motivated by the good performance of MI when perturbing u, we decided to use this Metropolis Independence proposals to update θ_{slow}. The results are shown in Figure 4.7. Regardless of the number of ensembles, the acceptance ratio for the updates of θ_{slow} was around 0.12. Assuming that a cycle $(A + B + C)$ has an idealized computational cost slightly larger than $40s$ and the acceptance ratio measured was



(a) Autocorrelation of θ_1.



(b) Autocorrelation of θ_2.

Figure 4.7: Autocorrelation of the two parameters that are more correlated with the Fast-slow decorrelation method and the APM MI+MI.

around 0.12, a completely independent sample from the target distribution is obtained each $320s$.

This update could be used in the Fast-slow decorrelation method as well. It does not have anything to do with the Pseudo-Marginal approach. However, it could be understood as a sign that the target distribution is not too complex and this justifies why the Pseudo-Marginal approach did not give a significant advantage with respect to the Fast-slow decorrelation method.

4.2.4 Pseudo-Marginal second approach

In this section we will evaluate the second variant of the method proposed. The first one was intended to marginalize the fast parameters and sample from $f(\theta_{slow})$, whereas this approach wants to sample from $f(\theta_{cosmo})$. Inside θ_{slow} there are cosmological parameters θ_{cosmo} and nuisance parameters $\theta_{slow,nuisance}$. In order to marginalize the second ones, we will make $\theta_{slow,nuisance}$ dependent on some additional random numbers $u2$, and include $u2$ into the unbiased estimator. The new unbiased estimator $\hat{f}(\theta_{cosmo}; u, u2)$, described in Algorithm 3, is intended to provide an accurate measurement of the marginalized distribution $f(\theta_{cosmo})$.

We will only have one setting of $\theta_{slow,nuisance}$ because it is expensive to evaluate

Number of ensembles	Acceptance ratio
1	0.547
10	0.623
100	0.632

Table 4.2: Acceptance ratio when perturbing the random numbers u and $u2$ with Metropolis Independence (MI) proposals in the APM2 MI+MH framework for different number of ensembles.

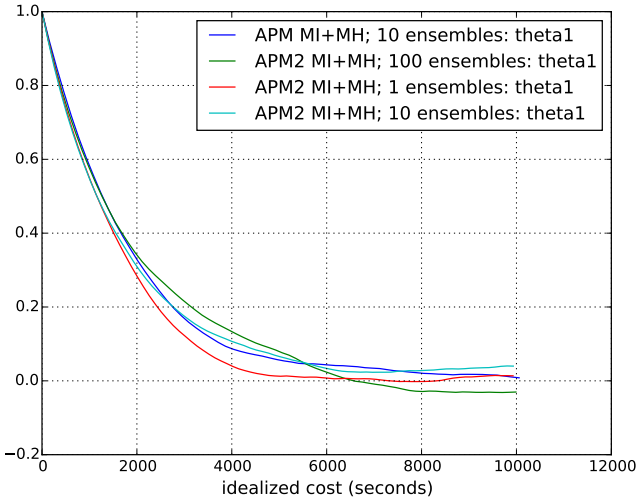
the function $f(\theta_{slow}, \theta_{fast})$ when any variable from θ_{slow} is modified. Similarly to the first approach, we will have many settings of θ_{fast} because it is cheap to evaluate $f(\theta_{slow}, \theta_{fast})$ when only fast variables are modified. Performing extra updates of the random numbers u is still possible to exploit the different speed of the parameters.

In order to compare with the other methods, we have decided to spend the same computational cost in both updates. We will follow the costs notation from previous section: A, B and C. The costs A and B are exactly the same, whereas in C there are now one slow update due to update $u2$ and many fast updates due to update u. The slow update is equal to the cost A and the fast update is equal to B. The new Pseudo-Marginal approach has been called APM2 and also uses MI to update the random numbers u and $u2$ and MH to update θ_{cosmo}. The acceptance rates for the variables u and $u2$ together are shown in Table 4.2. In Figure 4.8 we compare the performance of this new Pseudo-Marginal approach (APM2 MI+MH) with the best one obtained with the first variant: APM MI+MH with 10 ensembles.

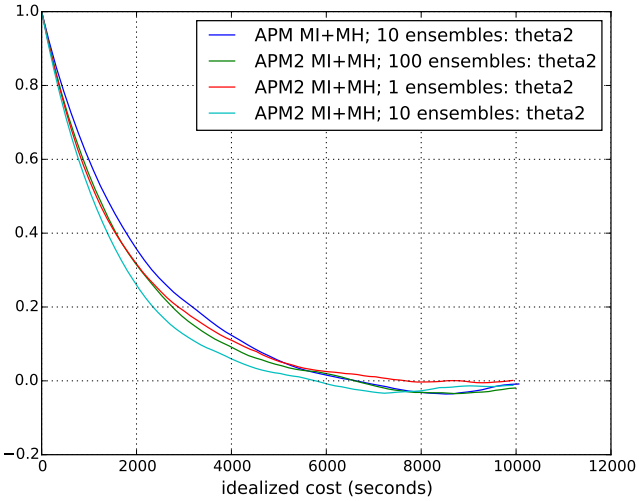
We can make the following conclusions:

- APM2 MI+MH works better than APM MI+MH. We can see in the Tables 4.1 and 4.2 that the random numbers are being updated properly, so the slow updates of APM2 perform better. Marginalizing $\theta_{slow,nuisance}$ is worthwhile. It performs better than running a standard Markov chain in $(\theta_{cosmo}, \theta_{slow,nuisance})$ and discard the values of $\theta_{slow,nuisance}$.

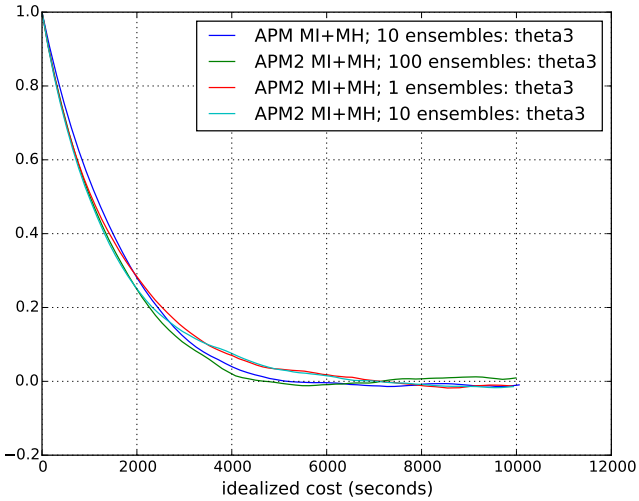
- The cost involved in C for the APM method could be parallelized in machines with multiple cores. This is not possible with the APM2 method because it contains the update of one slow variable. Therefore, this method does not have this potential gain.



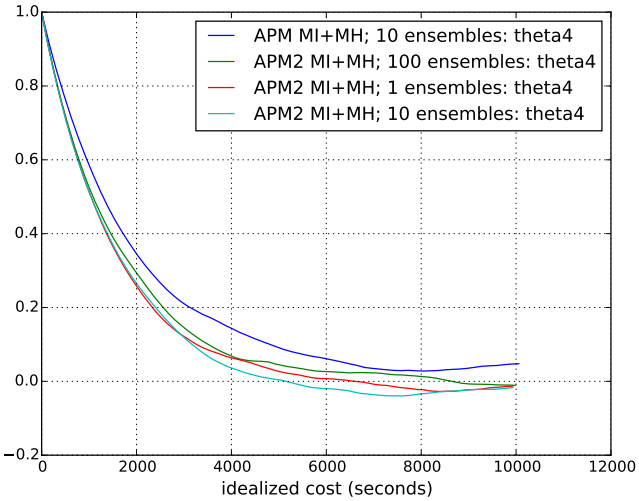
(a) Autocorrelation of θ_1.



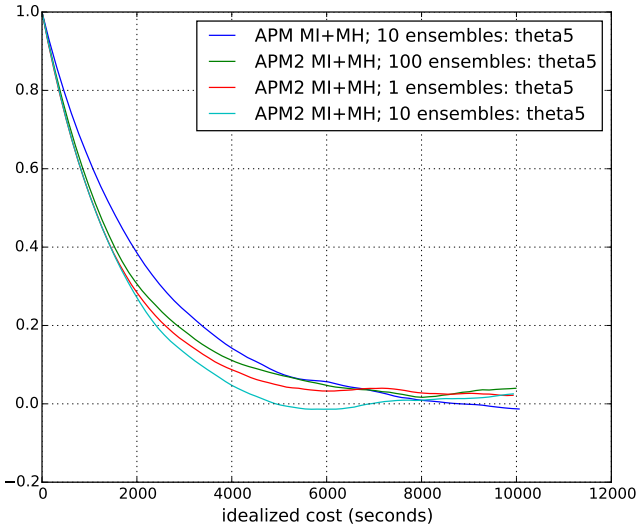
(b) Autocorrelation of θ_2.



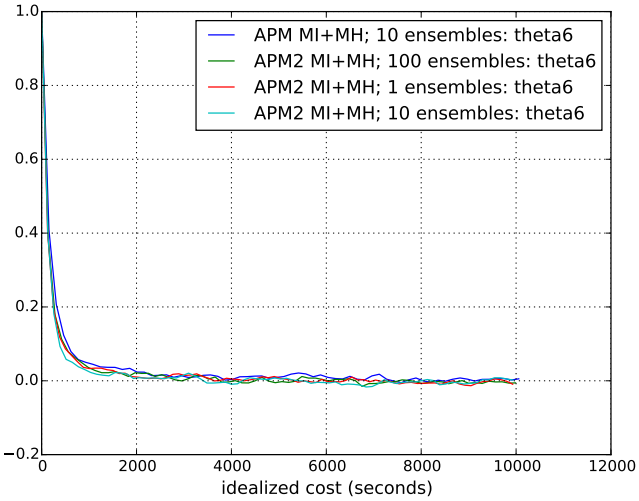
(c) Autocorrelation of θ_3.



(d) Autocorrelation of θ_4.



(e) Autocorrelation of θ_5.



(f) Autocorrelation of θ_6.

Figure 4.8: Autocorrelation of all cosmological variables for the second Pseudo-marginal variant (APM2 MI+MH) and the first one (APM MI+MH).

Chapter 5

Conclusion

In this project we have implemented different methods from the literature to explore cosmological models. We have also proposed to adapt a Pseudo-Marginal approach to the cosmological problem. All of them were tested in a synthetic probability distribution intended to be representative of a model realizable by the CosmoSIS package. The conclusions are:

- The Fast-slow decorrelation method is the one that gives the best performance from the existing methods. The Pseudo-Marginal approach proposed, with one ensemble, performs as good as the Fast-slow decorrelation method. When the number of ensembles of the Pseudo-Marginal approach is increased to 10, the performance is slightly higher. It shows that the decreased in variance of the unbiased estimator is worthwhile in spite of the additional cost of these extra ensembles.

- The Fast-slow decorrelation would be sampling from the marginalized distribution over the fast variables in the main update if the target distribution would be a multivariate Gaussian. The truncations for the model used are not too strong so that, the main update is sampling from a probability distribution close to the fast-marginalized probability distribution. It is expected that for a more complex function the Pseudo-Marginal will still obtain a reasonable low variance estimator of the marginal distribution whereas the Fast-slow decorrelation method could suffer a decrease in performance.

- Metropolis Independence (MI) proposals works much better than Metropolis-Hastings (MH) updates for this model and the estimated covariance used. This is also an indication that the model may not be that complicated.

- We have devised a general scheme to marginalize fast and slow variables that are not of interest. The proposed method, in the cosmological model used, works slightly better than the considered existing methods. It is expected to have a potential increase in performance for more complicated models, although this needs to be tested. Furthermore, it allows to parallelize the computation of the likelihood, which would allow larger gains in machines of multiple cores.

5.1 Further Work

Two ideas are proposed for further work:

- In order to affirm that the method works actually better than the Fast-slow decorrelation method, we need to evaluate it in different models. We could then test the hypothesis that the Pseudo-Marginal approach would adapt better to more complicated models. It will also be useful to evaluate how robust both methods are when the accuracy of the estimated covariance is decreased.

- Two variants of the Pseudo-Marginal approach have been proposed. One marginalizes the fast variables and the other one marginalizes the fast variables and some slow variables that are not of interest. In this second variant, moving slow variables to the second update have some downsides that have been identified in Section 4.2.4. Furthermore, we have only used one setting of the slow variables marginalized $\theta_{slow,nuisance}$. Having one setting makes that it could be viewed as a Markov chain in $(\theta_{cosmo}, \theta_{slow,nuisance})$. We have described in Section 2.4.1 how to set the proposal distribution $q()$ to marginalize some variables. We present here the possibility to mix the benefits of both Pseudo-Marginal variants by using the unbiased estimator $\hat{f}(\theta_{slow}, u)$ from the first variant and marginalize $\theta_{slow,nuisance}$ when updating $\theta_{slow} = (\theta_{cosmo}, \theta_{slow,nuisance})$ with the right proposal distribution as described in Section 2.4.1.

  This approach could also have some downsides. First, if we draw an independent sample of $\theta_{slow,nuisance}$ from the conditional distribution $h(\theta_{slow,nuisance}|\theta_{slow})$ once θ_{cosmo} is sampled, we may get a small acceptance ratio; if we perturb $\theta_{slow,nuisance}$ under $h(\theta_{slow,nuisance}|\theta_{slow})$ and decide to accept or reject the update we will have two degrees of freedom so the step size would not be easy to tune.

Bibliography

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, pages 697–725.

Foreman-Mackey, D., Hogg, D. W., Lang, D., and Goodman, J. (2013). emcee: the MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306.

Gelman, A., Roberts, G. O., Gilks, W. R., et al. (1996). Efficient Metropolis jumping rules. *Bayesian statistics*, 5(599-608):42.

Girolami, M., Lyne, A.-M., Strathmann, H., Simpson, D., and Atchade, Y. (2013). Playing russian roulette with intractable likelihoods. Technical report, Technical Report.

Haario, H., Saksman, E., and Tamminen, J. (1999). Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, 14(3):375–396.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Kosowsky, A., Milosavljevic, M., and Jimenez, R. (2002). Efficient cosmological parameter estimation from microwave background anisotropies. *Physical Review D*, 66(6):063007.

Lewis, A. (2013). Efficient sampling of fast and slow cosmological parameters. *Physical Review D*, 87(10):103529.

Lewis, A. and Bridle, S. (2002). Cosmological parameters from CMB and other data: A Monte Carlo approach. *Physical Review D*, 66(10):103511.

Murray, I. and Graham, M. (2016). Pseudo-marginal slice sampling. In *Artificial Intelligence and Statistics*, pages 911–919.

Neal, R. M. (2005). Taking bigger Metropolis steps by dragging fast variables. *arXiv preprint math/0502099*.

Neal, R. M. (2011). MCMC using ensembles of states for problems with fast and slow variables such as Gaussian process regression. *arXiv preprint arXiv:1101.0387*.

Zuntz, J., Paterno, M., Jennings, E., Rudd, D., Manzotti, A., Dodelson, S., Bridle, S., Sehrish, S., and Kowalkowski, J. (2015). CosmoSIS: modular cosmological parameter estimation. *Astronomy and Computing*, 12:45–59.