NeuroImage 112 (2015) 375–381

Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg

Technical Note

# Gradient-free MCMC methods for dynamic causal modelling

Biswa Sengupta *, Karl J. Friston, Will D. Penny

*Wellcome Trust Centre for Neuroimaging, Institute of Neurology, University College London, 12 Queen Square, London WC1N 3BG, UK*

ARTICLE INFO

*Article history:*
Received 3 October 2014
Accepted 6 March 2015
Available online 14 March 2015

ABSTRACT

In this technical note we compare the performance of four gradient-free MCMC samplers (random walk Metropolis sampling, slice-sampling, adaptive MCMC sampling and population-based MCMC sampling with tempering) in terms of the number of independent samples they can produce per unit computational time. For the Bayesian inversion of a single-node neural mass model, both adaptive and population-based samplers are more efficient compared with random walk Metropolis sampler or slice-sampling; yet adaptive MCMC sampling is more promising in terms of compute time. Slice-sampling yields the highest number of independent samples from the target density — albeit at almost 1000% increase in computational time, in comparison to the most efficient algorithm (i.e., the adaptive MCMC sampler).



## Introduction

This technical note reports comparative evaluations of common gradient-free sampling schemes that can be used for Bayesian inference in dynamic causal modelling. It is the first of a series of technical reports that hopes to provide a comprehensive survey of the various sampling schemes available — both gradient-free, and (first and second order) gradient-based schemes. These schemes provide a gold standard against which the performance of fixed form (e.g., variational) approximate Bayesian inference can be compared. Furthermore, with advances in computer science, the computational costs usually associated with sampling schemes may be sufficiently reduced to allow their routine use in applications like dynamic causal modelling.

Dynamical Causal Models (DCMs) are used routinely in neuroimaging as generative models of neurophysiological signals (Friston et al., 2003). Inference on their parameters usually proceeds using a parameterised probability density and maximising the (variational free energy) evidence lower bound (Friston et al., 2007). Typically a Laplace approximation (Tierney and Kadane, 1986) is used for inference in DCMs because it does not require algebraically involved updates, unlike variational Bayes — and is guaranteed to converge as a result of the central limit theorem (Wang and Titterington, 2004). But such deterministic algorithms have their limitations. For example, they underestimate the variability in the posterior density; they get locked in local minima and are unable to approximate multi-modal posteriors (MacKay, 2002). Markov Chain Monte Carlo (MCMC) schemes are stochastic sampling algorithms (Gelfand and Smith, 1990; Geman and Geman, 1984) that eschew these problems. The basic idea behind MCMC is to simulate a Markov chain with the posterior density as its invariant probability density (see Appendix for definitions). After the chain has converged, resulting samples are an approximation of the posterior density.

MCMC methods come in two flavours — gradient-free schemes and gradient-based schemes. Gradient-free methods typically take the form of a Gibbs sampler or some variant of the random walk Metropolis–Hastings algorithm; whilst gradient-based methods use the gradient of the joint log-likelihood function to simulate diffusion (a Langevin algorithm) (Roberts and Tweedie, 1996) or optimise auxiliary variables as in Hamiltonian Monte-Carlo (HMC) algorithm (Neal, 2010). Despite the progress in numerical analysis, gradient-based methods are expensive; however, they avoid the naïve random walk inherent in gradient-free samplers. For both classes of samplers, there exists a natural trade-off — between rapid (Markov) chain mixing versus compute time efficiency. A computationally efficient sampler would reach the invariant probability density quickly (rapid mixing) using least floating-point cycles (computational efficiency). Since the inception of stochastic sampling methods (Gelfand and Smith, 1990; Geman and Geman, 1984) extensive measure theoretic analyses (Meyn and Tweedie, 2009) have been conducted to gauge the mixing properties of Markov chains but these are at worst problem dependent.

In this note, we evaluate the suitability of gradient-free MCMC methods in terms of unit computation required for producing an independent sample from the posterior distribution. For this, we implemented three variants of the Metropolis–Hastings algorithm; along with a slice sampling algorithm. In addition to the standard random walk Metropolis algorithm, we implemented a sampling algorithm that tunes the properties of the proposal distribution; whilst another

* Corresponding author.
*E-mail addresses:* b.sengupta@ucl.ac.uk (B. Sengupta), k.friston@ucl.ac.uk (K.J. Friston), w.penny@ucl.ac.uk (W.D. Penny).

http://dx.doi.org/10.1016/j.neuroimage.2015.03.008

scheme makes non-local moves across multiple Markov chains at different temperatures — facilitating cross-over between chains. In what follows, the parameters of a (single-node) neural mass model (NMM) were estimated using these inference schemes and their computational efficiency benchmarked. We found that adaptive Monte Carlo methods based on stochastic approximations were the most efficient, followed by MCMC methods based on tempered chains. Discounting computation efficiency, the slice-sampler emerged as a clear winner — if performance was restricted to the number of independent samples they could produce.

## Methods

In this section, we briefly review the generative (dynamic causal) model used to simulate data that was subject to subsequent inference. These models are used to fit observed electrophysiological data and contain between 10 and 100 parameters. We then review the schemes (random walk Metropolis Hastings sampling, slice sampling, adaptive MCMC sampling and population MCMC sampling) that are subject to a comparative evaluation in the Results section.

Custom code was written in Matlab 2014a (The MathWorks Inc., USA) to simulate the Markov chains. For population MCMC sampling Parallel Computing Toolbox (The MathWorks Inc., USA) was used. Unless stated otherwise, out of the 2000 samples that were collected, the initial 600 samples were discarded as burn-in (see Appendix for definitions). In adaptive MCMC, out of the 600 burn-in samples, the initial 300 samples were used for proposal adaptation. All computations were performed on a 2011 Macbook Pro laptop.

### *Neural mass models*

To test the inference schemes under known parameters, we used a single node neural mass model (NMM) based on David et al. (2006) to create synthetic data (Fig. 1). This model comprises ten parameters ($\{\delta, g, h, \tau, u\} \subseteq \theta$ with $\delta$ (intrinsic delay), $\{g_{1\ldots4}\}$ (connection strengths), $h_{e/i}$ (maximum amplitude of post-synaptic potential), $\tau_{e/i}$ (rate-constant of the membrane) and $u$ (input to the neural population); for detailed description refer to David et al. (2006)) and nine ordinary differential equations (ODEs) that are a first-order approximation of delay-differential equations (DDEs) representing three distinct neural populations; namely, inhibitory interneurons ($x_7$), spiny-stellate ($x_1$) and pyramidal neurons ($x_9$),

$$
\begin{aligned}
\dot{x}_1\ (t) &= x_4(t)\\
\dot{x}_2\ (t) &= x_5(t)\\
\dot{x}_3\ (t) &= x_6(t)\\
\dot{x}_4\ (t) &= \frac{h_e\left(g_1\left(\frac{1}{e^{-0.56x_9(t-\delta)}+1}-0.5\right)+u\right)}{\tau_e} - \frac{x_1(t)}{{\tau_e}^2} - \frac{2x_4(t)}{\tau_e}\\
\dot{x}_5\ (t) &= \frac{g_2 h_e\left(\frac{1}{e^{-0.56x_1(t-\delta)}+1}-0.5\right)}{\tau_e} - \frac{x_2(t)}{{\tau_e}^2} - \frac{2x_5(t)}{\tau_e}\\
\dot{x}_6\ (t) &= \frac{g_4 h_i\left(\frac{1}{e^{-0.56x_7(t-\delta)}+1}-0.5\right)}{\tau_i} - \frac{x_3(t)}{{\tau_i}^2} - \frac{2x_6(t)}{\tau_i}\\
\dot{x}_7\ (t) &= x_8(t)\\
\dot{x}_8\ (t) &= \frac{g_3 h_e\left(\frac{1}{e^{-0.56x_9(t-\delta)}+1}-0.5\right)}{\tau_e} - \frac{x_7(t)}{{\tau_e}^2} - \frac{2x_8(t)}{\tau_e}\\
\dot{x}_9\ (t) &= x_5(t) - x_6(t).
\end{aligned}
\tag{1}
$$

These differential equations simulated for $T$ time-points provide the predicted response (for some known experimental input) which, assuming additive Gaussian noise with covariance $\Sigma$, provide a likelihood model of observed data ($y$) with corresponding log joint density,

$$
\begin{aligned}
\mathcal{J} = &-\frac{1}{2}ln(|\Sigma|) - \frac{T}{2}ln(2\pi) - \frac{1}{2}(x_9(\theta)-y)^T\Sigma^{-1}(x_9(\theta)-y)\\
&+ln\left(\frac{1}{\Gamma(k_1){k_2}^{k_1}}\theta^{k_1-1}e^{-\frac{\theta}{k_2}}\right).
\end{aligned}
\tag{2}
$$

**Table 1**
Model parameters used for dynamic causal modelling. Parameters describing the prior Gamma distribution. Also shown are the parameters for generating the ground truth (Fig. 1).

| Parameter | Shape ($k_1$) | Scale ($k_2$) | True parameters |
|---|---|---|---|
| $g_1$ | 18.16 | 0.03 | 0.42 |
| $g_2$ | 29.9 | 0.02 | 0.76 |
| $g_3$ | 29.14 | 0.005 | 0.15 |
| $g_4$ | 30.77 | 0.007 | 0.16 |
| $\delta$ | 22.87 | 0.51 | 12.13 |
| $\tau_i$ | 34.67 | 0.23 | 7.77 |
| $h_i$ | 20.44 | 0.96 | 27.88 |
| $\tau_e$ | 33.02 | 0.16 | 5.77 |
| $h_e$ | 24.17 | 0.07 | 1.63 |
| $u$ | 23.62 | 0.13 | 3.94 |

Priors on all parameters (Table 1) conform to a Gamma distribution with shape $k_1$ and scale $k_2$, where – by construction – approximately 46–50% of the parameters sampled from this prior result in unstable dynamics, marked by positive real eigenvalues of the Jacobian matrix. This ensured that the inference scheme can recover from dynamical instability. The shape and scale of the Gamma distribution were determined numerically by integrating 200,000 NMMs and performing stability analysis. The shape and scale parameters of the Gamma prior distribution were then chosen, such that 46–50% of the sampled parameters produced unstable dynamics. The fixed-point equations were solved using a Trust-Region Dogleg method (Nocedal and Wright, 2006).

Contrary to David et al. (2006) where experimental input was modelled as a combination of a Gamma density function and a discrete cosine set, we used a simpler Heaviside step function to perturb the spiny-stellate cells. Differential equations were integrated using CVODES (Hindmarsh and Serban, 2002) using implicit backward-differentiation formulas (BDFs). The resulting non-linear equations were solved using Newton's method. Initial simulations established that direct solvers based on dense matrices were computationally more efficient than the three preconditioned Krylov (iterative) solvers (GMRES, Bi-CGStab, and TFQMR) (Golub and Van Loan, 2012). We anticipate that for larger dynamical systems (e.g., a 20-node NMM) iterative solvers may be more efficient. The absolute and relative tolerances of the integrators were both fixed at $10^{-3}$.

The source-code will be released as a general purpose 'Monte-Carlo inference' toolbox for SPM (http://www.fil.ion.ucl.ac.uk/spm/).

### *Algorithm A — slice sampler*

Slice-sampling is a type of MCMC based on the fact that sampling a random variable can be attained by sampling uniformly under its probability density function and rejecting those that are outside (Neal, 2003). First of all we initialise our parameters to $\theta_0$ so that the target density $\pi(\theta_0) > 0$. Given this previous sample $\theta_i$ we sample a position $n_{i+1}$ uniformly on $[0, \pi(\theta_i)]$. Conceptually, the next step comprises of drawing a horizontal line across the curve at this position. This hypothetical line is nothing but a 'slice' of our target distribution. Consequently, we sample $\theta_{i+1}$ along the slice so that $\pi(\theta_{i+1}) \geq n_{i+1}$.

Numerically, to operationalise the inequality, a bracket is first constructed as $\theta_{min} \leq \theta_{i+1} \leq \theta_{max}$ and tested to see whether each end point lies within the slice. If it does, the endpoint is extended in that direction until it is outside the slice. This process is called

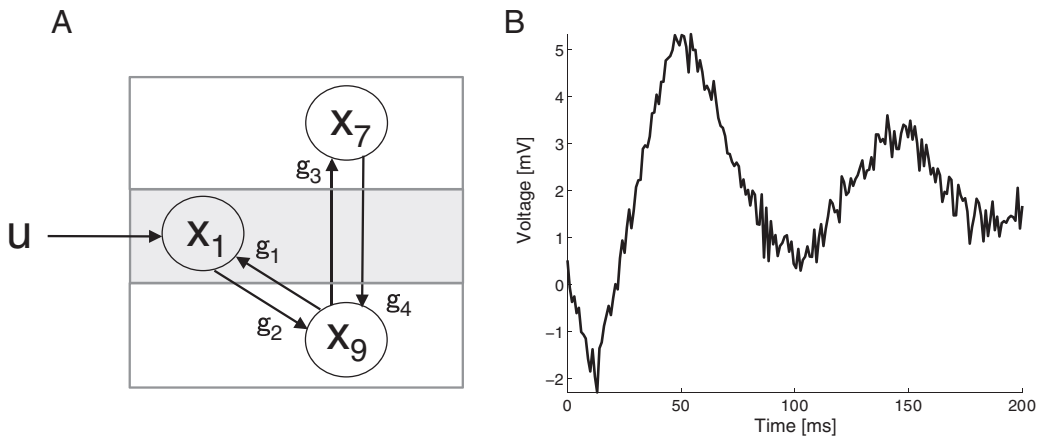


**Fig. 1.** A single node Neural Mass Model (NMM). (A) The forward model consists of 3 neural population — pyramidal ($x_9$), inhibitory interneuron ($x_7$) and spiny-stellate cells ($x_1$) connected by linearised delay links ($g_1, g_2, g_3$ and $g_4$) with $u$ as a Heaviside input. (B) The pyramidal cell voltage comprises the only observable of the model.

"stepping out". A candidate value $\widetilde{\theta}$ is then selected uniformly from the region — and is accepted as the next sample if it lies within the slice i.e., $\theta_{i+1} = \widetilde{\theta}$. If not, the slice shrinks, such that $\widetilde{\theta}$ forms one end of the slice containing $\theta_i$. The process is repeated until a sample is accepted. For multivariate distributions, we introduce an auxiliary position ($n$) for each dimension.

Unlike a Gibbs sampler, sampling using slices of the distribution does not require specification of the full conditionals. Similarly in contrast to a Metropolis–Hastings sampling algorithm, a slice-sampler does not require specification of a proposal distribution.

*Algorithm B — random walk metropolis sampler*

The random walk Metropolis (RWM) is the most common MCMC algorithm for Bayesian inference. Given a current value $\theta_i$ of a $d$-dimensional Markov chain, the next value is chosen according to a proposal distribution $\widetilde{\theta} \sim \pi\left(\widetilde{\theta}|\theta_i\right)$. We choose this to be a multi-variate Gaussian. The sample is then accepted with probability,

$$\alpha = 1 \wedge \frac{\pi\left(y|\widetilde{\theta}\right)\pi\left(\widetilde{\theta}\right) \times \pi\left(\theta|\widetilde{\theta}\right)}{\pi(y|\theta)\pi(\theta) \times \pi\left(\widetilde{\theta}|\theta\right)}. \tag{3}$$

$\wedge$ denotes minimum between the left and the right arguments. If $z \leq \alpha$ where $z \sim U(0,1)$ we set $\theta_{i+1} = \widetilde{\theta}$. Otherwise, we set $\theta_{i+1} = \theta_i$. The above formula embodies the notion that any proposal that takes the chain closer to a local mode is always accepted, whilst any other proposal is accepted with the probability equal to the relative densities of the posterior at the proposed and the current values.

*Algorithm C — adaptive MCMC sampler*

The random walk Metropolis (RWM) scheme generally has a slow convergence to the target density because of the inherent random walks (Chumbley et al., 2007). Using the history of samples that are generated from a Markov chain, the adaptive MCMC algorithm (Andrieu and Thoms, 2008; Haario et al., 2001) adapts the expectation and covariance matrix of the proposal distribution using stochastic approximations (Kushner and Yin, 2003). Stochastic approximation is an iterative algorithm that finds extrema (roots) of cost–functions using noisy samples. In adaptive MCMC this cost–function is based on the empirical mean ($\mu$) and covariance ($\Sigma$) of the target density as well as the pre-determined acceptance rate (to update the scalar scale parameter $\lambda$).

Specifically, a Robbins–Monro algorithm is used (Robbins and Monro, 1951); wherein given current parameters ($\theta_0$), mean ($\mu_0$) and covariance ($\Sigma_0$) of the proposal distribution we first sample $\theta_{i+1} \sim \mathcal{N}(\mu_i, \lambda_i \Sigma_i)$ where $\lambda_0$ is initialised to 1. Similarly, $\Sigma_0$ was initialised to an identity matrix. Secondly, using the Metropolis–Hastings criteria (Eq. (3)) we set $\mu_{i+1} = \theta_{i+1}$. If not, we reject the sample and set $\mu_{i+1} = \mu_i$. The current and target acceptance probabilities are $\alpha_{i \to i+1}$ and $\alpha_{target}$, respectively. It may be easy to understand such a scheme as a stochastic realisation of a deterministic prediction-error learning rule, guided by the Metropolis–Hastings acceptance ratio.

For the subsequent iteration, we adapt the mean ($\mu$), the covariance ($\Sigma$) and the global scale of the covariance matrix ($\lambda$) with an iteration dependent step-size ($\gamma$) as follows,

$$\gamma_{i+1} = \frac{1}{i+1} \tag{4}$$

$$\begin{aligned} log(\lambda_{i+1}) &= log(\lambda_i) + \gamma_{i+1}\left[\alpha_{i \to i+1} - \alpha_{target}\right] \\ \mu_{i+1} &= \mu_i + \gamma_{i+1}\left[\theta_{i+1} - \mu_i\right] \\ \Sigma_{i+1} &= \Sigma_i + \gamma_{i+1}\left[(\theta_{i+1} - \mu_i)(\theta_{i+1} - \mu_i)^T - \Sigma_i\right]. \end{aligned} \tag{5}$$

*Algorithm D — population MCMC sampler*

In the two preceding MCMC schemes, one simulates a single Markov chain, where the posterior sample density is said to have converged if multiple starting points yield identical invariant distributions. Slow chain mixing results from non-convexity of the posterior density. In order to promote chain mixing, one can run multiple chains with varying temperatures and implement non-local proposal swaps between chains (Geyer, 1992a). These exchanges also make the algorithm a candidate for sampling from multimodal densities (Frantz et al., 1990). Such an algorithm is known as the population MCMC sampler (Geyer, 1992a). It has been re-invented under various guises (Replica Exchange (Swendsen and Wang, 1986), Metropolis-Coupled MCMC (Geyer, 1992a), population MCMC (Laskey and Myers, 2003), Parallel Tempering (Earl and Deem, 2005), among others) but the standard approach is to initiate multiple Markov chains (indexed by i) totalling $N$ such that the inverse temperature ($\beta_i$) of each chain is distributed according to

$$\beta_i = 1 - \left(\frac{i}{N}\right)^p \tag{6}$$

where $p = 5$ and $N = 4$ chains, with the posterior density specified as,

$$\pi(\theta|y) \propto \pi(y|\theta)^{\beta}\pi(\theta). \tag{7}$$

With a $\beta$ of 1, the chain represents the joint log-likelihood, whilst a $\beta$ of 0 represents a Markov chain that samples from the prior — the lower the inverse-temperature, the smoother the posterior density. One can visualize the multi-modal target distribution melting with an increase in temperature. At each temperature, the resulting distribution is explored using a single Markov chain whilst a product distribution is considered when moving between individual chains. This enables the sampler to take into account all of the chains, at different temperature levels.

Operationally, each chain uses a local (to that chain) Metropolis–Hastings acceptance criterion to either accept or reject the sample i.e., $z < (1 \wedge \exp(\mathcal{H}_{\text{old}} - \mathcal{H}_{\text{new}}))$ where $z \sim U(0, 1)$ and $\mathcal{H}$ is the unnormalised joint log-likelihood. After the $k$th sample is collected from the local chain, an additional Metropolis–Hastings acceptance criteria is imposed — whereby a pair of chains ($t_i$ and $t_j$) is selected randomly and their samples are swapped with the following acceptance ratio,

$$1 \wedge \frac{\mathcal{L}\left(y\middle|\theta_j\right)^{t_i} \times \mathcal{L}(y|\theta_i)^{t_j}}{\mathcal{L}(y|\theta_i)^{t_i} \times \mathcal{L}\left(y\middle|\theta_j\right)^{t_j}} \tag{8}$$

$\mathcal{L}$ is the log-likelihood of the predicted response. It is customary to use a uniform tempering schedule; i.e., $\beta_i = \frac{i}{N}$ as discussed in Jasra et al.(2007). For linear regression models Calderhead and Girolami (2009) showed that a power law distribution for the tempering schedule is most optimal. They showed that using uniformly spaced temperature schedule on the other-hand produced worse results, even if the number of Markov chains is increased ten-fold. Therefore, in this paper, we evaluate the sampling efficiency as that obtained by both uniform and power-law temperature schedules.

Population MCMC is reminiscent of an embarrassingly parallel problem; wherein communication between multi-threaded processes is minimised by performing proposal exchange only after $k$ (fixed at 10) iterations.

This concludes our brief description of the gradient-free sampling schemes considered in this paper.

## Results

We used a single node neural mass model (NMM, Fig. 1) to characterise the computational performance of four different MCMC sampling algorithms, for parameter inference. Inference was performed under the assumption that the (neuronal) system is partially observable i.e., only the pyramidal cell voltage ($x_9$ in Eq. (1)) was available.

The efficiency of a MCMC sampler is defined as the ratio of the computation time and the number of effective samples produced in that time. The effective sample size (ESS) for each parameter is calculated using $\text{ESS} = R\left\{1 + 2\sum_q \gamma(q)\right\}^{-1}$, where $R$ is the number of posterior samples post-burn-in and $\sum_q \gamma(q)$ is the sum of $Q$ monotonic auto-correlations. This auto-correlation is estimated using the initial monotone sequence estimator (Theorem 3.1 in Geyer (1992b)). The minimum ESS reports the number of samples that is effectively uncorrelated over all parameters. Similarly, the time normalised (wall-time/minimum ESS) ESS tells us how much time we spend sampling a single uncorrelated sample, providing us with a measure of the worst-case behaviour (Cormen et al., 2001) of the sampling algorithm. In short, an efficient sampler would produce a large ESS at the shortest possible time.

We used a normal symmetric random walk Metropolis (RWM) scheme — as used in previous work on sampling schemes for DCM by Chumbley et al. (2007). The $l_2$ error-norm was among the highest over all schemes considered (10.8) (Fig. 2A, E). The RWM algorithm resulted in the lowest ESS (Table 2). This is because chain mixing is confounded by the inherent random walk displayed by this class of algorithm. The slice-sampler on the other hand had the highest ESS, albeit at increased computational cost (Fig. 2B, F). Also, it had the lowest $l_2$ error (Table 2).

In terms of computational time, adaptive Metropolis (Fig. 2C, G) with stochastic approximations of the mean, covariance and the scale of the proposal distribution emerged as the clear winner (Table 2). It had lower ESS in comparison to slice-sampling but took 90% less time to produce a single independent sample. The increased ESS reflects the fact that – unlike the RWM – the proposal distribution has been adapted to guarantee a prescribed acceptance rate of 23%.

So far we have only considered a single Markov chain. Multiple chains running at a variety of temperatures can be used to not only facilitate rapid chain mixing but also to sample from multi-modal posterior densities (Fig. 2D, H). At about 150% increase in compute time (with respect to adaptive MCMC), the population Metropolis method (with 4 chains running at 4 different temperatures with proposal swaps every 10 iterations) has the highest ESS after slice-sampling, but with a 6-fold decrease in compute time per independent sample. This is dependent upon the temperature spacing of the parallel chains; where uniform spacing of inverse temperature performs poorly (Table 2).

Adaptive Metropolis and population based Markov chains appear to be equally efficient; although the latter requires expert intervention in choosing the number of chains, the form of inverse temperature ladder and the selection of proposal exchange partners. Adaptive Metropolis on the other hand did not have any parameters that require tuning. In summary, for inversion of these sorts of DCMs, our (gradient-free) sampler of choice is the single chain adaptive MCMC algorithm.

## Discussion

In this note, we compared four gradient-free MCMC methods — random-walk Metropolis sampler, slice-sampler, adaptive MCMC sampler and population-based MCMC sampler in terms of their effective sample size (ESS). Both adaptive and population MCMC take between 0.4 and 0.6 min (on a 2011 Macbook Pro laptop) to generate a single uncorrelated sample. Adaptive MCMC does this by matching the proposal density to the required target density, whilst proposal-exchanges enable neighbouring chains to mix more quickly in population MCMC sampling. This is particularly useful for DCMs, where such population of Markov chains enable the inference algorithm to ameliorate issues like local minima and multi-modal posterior densities that are characteristic of many variational algorithms. The population MCMC that we have used is similar to a genetic algorithm (GA), where the samples from different chains interact to mimic natural selection. The key difference is that GAs find a single optimum point, whilst population MCMC furnishes a probability density. The temperature ladder can be seen as a cross-over process where fitter samples move to a lower temperature.

Poor-mixing results when the Markov chain is confined to isolated modes or mix poorly along samples with strong correlations. Indeed using multiple chains allows for a mode hopping characteristic for the underlying Markov chain. This is especially useful when sampling from multimodal posteriors. This is because non-local moves are made that result in crossing the barrier imposed by a local potential well. In addition to the population estimator that we have evaluated, a multitude of mode-hopping MCMC samplers exist for tortuous posterior densities. This ranges from Jump-walking (J-Walking) estimator (Frantz et al., 1990) with a potential problem of not satisfying detailed balance to Smart-darting (S-darting) where detailed balance of population walkers is maintained (Andricioaei et al., 2001).

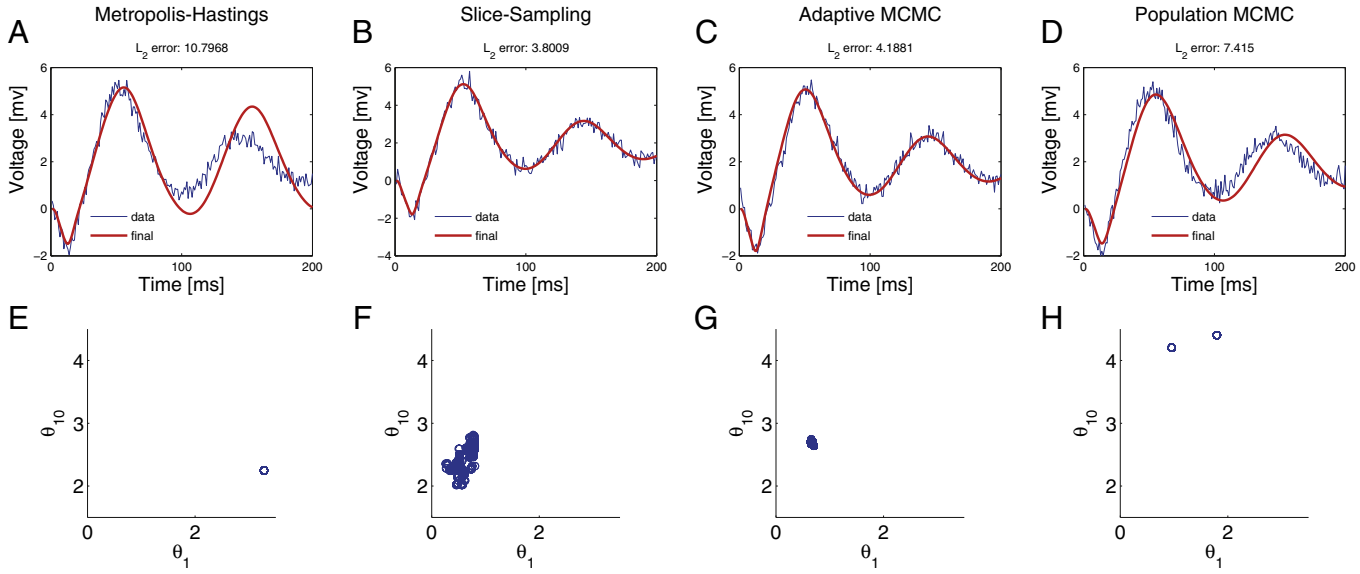


**Fig. 2.** Efficiency of the MCMC methods. (A) Predicted voltage using the posterior mean computed from 1400 samples based on random walk Metropolis–Hastings algorithm. (B) Same as A but with the slice-sampling algorithm. (C) Same as A but with adaptive Metropolis algorithm based on stochastic approximations. (D) Same as A but with population Metropolis algorithm based on proposal exchange. (E) Schematic displaying (effective) samples drawn from the posterior density using the MH algorithm. Parameters 1 and 10 are plotted. (F) Same as E but using the slice-sampling algorithm. (G) Same as E but using the adaptive Metropolis algorithm. (H) Same as E but using the population Metropolis algorithm.

In neuroimaging, the practitioner is not only interested in estimating the distribution of parameters that explain the EEG, MEG or fMRI signal but also build a variety of models to test competing hypothesis about the same observed data. Population MCMC – in the form of power posteriors – not only helps in parameter inference but also facilitates estimation of the partition function or model evidence. Via thermodynamic integration (Friel and Pettitt, 2008; Lartillot and Philippe, 2006) the model evidence can be obtained by running multiple Markov chains in parallel and numerically integrating samples over a variety of temperatures (using quadratures) to compute the model evidence (Eqn. 15 in Calderhead and Girolami (2009)). This becomes important when one has to choose between models using model comparison (Claeskens and Lid Hjort, 2008). Standard MCMC technology that relies on unnormalised probability densities infers the model evidence from samples generated by independent chains. Unlike thermodynamic integration, such estimates of the model evidence are highly variable — even in the high sample-size limit (Calderhead and Girolami, 2009; Jasra et al., 2007), rendering Bayesian model comparison useless. In population MCMC, since multiple chains take about the same amount of time as single chains due to the inherent parallelism of the scheme, one can evaluate model evidence without additional cost.

Unlike Lartillot and Philippe (2006) who suggest a uniform schedule for the temperature ladder, we verified that using a temperature ladder with power-law characteristics was beneficial (in terms of ESS) as pointed out by Calderhead and Girolami (2009). Such a schedule is not only beneficial from the point of the ESS but also in reducing the variance of the model evidence (Calderhead and Girolami, 2009). This is because geometric schedules – that model power-law densities – are the extremal solutions of the Monte Carlo variance (Gelman and Meng, 1998).

**Table 2**
Effective sample size (ESS) obtained from various samplers. Wall-time and average ESS for 10 parameters. Worst-case time normalised ESS is computed using the minimum ESS for each method.

| Sampler | Time (minutes) | Mean ESS (samples) | Time/min ESS (minutes/smpl) | $l_2$ error |
|---|---|---|---|---|
| Slice sampler | 11.8 | 7.23 | 3.68 | 3.8 |
| Metropolis–Hastings | 1.22 | 1 | 1.22 | 10.8 |
| Adaptive Metropolis | 1.06 | 4.07 | 0.38 | 4.2 |
| Population Metropolis (power) | 2.67 | 4.47 | 0.59 | 7.4 |
| Population Metropolis (uniform) | 2.61 | 1 | 2.61 | 8.6 |

Our inference algorithm based on adaptive MCMC sampler adapted the proposal distribution only during the burn-in iterations. This was done to avoid using past information infinitely often, preserving the Markov property of the transition kernel. An alternate methodology adopted by Gelfand and Sahu (1994) is to run several chains in parallel and use sampling-importance-resampling (SIR) (Rubin, 1998) to form kernels that have higher ESS whilst suppressing those chains that do not, using the approximation to the marginal distribution of the chain as a proposal distribution. It is vital to keep in mind that continued adaptation can disturb the invariant distribution of the chain. Although computationally inefficient, adaptation using delayed rejection (Tierney and Mira, 1999) or regeneration (Gilks et al., 1998) can be helpful.

An important issue – when using MCMC for Bayesian inference – is determining when the chain has converged. This criterion is crucial and therefore forms a large part of ongoing research that ascertains rapid convergence. Running an ergodic sampler for an infinite amount of time will result in convergence on the ground truth, per definition — tougher convergence criteria can necessitate longer runtimes. Measure-theoretic analysis of most MCMC samplers gives an estimate of the number of samples required to ensure convergence, according to a total variation distance (with a specified tolerance bound) to the true posterior density. For empirical problems this is seldom possible. A simpler but computationally wasteful strategy involves running multiple – yet independent – chains and ensuring that the posterior density obtained by each chain is identical in terms of its lower moments. A more cogent diagnostics to estimate convergence of the Markov chain uses the normal theory approximations of Gelman and Rubin (1992). This introduces a shrink factor that tends to 1 (depending on between chain and within chain convergence) as the Markov chain converges. For a discussion of convergence estimators, see Table 1 in Cowles and Carlin (1996).

There is no one sampler that is suitable for all inference problems; MCMC samplers that are based on geometric formulation of gradients, adaptation and tempering/annealing have over the years reduced concerns about local minima and sampling of multi-modal posteriors typically faced by deterministic algorithms. An important

future development would be in terms of combining gradient-free MCMC estimators with their gradient-based counterparts. For NMMs, we notice that the gradient based manifold Langevin samplers are computationally efficient (Sengupta et al., under review); yet the ESS is not as high as the Hamiltonian Monte Carlo (HMC) sampler. Thus, a gradient-based algorithm can be used as a starting algorithm followed with a population gradient-free MCMC sampler. The proposal distribution of each chain could be individually adapted, forming an adequate trade-off between computational time and the number of independent samples.

## Acknowledgments

BS, KJF and WDP are supported by the Wellcome Trust [091593/Z/10/Z]. BS is thankful to the Issac Newton Institute for Mathematical Sciences for hosting him during the 'Monte Carlo Inference for Complex Statistical Models' workshop. A part of this research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

## Appendix A. Basic MCMC terminology

Invariant distribution A probability distribution $\pi$ is called invariant if and only if $\pi\mathcal{T} = \mathcal{T}$ i.e., $\pi$ is a left eigenvector for $\mathcal{T}$ with eigenvalue 1. $\mathcal{T}$ is the transition kernel (a conditional probability).

Criteria for an invariant distribution For the distribution of $\theta_t$ to converge to its invariant or stationary distribution, the Markov chain has to be (a) irreducible — starting from any point the chain can reach any non-empty set with positive probability (also known as probabilistic connectedness condition), (b) aperiodic — returns to a state at irregular times; this stops the chain from oscillating and (c) positive recurrent — if the initial $\theta_0$ is sampled from $\pi(\cdot)$ then all subsequent iterates will also be distributed according to $\pi(\cdot)$.

Ergodicity of the Markov chain A state is ergodic if it is aperiodic and positive recurrent, which means that the state has a period of 1 and has finite average recurrence time. If all states of a (irreducible) Markov chain are ergodic, then the chain is said to be ergodic. Consequently, a Markov chain will have a unique invariant probability density (in our case the approximate posterior density) if and only if the states are positive recurrent.

Geometric ergodicity The distribution of $\theta$ is geometrically ergodic in total variation norm if it is (a) ergodic and (b) there exists a $\kappa$ in [0, 1) and a function $\mathcal{V}>1$ s.t. $\sum_j \left|\mathcal{T}_{ij}(t)-\pi(j)\right| \leq \mathcal{V}(i)\kappa^t$. The smallest $\kappa$ for which the function $\mathcal{V}$ exists is called the rate of convergence.

Uniform ergodicity An ergodic Markov chain is uniformly ergodic if there exists a finite constant $\mathcal{V}$ and a $\kappa$ in [0,1) s.t. $\sum_j \left|\mathcal{T}_{ij}(t)-\pi(j)\right| \leq \mathcal{V}\kappa^t$.

Convergence of RWM A symmetric random walk Metropolis Hastings (RWM) algorithm cannot be uniformly ergodic when the state space is not bounded (see Theorem 3.1 and 3.2 in Mengersen and Tweedie (1996)), although it can be geometrically ergodic. Geometric ergodicity is equivalent to the acceptance probability being uniformly bounded away from zero.

Burn-in Burn-in refers to the practice of discarding initial iterations of a Markov chain to specify initial distributions of the form $\pi\mathcal{T}^l$. $l$ is the number of burn-in iterations. Note that the strong law of large numbers and the central limit theorem holds regardless of the starting distribution.

## References

Andricioaei, I., Voter, A.F., Straub, J.E., 2001. Smart darting Monte Carlo. J. Chem. Phys. 114, 6994–7000 (0).
Andrieu, Christophe, Thoms, Johannes, 2008. A tutorial on adaptive MCMC. Stat. Comput. 180 (4), 343–373 (0).
Calderhead, Ben, Girolami, Mark, 2009. Estimating Bayes factors via thermodynamic integration and population MCMC. Comput. Stat. Data Anal. 53, 4028–4045 (0).
Chumbley, Justin R., Friston, Karl J., Fearn, Tom, Kiebel, Stefan J., 2007. A Metropolis–Hastings algorithm for dynamic causal models. Neuroimage 380 (3), 478–487 (0).
Claeskens, Gerda, Lid Hjort, Nils, 2008. Model selection and model averaging. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press (ISBN 978-0-521-85225-8).
Cormen, Thomas H., Stein, Clifford, Rivest, Ronald L., Leiserson, Charles E., 2001. Introduction to Algorithms. 2nd edition. McGraw-Hill Higher Education (0070131511).
Cowles, M.K., Carlin, B.P., 1996. Markov Chain Monte Carlo convergence diagnostics: a comparative review. J. Am. Stat. Assoc. 910 (434), 883–904 (0).
David, Olivier, Kilner, James M., Friston, Karl J., 2006. Mechanisms of evoked and induced responses in MEG/EEG. Neuroimage 310 (4), 1580–1591 (0).
Earl, David J., Deem, Michael W., 2005. Parallel tempering: theory, applications, and new perspectives. Phys. Chem. Chem. Phys. 70 (23), 3910–3916 (0).
Frantz, D.D., Freeman, D.L., Doll, J.D., 1990. Reducing quasi-ergodic behavior in Monte Carlo simulations by J-walking: applications to atomic clusters. J. Chem. Phys. 930 (4), 2769–2784 (0).
Friel, N., Pettitt, A.N., 2008. Marginal likelihood estimation via power posteriors. J. R. Stat. Soc. Ser. B Stat Methodol. 700 (3), 589–607 (0).
Friston, K.J., Harrison, L., Penny, W., 2003. Dynamic causal modelling. Neuroimage 190 (4), 1273–1302 (0).
Friston, K., Mattout, J., Trujillo-Barreto, N., Ashburner, J., Penny, W., 2007. Variational free energy and the Laplace approximation. Neuroimage 34, 220–234 (0).
Gelfand, A.E., Sahu, S.K., 1994. On Markov Chain Monte Carlo acceleration. J. Comput. Graph. Stat. 3, 261–276 (0).
Gelfand, Alan E., Smith, Adrian F.M., 1990. Sampling-based approaches to calculating marginal densities. J. Am. Stat. Assoc. 850 (410), 398–409 (0).
Gelman, Andrew, Meng, Xiao-Li, 1998. Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. Stat. Sci. 130 (2), 163–185 (0).
Gelman, Andrew, Rubin, Donald B., 1992. Inference from iterative simulation using multiple sequences. Stat. Sci. 70 (4), 457–472 (0).
Geman, Stuart, Geman, Donald, 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. 60 (6), 721–741 (0, November).
Geyer, Charles J., 1992a. Markov Chain Monte Carlo Maximum Likelihood. Defense Technical Information Center.
Geyer, Charles J., 1992b. Practical Markov Chain Monte Carlo. Stat. Sci. (ISSN: 08834237) 70 (4), 473–483 (0).
Gilks, W.R., Roberts, G.O., Sahu, S.K., 1998. Adaptive Markov Chain Monte Carlo through regeneration. J. Am. Stat. Assoc. 93, 763–769 (0).
Golub, Gene H., Van Loan, Charles F., 2012. Matrix Computations. 3rd ed. Johns Hopkins University Press, Baltimore, MD, USA.
Haario, H., Saksman, E., Tamminen, J., 2001. An adaptive Metropolis algorithm. Bernoulli 70 (2), 223–242 (0).
Hindmarsh, A., Serban, R., 2002. User documentation for CVODES, and ODE solver with sensitivity analysis capabilities. Technical Report. Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory.
Jasra, Ajay, Stephens, David A., Holmes, Christopher C., 2007. On population-based simulation for static inference. Stat. Comput. 170 (3), 263–279 (0).
Kushner, H., Yin, G., 2003. Stochastic Approximation and Recursive Algorithms and Applications. Springer.
Lartillot, Nicolas, Philippe, Hervé, 2006. Computing Bayes factors using thermodynamic integration. Syst. Biol. 550 (2), 195–207 (0).
Laskey, Kathryn, Myers, James, 2003. Population Markov Chain Monte Carlo. Machine Learning. University Press, pp. 175–196.
MacKay, David J.C., 2002. Information Theory, Inference & Learning Algorithms. Cambridge University Press, New York, NY, USA 0521642981.
Mengersen, K.L., Tweedie, R.L., 1996. Rates of convergence of the Hastings and Metropolis algorithms. Ann. Stat. 1, 101–121 (0).
Meyn, Sean, Tweedie, Richard L., 2009. Markov Chains and Stochastic Stability. 2nd edition. Cambridge University Press, New York, NY, USA (ISBN 0521731828, 9780521731829).
Neal, Radford M., 2003. Slice sampling. Ann. Stat. 310 (3), 705–767 (0).
Neal, Radford M., 2010. MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo 54 pp. 113–162 (0).
Nocedal, J., Wright, S.J., 2006. Numerical Optimization. 2nd edition. Springer, New York.
Robbins, H., Monro, S., 1951. A stochastic approximation method. Ann. Math. Stat. 22 (3), 400 (0).
Roberts, G., Tweedie, R., 1996. Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli 20 (4), 341–363 (0).
Rubin, D.B., 1998. Bayesian Statistics, Chapter Using the Sir Algorithm to Simulate Posterior Distributions. Oxford University Press, pp. 395–402.
Sengupta, B., Friston, K.J., Penny, W.D., 2015w. Gradient-based MCMC for dynamic causal modelling. Neuroimage (under review).

Swendsen, Robert H., Wang, Jian-Sheng, 1986. Replica Monte Carlo simulation of spin-glasses. Phys. Rev. Lett. 57, 2607–2609 (0).
Tierney, Luke, Kadane, Joseph B., 1986. Accurate approximations for posterior moments and marginal densities. J. Am. Stat. Assoc. 810 (393), 82–86 (0).
Tierney, L., Mira, A., 1999. Some adaptive Monte Carlo methods for Bayesian inference. Stat. Med. 18, 2507–2515 (0).
Wang, Bo, Titterington, D.M., 2004. Convergence and asymptotic normality of variational Bayesian approximations for exponential family models with missing values. Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, UAI'04. AUAI Press, Arlington, Virginia, United States, pp. 577–584.