Riemannian Manifold Hamiltonian Monte Carlo

Mark Girolami

Department of Computing Science, University of Glasgow, UK

Department of Statistics, University of Glasgow, UK

Ben Calderhead

Department of Computing Science, University of Glasgow, UK

Siu A. Chin

Department of Physics, Texas A&M University, Texas, USA

Summary. The paper proposes a Riemannian Manifold Hamiltonian Monte Carlo sampler to resolve the shortcomings of existing Monte Carlo algorithms when sampling from target densities that may be high dimensional and exhibit strong correlations. The method provides a fully automated adaptation mechanism that circumvents the costly pilot runs required to tune proposal densities for Metropolis-Hastings or indeed Hybrid Monte Carlo and Metropolis Adjusted Langevin Algorithms. This allows for highly efficient sampling even in very high dimensions where different scalings may be required for the transient and stationary phases of the Markov chain. The proposed method exploits the Riemannian structure of the parameter space of statistical models and thus automatically adapts to the local manifold structure at each step based on the metric tensor. A semi-explicit second order symplectic integrator for non-separable Hamiltonians is derived for simulating paths across this manifold which provides highly efficient convergence and exploration of the target density. The performance of the Riemannian Manifold Hamiltonian Monte Carlo method is assessed by performing posterior inference on logistic regression models, log-Gaussian Cox point processes, stochastic volatility models, and Bayesian estimation of parameter posteriors of dynamical systems described by nonlinear differential equations. Substantial improvements in the time normalised Effective Sample Size are reported when compared to alternative sampling approaches. Matlab code at `http://www.dcs.gla.ac.uk/inference/rmhmc` allows replication of all results.

1. Introduction

For an unnormalised probability density function, $\tilde{p}(\boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \mathbb{R}^D$, the normalised density follows as $p(\boldsymbol{\theta}) = \tilde{p}(\boldsymbol{\theta})/\int \tilde{p}(\boldsymbol{\theta})d\boldsymbol{\theta}$, which for many statistical models is analytically intractable. Monte Carlo estimates of integrals with respect to $p(\boldsymbol{\theta})$, which commonly appear in Bayesian statistics, are therefore required (Gilks *et al.*, 1996). The predominant methodology for sampling from such a probability density is Markov chain Monte Carlo (MCMC) see e.g. (Robert, 2004; Gelman *et al.*, 2004; Gilks *et al.*, 1996; Liu, 2001). The most general algorithm defining a Markov process with invariant density $p(\boldsymbol{\theta})$ is the *Metropolis-Hastings* algorithm (Metropolis *et al.*, 1953; Hasting, 1970), which is arguably one of the *most successful and influential* Monte Carlo algorithms (Beichl and Sullivan, 2000) .

The Metropolis-Hastings algorithm proposes transitions $\boldsymbol{\theta} \mapsto \boldsymbol{\theta}^*$ with density $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$, which are then accepted with probability $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\{1, p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)/p(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\}$. This acceptance probability ensures that the Markov chain is reversible with respect to the stationary target density $p(\boldsymbol{\theta})$ and satisfies detailed balance, see for example Robert (2004); Neal (1993a, 1996); Liu

(2001). Typically, the proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ which drives the Markov chain takes the form of a random walk, e.g. $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}^*|\boldsymbol{\theta}, \boldsymbol{\Lambda})$ is a D-dimensional Normal distribution with mean $\boldsymbol{\theta}$ and covariance $\boldsymbol{\Lambda}$.

High acceptance rates can be achieved by proposing smaller transitions, however larger amounts of time will then be required to make long traversals of parameter space. In high dimensions, when D is large, the random walk can become inefficient resulting in low acceptance rates, poor mixing of the chain and highly correlated samples. A consequence of this is a small effective sample size (ESS) from the chain, see Robert (2004); Gilks *et al.* (1996); Neal (1996); Liu (2001). Whilst there have been a number of suggestions to overcome this inefficiency, guaranteeing detailed balance and ergodicity of the chain places constraints on what can be achieved in alleviating this problem (Andrieu and Thoms, 2008; Robert, 2004; Neal, 1993a). Design of a good general purpose proposal mechanism providing large proposal transitions that are accepted with high probability remains something of an engineering art-form.

Major steps forward in this regard were made when a proposal process derived from a discretised Langevin diffusion with a deterministic component based on the gradient information of the target density was suggested in the Metropolis Adjusted Langevin Algorithm (MALA) (Roberts and Stramer, 2003). Likewise the Hybrid Monte Carlo (HMC) method (Duane *et al.*, 1987) was proposed in the statistical physics literature as a means of efficiently simulating states from a physical system which was then applied to problems of statistical inference (Neal, 1993a,b, 1996; Liu, 2001). In HMC, a deterministic proposal process is employed along with additional stochastic proposals that together provide an ergodic Markov chain capable of making large transitions that are accepted with high probability. Given the potential efficiency gains to be obtained in MCMC sampling from such proposal mechanisms a brief review of HMC within the context of statistical inference is provided in the following section. In Section 3, a generalisation of HMC is presented, which takes advantage of the natural Riemannian structure of the parameter space and allows for more efficient proposal transitions to be made. Finally, in Sections 4 and 5, this new methodology is demonstrated on a number of interesting statistical problems, i.e. Bayesian logistic regression, stochastic volatility modeling, log-Gaussian Cox point processes, and parameter inference in dynamical systems.

2. Hybrid Monte Carlo

Consider the random variable $\boldsymbol{\theta} \in \mathbb{R}^D$ with density $p(\boldsymbol{\theta})$ and an independent auxiliary variable $\mathbf{p} \in \mathbb{R}^D$ with density $p(\mathbf{p}) = \mathcal{N}(\mathbf{p}|\mathbf{0}, \mathbf{M})$. The joint density follows in factorised form as $p(\boldsymbol{\theta}, \mathbf{p}) = p(\boldsymbol{\theta})p(\mathbf{p}) = p(\boldsymbol{\theta})\mathcal{N}(\mathbf{p}|\mathbf{0}, \mathbf{M})$. Denoting the log of the desired density as $\mathcal{L}(\boldsymbol{\theta}) \equiv \log p(\boldsymbol{\theta})$, the negative joint log-likelihood is

$$H(\boldsymbol{\theta}, \mathbf{p}) = -\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2}\log(2\pi)^D|\mathbf{M}| + \frac{1}{2}\mathbf{p}^\mathsf{T}\mathbf{M}^{-1}\mathbf{p} \tag{1}$$

The physical analogy of this negative joint log-likelihood is a Hamiltonian (Duane *et al.*, 1987; Leimkuhler and Reich, 2004), which describes the sum of a potential energy function $-\mathcal{L}(\boldsymbol{\theta})$ defined at the position $\boldsymbol{\theta}$, and a kinetic energy term $\mathbf{p}^\mathsf{T}\mathbf{M}^{-1}\mathbf{p}/2$. The auxiliary variable $\mathbf{p}$ is interpreted as a momentum variable and the covariance matrix $\mathbf{M}$ denotes a mass matrix.

The score function (Schervish, 1995), with respect to $\boldsymbol{\theta}$ and $\mathbf{p}$, of the log joint density over the two random variables has a physical interpretation as the time evolution, with respect to a fictitious time τ, of the physical system as given by Hamilton's equations,

$$\frac{d\boldsymbol{\theta}}{d\tau} = \frac{\partial H}{\partial \mathbf{p}} = \mathbf{M}^{-1}\mathbf{p} \qquad \frac{d\mathbf{p}}{d\tau} = -\frac{\partial H}{\partial \boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) \tag{2}$$

As the derivative $\partial U_n/\partial \delta_n = -\delta_n(\mathbf{f}_n - \mathbf{m}_n)^\mathsf{T}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)$ then

$$\begin{aligned}\mathcal{I}_{\delta_n} &= \gamma_n E\left\{(\mathbf{f}_n - \mathbf{m}_n)^\mathsf{T}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)(\mathbf{f}_n - \mathbf{m}_n)^\mathsf{T}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)\right\} - \\ &\quad \gamma_n E\left\{(\mathbf{f}_n - \mathbf{m}_n)^\mathsf{T}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)\right\}^2\end{aligned}$$

As in the previous section we approximate the sampling density $p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$ by the surrogate Gaussian $\mathcal{N}(\mathbf{m}_n|\mathbf{f}_n(\mathbf{X}, \boldsymbol{\theta}, \mathbf{t}), \mathbf{K}_n + \mathbf{I}\gamma_n)$, in which case after some standard manipulation

$$\mathcal{I}_{\delta_n} \approx 2\gamma_n \mathsf{trace}\left((\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}\right) \tag{56}$$

E. Metric Tensor & Derivative of Gaussian Process Regression Model

The marginal likelihood for a linear regression model under a GP prior is defined in the main text as $p(\mathbf{y}|\boldsymbol{\varphi}, \sigma) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K})$, where $\mathbf{K} = \sigma^2\mathbf{I} + \mathbf{C}_{\boldsymbol{\varphi}}$ and denoting the full set of parameters as $\boldsymbol{\phi} \equiv (\boldsymbol{\varphi}, \boldsymbol{\sigma})$ then the derivative of the log-marginal and the metric tensor - Fisher Information matrix - follow in standard form as

$$\frac{\partial}{\partial \phi_i}\log\left(p(\mathbf{y}|\boldsymbol{\phi})\right) = \frac{1}{2}\mathsf{trace}\left(\left(\mathbf{K}^{-1}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}^{-1} - \mathbf{K}^{-1}\right)\frac{\partial \mathbf{K}}{\partial \phi_i}\right) \tag{57}$$

$$\mathbf{G}(\boldsymbol{\phi})_{ij} = \frac{1}{2}\mathsf{trace}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \phi_i}\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \phi_j}\right) \tag{58}$$

Application of standard derivative of trace operators provides an analytic expression for the derivative of the metric tensor with respect to the parameters

$$\frac{\partial \mathbf{G}(\boldsymbol{\phi})_{ij}}{\partial \phi_k} = \frac{\partial}{\partial \phi_k}\left[\frac{1}{2}\mathsf{trace}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \phi_i}\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \phi_j}\right)\right] \tag{59}$$

In our experiments we employ an infinitely differentiable stationary covariance function to calculate the $(i,j)^{th}$ entry of the covariance matrix,

$$\mathbf{K}_{i,j} = \varphi_1 \exp\left(-\frac{1}{2\varphi_2^2}(t_j - t_i)^2\right) + \sigma\delta_{ij} \tag{60}$$

The Fisher Information matrix above may therefore be obtained using the first and second partial derivatives of the covariance function. The first partial derivatives follow as,

$$\frac{\partial \mathbf{K}_{i,j}}{\partial \varphi_1} = \frac{1}{\varphi_1}(\mathbf{K}_{i,j} - \sigma\delta_{ij}), \quad \frac{\partial \mathbf{K}_{i,j}}{\partial \varphi_2} = \frac{1}{\varphi_2^3}(\mathbf{K}_{i,j} - \sigma\delta_{ij})(t_j - t_i)^2, \quad \frac{\partial \mathbf{K}_{i,j}}{\partial \sigma} = \delta_{ij}$$

The second partial derivatives may also be easily calculated, and indeed out of the nine second partial derivatives, only three of them are non-zero which eases their computation.

$$\begin{aligned}\frac{\partial^2 \mathbf{K}_{i,j}}{\partial \varphi_1^2} &= \frac{\partial^2 \mathbf{K}_{i,j}}{\partial \varphi_1 \partial \sigma} = \frac{\partial^2 \mathbf{K}_{i,j}}{\partial \varphi_2 \partial \sigma} = \frac{\partial^2 \mathbf{K}_{i,j}}{\partial \sigma \partial \varphi_1} = \frac{\partial^2 \mathbf{K}_{i,j}}{\partial \sigma \partial \varphi_2} = \frac{\partial^2 \mathbf{K}_{i,j}}{\partial \sigma^2} = 0 \\ \frac{\partial^2 \mathbf{K}_{i,j}}{\partial \varphi_1 \partial \varphi_2} &= \frac{1}{\varphi_1}\frac{\partial \mathbf{K}_{i,j}}{\partial \varphi_2}, \quad \frac{\partial^2 \mathbf{K}_{i,j}}{\partial \varphi_2^2} = \frac{\partial \mathbf{K}_{i,j}}{\partial \varphi_1}\frac{\varphi_1}{\varphi_2^6}(1 - 3\varphi_2^2)(t_j - t_i)^2\end{aligned}$$

D. Derivation of Metric Tensor for Nonlinear Differential Equations

We consider both the ODE parameters $\boldsymbol{\theta}$ and the corresponding error variances $\boldsymbol{\gamma}$ independently when sampling from the conditional posterior (21) and so consider the Fisher Information for each set of variables separately. Let us first of all consider $\boldsymbol{\theta}$. The sampling density is $p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$ and so the log-likelihood takes the form

$$\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) = -\frac{1}{2}U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) - \log\left(\mathcal{Z}(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})\right) \tag{53}$$

D.1. Fisher Information for ODE parameters $\boldsymbol{\theta}$

A straightforward result shows that the Fisher Information matrix for the above generic form of likelihood is

$$\mathcal{I}_{\boldsymbol{\theta}} = E\left\{\nabla_{\boldsymbol{\theta}}\mathcal{L}\nabla_{\boldsymbol{\theta}}\mathcal{L}^{\mathsf{T}}\right\} = \frac{1}{4}E\left\{\nabla_{\boldsymbol{\theta}}U\nabla_{\boldsymbol{\theta}}U^{\mathsf{T}}\right\} - \frac{1}{4}E\left\{\nabla_{\boldsymbol{\theta}}U\right\}E\left\{\nabla_{\boldsymbol{\theta}}U^{\mathsf{T}}\right\} \tag{54}$$

where the expectation is taken with respect to $p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$. Now the required derivative vector is $\nabla_{\boldsymbol{\theta}}U = \sum_{n=1}^{N}\mathbf{F}_n(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)$ where the $D \times T$ matrix $\mathbf{F}_n$ has elements $\mathbf{F}_{n_{d,s}} = \partial\mathbf{f}_n(\boldsymbol{\theta}, \mathbf{X}, t = s)/\partial\theta_d$. It then follows that

$$\begin{aligned}\mathcal{I}_{\boldsymbol{\theta}} &= \frac{1}{4}\sum_{n=1}^{N}E\left\{\mathbf{F}_n(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)(\mathbf{f}_n - \mathbf{m}_n)^{\mathsf{T}}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}\mathbf{F}_n^{\mathsf{T}}\right\} - \\ &\quad \frac{1}{4}\sum_{n=1}^{N}E\left\{\mathbf{F}_n(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)\right\}\sum_{n=1}^{N}E\left\{(\mathbf{f}_n - \mathbf{m}_n)^{\mathsf{T}}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}\mathbf{F}_n^{\mathsf{T}}\right\}\end{aligned}$$

An exact analytic form does not follow due to the nonlinearity of the function $U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$ therefore at this stage estimates of the Fisher Information must be made by sampling from the density $p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$ or approximations must be made.

Here we make approximations by employing a surrogate sampling density for $p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$ over the random vectors $\mathbf{m}_n$ which is $\mathcal{N}(\mathbf{m}_n|\mathbf{f}_n(\mathbf{X}, \boldsymbol{\theta}, \mathbf{t}), \mathbf{K}_n + \mathbf{I}\gamma_n)$ for each n. One further approximation we make is that the elements of the matrices $\mathbf{F}_n$ are constant relative to the vectors $\mathbf{f}_n$ and $\mathbf{m}_n$. With these in place and noting that $E\{\mathbf{f}_n - \mathbf{m}_n\} = \mathbf{0}$ and $E\{(\mathbf{f}_n - \mathbf{m}_n)(\mathbf{f}_n - \mathbf{m}_n)^{\mathsf{T}}\} = \mathbf{K}_n + \mathbf{I}\gamma_n$ under the surrogate density, then

$$\mathcal{I}_{\boldsymbol{\theta}} \cong \sum_{n=1}^{N}\mathbf{F}_n(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}\mathbf{F}_n^{\mathsf{T}}$$

Whilst approximations have been made to arrive at this convenient analytic form for the Fisher Information it should be highlighted that in terms of employing this expression as a metric tensor the elements are in place to describe approximately the local geometric structure given that the elements of each matrix $\mathbf{F}_n$ are the time derivatives of the sensitivity coefficients of the systems of ODEs. However we discuss the implications of this approximation further in Section 7.3.

D.2. Fisher Information for Model Mismatch Variance γ

The Fisher Information for each $\sqrt{\gamma_n}$, which we denote by δ_n, is represented as

$$\mathcal{I}_{\delta_n} = E\left\{\left(\frac{\partial U}{\partial\delta_n}\right)^2\right\} - E\left\{\frac{\partial U}{\partial\delta_n}\right\}^2 \tag{55}$$

These deterministic equations can be exploited in defining a proposal process for both sets of random variables by firstly drawing a sample of $\mathbf{p}$ from $\mathcal{N}(\mathbf{p}|\mathbf{0}, \mathbf{M})$, then numerically integrating equation (2) to provide the evolution process in the joint space. If the numerical integrator is such that it provides mappings $(\boldsymbol{\theta}, \mathbf{p}) \mapsto (\boldsymbol{\theta}^*, \mathbf{p}^*)$ that are both time-reversible and volume preserving, then the use of the Hastings ratio in defining an acceptance probability $\min[1, \exp\{-H(\boldsymbol{\theta}^*, \mathbf{p}^*) + \mathbf{H}(\boldsymbol{\theta}, \mathbf{p})\}]$ produces an ergodic, time reversible Markov chain that satisfies detailed balance and whose stationary density is $p(\boldsymbol{\theta}, \mathbf{p})$ (Duane *et al.*, 1987; Liu, 2001; Neal, 1996).

The class of explicit symplectic numerical integrators are both time reversible and volume preserving (Leimkuhler and Reich, 2004) and as such would be appropriate in devising the desired Markov chain. The Leapfrog algorithm was introduced as a symplectic integrator in the original paper of Duane *et al.* (1987), and employed in statistical applications e.g. (Liu, 2001; Neal, 1993b) as described below,

$$\begin{aligned}\mathbf{p}(\tau + \epsilon/2) &= \mathbf{p}(\tau) + \epsilon\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}(\tau))/2 &(3)\\ \boldsymbol{\theta}(\tau + \epsilon) &= \boldsymbol{\theta}(\tau) + \epsilon\mathbf{M}^{-1}\mathbf{p}(\tau + \epsilon/2) &(4)\\ \mathbf{p}(\tau + \epsilon) &= \mathbf{p}(\tau + \epsilon/2) + \epsilon\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}(\tau + \epsilon))/2 &(5)\end{aligned}$$

Since the joint likelihood is factorisable (i.e. in physical terms, the Hamiltonian is separable), it is obvious by inspection that each complete Leapfrog step (equations (3), (4) and (5)) is reversible by the negation of the integration step-size, ϵ. Likewise as the Jacobians of the transformations $(\boldsymbol{\theta}, \mathbf{p}) \mapsto (\boldsymbol{\theta}, \mathbf{p} + \epsilon\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})/2)$ and $(\boldsymbol{\theta}, \mathbf{p}) \mapsto (\boldsymbol{\theta} + \epsilon\mathbf{M}^{-1}\mathbf{p}, \mathbf{p})$ have unit determinant then volume is preserved, and thus detailed balance will be satisfied in an HMC scheme that employs an acceptance ratio $\min[1, \exp\{-H(\boldsymbol{\theta}^*, \mathbf{p}^*) + H(\boldsymbol{\theta}, \mathbf{p})\}]$. Random values of $\mathbf{p} \sim \mathcal{N}(\mathbf{p}|\mathbf{0}, \mathbf{M})$ are used prior to each deterministic sequence of Leapfrog steps to ensure the full space is explored, and consequently the ergodicity of the chain is preserved, see Neal (1996); Liu (2001) for a detailed description of the HMC procedure.

It should be noted that the combination of equations (3) and (4) in a single step of the Leapfrog algorithm yields an update of the form

$$\boldsymbol{\theta}(\tau + \epsilon) = \boldsymbol{\theta}(\tau) + \frac{\epsilon^2}{2}\mathbf{M}^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}(\tau)) + \epsilon\mathbf{M}^{-1}\mathbf{p}(\tau) \tag{6}$$

which is nothing more than a discrete pre-conditioned Langevin diffusion as employed in MALA (Roberts and Stramer, 2003) (see Neal (1993a, 1996) for further discussion on this point).

The ability of HMC to overcome random walks in MCMC sampling suggests it should be a highly successful tool for Bayesian inference. A study suggests in excess of 300 papers cite the original (Duane *et al.*, 1987) paper within the literature devoted to Molecular Modelling and Simulation, Physics and Chemistry. However there are a much smaller number of citations in the literature devoted to Statistical Methodology and Application, e.g. (Liu, 2001; Neal, 1996, 1993b; Gustafson, 1997; Ishwaran, 1999), indicating that it may have largely passed into desuetude.

Whilst the choice of the step size ϵ and number of Leapfrog steps can be tuned based on the overall acceptance rate of the HMC sampler, it is unclear how to select the values of the weight matrix $\mathbf{M}$ in any automated manner that does not require some knowledge of the target density. Although rules of thumb are suggested (Liu, 2001; Neal, 1993a, 1996) these typically rely on knowledge of the marginal variance of the target density, which is of course not known at the time of simulation and thus requires preliminary pilot runs of HMC, this is also the case for MALA although asymptotic settings are suggested in Christensen *et al.* (2005). The experimental sections of this paper will demonstrate how crucial this tuning is to obtain acceptable performance of HMC and MALA.

The potential of the HMC methodology may be more fully realised by employing *stochastic* transitions that take into account the local structure of the target density when proposing moves to different likelihood regions, as this may improve the overall mixing of the chain. Therefore rather than employing a fixed global covariance matrix in the proposal density $\mathcal{N}(\mathbf{p}|\mathbf{0}, \mathbf{M})$, a position specific covariance would be adopted. Furthermore, the *deterministic* proposal mechanism of HMC, when viewed as the deterministic component of the discrete pre-conditioned Langevin diffusion, equation (6), relies on the likelihood gradient pre-conditioned by the inverse of a globally constant metric tensor i.e. a mass matrix. However, given the Riemannian structure of the parameter space of statistical models (Amari, 1990; Kass, 1989) the adoption of the position specific metric tensor should yield more effective deterministic transitions in the overall algorithm. The following section now formalises both of the above considerations by defining the overall Hamiltonian on the Riemann Manifold.

3. Riemann Manifold Hamiltonian Monte Carlo

The parameter space of a statistical model possesses a Riemannian structure (Amari, 1990, 1997; Kass, 1989; Murray and Rice, 1993), whose invariant metric is defined by the Fisher Information (Rao, 1945; Amari, 1990, 1997). Therefore the natural geometric structure of the density model $p(\boldsymbol{\theta})$ is defined by the Riemannian manifold and associated metric tensor. Zlochin and Baram (2001) originally attempted to exploit this manifold structure, however their use of non-symplectic numerical integration prevented them from developing an overall HMC procedure and resulted in an approximate method of simulation instead of a proper MCMC algorithm, drastically limiting its applicability and usefulness. We show how the Riemannian manifold structure may be exploited within a correct MCMC framework. This overcomes the difficulties of implementing HMC and yields an automated means of tuning the overall method. We begin by considering the definition of Hamiltonian dynamics on the Riemannian manifold (Chavel, 1993).

As the Hamiltonian is $-\log p(\boldsymbol{\theta}, \mathbf{p}) = -\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}(\mathbf{p})$ the parameterised family of probability densities $p(\boldsymbol{\theta})$ is a D-dimensional Riemann manifold with metric tensor, $\mathbf{G}(\boldsymbol{\theta})$, defined by the non-degenerate Fisher Information matrix $E\{\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})^{\mathsf{T}}\}$, (Rao, 1945; Amari, 1990, 1997). From equation (2), it follows that $\mathbf{p} = \mathbf{M}\dot{\boldsymbol{\theta}}$, so the norm of each $\dot{\boldsymbol{\theta}}$ under the metric $\mathbf{M}$ follows as $\|\dot{\boldsymbol{\theta}}\|^2_{\mathbf{M}} = \dot{\boldsymbol{\theta}}^{\mathsf{T}}\mathbf{M}\dot{\boldsymbol{\theta}} = \mathbf{p}^{\mathsf{T}}\mathbf{M}^{-1}\mathbf{p}$. In a more general form as the statistical model is defined on a Riemannian manifold, the metric tensor defines the position specific norm such that $\|\dot{\boldsymbol{\theta}}\|^2_{\mathbf{G}(\boldsymbol{\theta})} = \dot{\boldsymbol{\theta}}^{\mathsf{T}}\mathbf{G}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \mathbf{p}^{\mathsf{T}}\mathbf{G}^{-1}(\boldsymbol{\theta})\mathbf{p}$ and thus the kinetic energy term can be defined via the inverse metric. In order to ensure that the Hamiltonian can be interpreted as a log-density, the addition of the normalising term for the Gaussian is required, i.e. $\frac{1}{2}\log(2\pi)^D|\mathbf{G}(\boldsymbol{\theta})|$. Therefore, the Hamiltonian defined on the Riemann manifold follows as

$$H(\boldsymbol{\theta}, \mathbf{p}) = \phi(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^{\mathsf{T}}\mathbf{G}(\boldsymbol{\theta})^{-1}\mathbf{p} \tag{7}$$

where $\phi(\boldsymbol{\theta}) = -\mathcal{L}(\boldsymbol{\theta}) + \frac{1}{2}\log(2\pi)^D|\mathbf{G}(\boldsymbol{\theta})|$ so that $\exp(-H(\boldsymbol{\theta}, \mathbf{p})) = p(\boldsymbol{\theta}, \mathbf{p}) = p(\boldsymbol{\theta})p(\mathbf{p}|\boldsymbol{\theta})$ and the marginal density $p(\boldsymbol{\theta}) = \int \exp(-H(\boldsymbol{\theta}, \mathbf{p}))d\mathbf{p}$ is the desired target density. Unlike the previous case for HMC this joint density is no longer factorisable and therefore the log-likelihood does not correspond to a separable Hamiltonian. The conditional distribution for momentum values given parameter values is a zero-mean Gaussian with the point specific metric tensor acting as the covariance matrix $p(\mathbf{p}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{p}|\mathbf{0}, \mathbf{G}(\boldsymbol{\theta}))$, which in part resolves the scaling issues associated with HMC and MALA, as will be demonstrated in Sections 4 and 7. The following section develops an explicit symplectic integrator for the Riemannian Manifold Hamiltonian Monte Carlo method.

$$\mathbf{G} = \begin{bmatrix} \frac{2T}{\beta^2} & 0 & 0 \\ 0 & T+1 & 2\phi \\ 0 & 2\phi & \phi^2(3-T)+(T-1) \end{bmatrix}$$

and the partial derivatives of the metric tensor follow as

$$\frac{\partial \mathbf{G}}{\partial \beta} = \begin{bmatrix} -\frac{4T}{\beta^3} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \frac{\partial \mathbf{G}}{\partial \gamma} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \frac{\partial \mathbf{G}}{\partial \alpha} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 2(1-\phi^2) \\ 0 & 2(1-\phi^2) & 2\phi(1-\phi^2)(3-T) \end{bmatrix}$$

C. Derivation of Conditional Density for ODE parameters

A model of the density $p(\dot{\mathbf{X}}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$ is required which will assign high probability mass to values of the state derivatives when the ODE parameters and the corresponding GP parameters are consistent. One way this can be achieved is to ensure that the GP regression model, $\mathcal{N}(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{m}_n, \mathbf{K}_n)$, and the model of structural mismatch, $\mathcal{N}(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{f}_n(\mathbf{X}, \boldsymbol{\theta}, \mathbf{t}), \mathbf{I}\gamma_n)$, both assign similar probability mass to state-derivative values having consistent regression and structural (ODE) parameters. This requirement can be satisfied by making the modeling choice that, $p(\dot{\mathbf{X}}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$, be represented as a product of Gaussians such that

$$p(\dot{\mathbf{X}}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) = \frac{p(\dot{\mathbf{X}}, \mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})}{p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})} = \frac{\prod_n \mathcal{N}(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{m}_n, \mathbf{K}_n)\mathcal{N}(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{f}_n(\mathbf{X}, \boldsymbol{\theta}, \mathbf{t}), \mathbf{I}\gamma_n)}{\prod_n \mathcal{N}(\mathbf{m}_n|\mathbf{f}_n(\mathbf{X}, \boldsymbol{\theta}, \mathbf{t}), \mathbf{K}_n + \mathbf{I}\gamma_n)}$$

By equating both denominators of the above expression to obtain the marginal density for the state values after the state derivatives have been marginalised and denoting $U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) = \sum_{n=1}^{N}(\mathbf{f}_n - \mathbf{m}_n)^{\mathsf{T}}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)$ then it is clear that

$$p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) = \frac{1}{\mathcal{Z}(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})} \exp\left(-\frac{1}{2}U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})\right)$$

where the normalising constant is simply $\mathcal{Z}(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) = \int \exp\left(-\frac{1}{2}U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})\right) d\mathbf{X}$. It then follows that

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\gamma}|\mathbf{X}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) &= \frac{1}{p(\mathbf{X}|\boldsymbol{\varphi})}\frac{1}{\mathcal{Z}(\boldsymbol{\varphi}, \boldsymbol{\sigma})}\exp\left(-\frac{1}{2}U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})\right)\pi(\boldsymbol{\theta}, \boldsymbol{\gamma}) \\ &= \frac{1}{\prod_n \mathcal{N}(\mathbf{X}_{n,\cdot}|\mathbf{0}, \mathbf{C}_{\boldsymbol{\varphi}_n})}\frac{1}{\mathcal{Z}(\boldsymbol{\varphi}, \boldsymbol{\sigma})}\exp\left(-\frac{1}{2}U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})\right)\pi(\boldsymbol{\theta}, \boldsymbol{\gamma}) \end{aligned}$$

where $\mathcal{Z}(\boldsymbol{\varphi}, \boldsymbol{\sigma}) = \int \exp\left(-\frac{1}{2}U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})\right)\pi(\boldsymbol{\theta}, \boldsymbol{\gamma})d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\gamma}$. Now as each of the terms appearing in the denominator $\mathcal{Z}(\boldsymbol{\varphi}, \boldsymbol{\sigma})\prod_n \mathcal{N}(\mathbf{X}_{n,\cdot}|\mathbf{0}, \mathbf{C}_{\boldsymbol{\varphi}_n})$ will be the same value in the numerator and denominator of the Hastings ratio then it cancels out giving that

$$p(\boldsymbol{\theta}, \boldsymbol{\gamma}|\mathbf{X}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) \propto \exp\left(-\frac{1}{2}U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})\right)\pi(\boldsymbol{\theta}, \boldsymbol{\gamma}) \tag{52}$$

Since the gradients appear only linearly and their conditional distribution given $\mathbf{X}$ is Gaussian they can be marginalized exactly. In other words, given observations $\mathbf{Y}$, we can sample from the conditional distribution for $\mathbf{X}$ and marginalize the augmented derivative space. The differential equation need never now be explicitly solved, its implicit solution is integrated into the sampling scheme.

$$
\begin{aligned}
\mathbf{p}_1 &= \mathbf{p}_0 + (\epsilon/2)\boldsymbol{\varphi}(\boldsymbol{\theta}_0) \\
\boldsymbol{\theta}_1 &= \boldsymbol{\theta}_0 + (\epsilon/2)\mathbf{G}^{-1}(\boldsymbol{\theta}_0)\mathbf{p}_1 \\
\mathbf{p}_2 &= \mathbf{g}(\boldsymbol{\theta}_1, \mathbf{p}_1, \epsilon) \\
\boldsymbol{\theta} &= \boldsymbol{\theta}_1 + (\epsilon/2)\mathbf{G}^{-1}(\boldsymbol{\theta}_1)\mathbf{p}_2 \\
\mathbf{p} &= \mathbf{p}_2 + (\epsilon/2)\boldsymbol{\varphi}(\boldsymbol{\theta})
\end{aligned}
$$

where the second and fourth step may be iterated if required. The necessary time-reversible volume preserving integrator is now available to fully define the RM-HMC algorithm.

B. Derivation of Stochastic Volatility Equations

The joint likelihood can be written as

$$
p(\mathbf{y}, \mathbf{x}, \beta, \sigma, \phi) = \prod_{t=1}^{T} p(y_t|x_t, \beta)p(x_1)\prod_{t=2}^{T} p(x_t|x_{t-1}, \sigma, \phi)\pi(\beta)\pi(\sigma)\pi(\phi) \tag{44}
$$

where, following Liu (2001), we use the priors $p(\beta^2) \propto \beta^{-2}$, $\sigma^2 \sim$ Inv-$\chi^2(10, 0.05)$ and $(\phi - 1)/2 \sim$ Beta$(20, 1.5)$. We now introduce a transformation of the parameters to ensure they have the appropriate support when sampling i.e. $\sigma = \exp(\gamma)$ and $\phi = \tanh(\alpha)$, and the partial derivatives of log-joint likelihood with respect to transformed parameters are as follows

$$
\begin{aligned}
\frac{\partial L}{\partial \beta} &= -\frac{T}{\beta} + \sum_{t=1}^{T} \frac{y_t^2}{\beta^3 \exp(x_t)} - \frac{1}{\beta} && (45) \\
\frac{\partial L}{\partial \gamma} &= \left(-T + \frac{x_1^2(1-\phi^2)}{\sigma^2} + \sum_{t=2}^{T} \frac{(x_t - \phi x_{t-1})^2}{\sigma^2}\right) + \frac{v\rho}{\sigma^2} - v && (46) \\
\frac{\partial L}{\partial \alpha} &= \left(-\frac{\phi}{(1-\phi^2)} + \frac{\phi x_1^2}{\sigma^2} + \sum_{t=2}^{T} \frac{x_{t-1}(x_t - \phi x_{t-1})}{\sigma^2}\right)(1-\phi^2) && (47) \\
&\quad + \left[-\frac{2\phi}{(1-\phi^2)} + \frac{2(a-1)}{\phi+1} - \frac{2(b-1)}{1-\phi}\right](1-\phi^2) && (48)
\end{aligned}
$$

If we want to sample the parameters using RM-HMC, then we also need expressions for the metric tensor and its partial derivatives with respect to β, γ and α. We can obtain the following expressions for the individual components of the metric tensor

$$
\begin{aligned}
E\left\{\frac{\partial L}{\partial \beta}\frac{\partial L}{\partial \beta}\right\} &= \frac{2T}{\beta^2}, \quad E\left\{\frac{\partial L}{\partial \gamma}\frac{\partial L}{\partial \gamma}\right\} = T + 1 && (49) \\
E\left\{\frac{\partial L}{\partial \gamma}\frac{\partial L}{\partial \alpha}\right\} &= 2\phi, \quad E\left\{\frac{\partial L}{\partial \alpha}\frac{\partial L}{\partial \alpha}\right\} = \phi^2(3-T) + (T-1) && (50) \\
E\left\{\frac{\partial L}{\partial \beta}\frac{\partial L}{\partial \gamma}\right\} &= E\left\{\frac{\partial L}{\partial \gamma}\frac{\partial L}{\partial \beta}\right\} = E\left\{\frac{\partial L}{\partial \beta}\frac{\partial L}{\partial \alpha}\right\} = E\left\{\frac{\partial L}{\partial \alpha}\frac{\partial L}{\partial \beta}\right\} = 0 && (51)
\end{aligned}
$$

Thus the metric tensor may be written as

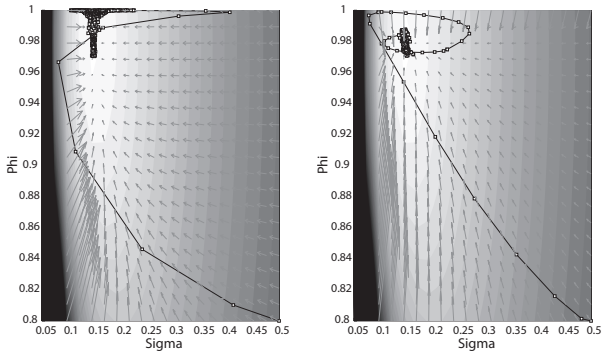

Fig. 1. The above contours were plotted from the stochastic volatility model investigated later in the paper. The latent volatilities and the parameter β are set to their true values, while the log-joint likelihood given different values of the parameters σ and ϕ is shown by the contour plot. The left hand plot shows the evolution of a Markov chain using HMC with a unit mass matrix, while the right hand plot shows the evolution of a chain from the same starting point using RM-HMC. Note how the use of the metric allows RM-HMC to converge much quicker to the target density.

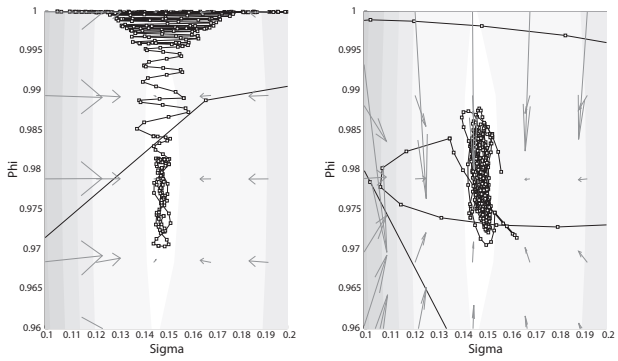

Fig. 2. Here we see a close-up of the Markov chain paths shown in Figure 1. It is clear that RM-HMC effectively normalises the gradients in each direction, whereas HMC, with a unit mass matrix, exhibits stronger gradients along the horizontal direction compared to the vertical direction, and therefore takes longer to explore the space fully. A carefully tuned mass matrix may improve HMC sampling, while RM-HMC deals with this automatically.

3.1. Symplectic Integration of a Non-Separable Hamiltonian on a Riemann Manifold

The dynamics of the non-separable Hamiltonian follow as

$$\frac{d\theta_i}{d\tau} = \frac{\partial H}{\partial p_i} = \left(\mathbf{G}(\boldsymbol{\theta})^{-1}\mathbf{p}\right)_i \quad (8)$$

$$\frac{dp_i}{d\tau} = -\frac{\partial H}{\partial \theta_i} = \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \theta_i} - \frac{1}{2}\mathsf{Tr}\left[\mathbf{G}(\boldsymbol{\theta})^{-1}\frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \theta_i}\right] + \frac{1}{2}\mathbf{p}^{\mathsf{T}}\mathbf{G}(\boldsymbol{\theta})^{-1}\frac{\partial \mathbf{G}(\boldsymbol{\theta})}{\partial \theta_i}\mathbf{G}(\boldsymbol{\theta})^{-1}\mathbf{p} \quad (9)$$

The Hamiltonian dynamics on the manifold are simulated by solving the continuous time derivatives and it is straightforward to see that they satisfy Liouville's theorem of volume preservation (Leimkuhler and Reich, 2004). However, for the discrete integrator it is not so straightforward. Naively employing the discrete Leapfrog integrator (equations (3), (4) and (5)), as in (Zlochin and Baram, 2001), gives transformations of the form $(\boldsymbol{\theta}, \mathbf{p}) \mapsto (\boldsymbol{\theta}, \mathbf{p} - \epsilon\varphi(\boldsymbol{\theta}, \mathbf{p}))$ and $(\boldsymbol{\theta}, \mathbf{p}) \mapsto (\boldsymbol{\theta} + \epsilon\phi(\boldsymbol{\theta}, \mathbf{p}), \mathbf{p})$, neither of which admits a Jacobian with unit determinant. In addition, it is straightforward to see that reversibility for $\boldsymbol{\theta}$ and $\mathbf{p}$ is not satisfied for finite step-size ϵ, as $\mathbf{G}(\boldsymbol{\theta}(\tau)) \neq \mathbf{G}(\boldsymbol{\theta}(\tau + \epsilon))$ and $\mathbf{p}(\tau)^{\mathsf{T}}\mathbf{F}(\boldsymbol{\theta})\mathbf{p}(\tau) \neq \mathbf{p}(\tau + \epsilon)^{\mathsf{T}}\mathbf{F}(\boldsymbol{\theta})\mathbf{p}(\tau + \epsilon)$. Therefore proposals generated from this integrator will not satisfy detailed balance in a Hybrid Monte Carlo scheme. What is required is a symplectic numerical integrator for solving this non-separable Hamiltonian to ensure a correct MCMC algorithm. Fully implicit integrators are available, however these require further numerical solution of the associated implicitly defined equations. A general purpose non-implicit symplectic integrator for non-separable Hamiltonians with position specific mass matrices, as defined by the Riemannian metric tensor, is desirable. We have developed such an integrator and the detailed derivation can be found in Appendix A. This resulting algorithm is employed in defining the Riemannian Manifold Hamiltonian Monte Carlo (RM-HMC) procedure below.

3.2. The Overall RM-HMC Algorithm

The overall proposal generating process, $(\mathbf{p}_0, \boldsymbol{\theta}_0) \mapsto (\mathbf{p}, \boldsymbol{\theta})$, is denoted by the vector function $(\mathbf{p}, \boldsymbol{\theta}) = \mathbf{f}(\mathbf{p}_0, \boldsymbol{\theta}_0, \epsilon, N_2)$ where $\mathbf{p}_0, \boldsymbol{\theta}_0$ are the current momentum and parameter values respectively, ϵ is the integration step size and N_2 is the number of iterations of step (12) below. We denote by N_1 the number of repeated applications of the vector function $\mathbf{f}$ to obtain a proposal. Scheme 1 for the full symplectic integrator has the following five steps

$$\mathbf{p}_1 = \mathbf{p}_0 - \epsilon\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}_0)/2 \quad (10)$$

$$\mathbf{p}_2 = \mathbf{g}(\boldsymbol{\theta}_0, \mathbf{p}_1, \epsilon/2) \quad (11)$$

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}_0 + \epsilon\mathbf{G}^{-1}(\boldsymbol{\theta}_0)\mathbf{p}_2, \quad (12)$$

$$\mathbf{p}_3 = \mathbf{g}(\boldsymbol{\theta}^*, \mathbf{p}_2, \epsilon/2) \quad (13)$$

$$\mathbf{p}^* = \mathbf{p}_3 - \epsilon\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}^*)/2 \quad (14)$$

where the vector function $\mathbf{g}(\boldsymbol{\theta}, \mathbf{p}, \epsilon/2)$ is defined in Appendix A.2.1. We note that this function requires the matrix of derivatives of the metric tensor with respect to each parameter. This may be derived analytically for a number of applications, as shall be demonstrated shortly. The repeated application of the function $(\mathbf{p}^*, \boldsymbol{\theta}^*) = \mathbf{f}(\mathbf{p}_0, \boldsymbol{\theta}_0, \epsilon, N_2)$ provides the means to obtain a deterministic proposal that is guided not only by the derivative information of the target density, as in HMC or MALA, but also exploits the local geometric structure of the manifold as determined by the

$$\exp(\epsilon\hat{A}) = \prod_{i=1}^{D-1}\exp(\epsilon\hat{A}_i/2)\exp(\epsilon\hat{A}_D)\prod_{j=D-1}^{1}\exp(\epsilon\hat{A}_j/2) + \mathcal{O}(\epsilon^3) \quad (43)$$

where the factorization is left-right symmetric guaranteeing time-reversibility. Now $\mathbf{p}^{\mathsf{T}}\mathbf{A}^k(\boldsymbol{\theta})\mathbf{p} = \alpha_k p_k^2 + \beta_k(\mathbf{p}_{-k})p_k + \gamma_k(\mathbf{p}_{-k})$, where $\mathbf{p}_{-k}$ denotes the vector with the k'th element removed and

$$\alpha_k = A^k_{kk}, \quad \beta_k(\mathbf{p}_{-k}) = 2\sum_{i\neq k}A^k_{ik}p_i, \quad \gamma_k(\mathbf{p}_{-k}) = \sum_{i,j\neq k}A^k_{ij}p_ip_j$$

allows each element of the operator to be written as

$$\exp(\epsilon\hat{A}_k) = \exp\left(\epsilon\left(\alpha_k p_k^2 + \beta_k(\mathbf{p}_{-k})p_k + \gamma_k(\mathbf{p}_{-k})\right)\frac{\partial}{\partial p_k}\right)$$

The overall factorisation for $\exp(\epsilon\hat{A})$ amounts to repeated calls to $\mathbf{p} = g(\mathbf{p}^*, \epsilon, \alpha, \beta, \gamma)$, (equations starting from 36), where $\mathbf{p}^*$ is the current momentum value. These calls must follow the sequential order of equation (43), such that the vector update $\mathbf{p}^* \mapsto \mathbf{p}$ is denoted by the vector function $\mathbf{p} = \mathbf{g}(\boldsymbol{\theta}, \mathbf{p}^*, \epsilon)$, which is defined as

$$\begin{aligned}
\hat{p}_k &= g\left(p^*_k, \epsilon/2, \alpha_k, \beta_k([\hat{\mathbf{p}}_{1:k-1}\ \ \mathbf{p^*}_{k+1:D}]), \gamma_d([\hat{\mathbf{p}}_{1:k-1}\ \ \mathbf{p^*}_{k+1:D}])\right) \quad \text{for } k = 1 \text{ to } (D-1)\\
p_D &= g\left(p^*_D, \epsilon, \alpha_D, \beta_D(\hat{\mathbf{p}}_{1:D-1}), \gamma_D(\hat{\mathbf{p}}_{1:D-1})\right)\\
p_k &= g\left(\hat{p}_k, \epsilon/2, \alpha_k, \beta_k([\hat{\mathbf{p}}_{1:k-1}\ \ \mathbf{p}_{k+1:D}]), \gamma_d([\hat{\mathbf{p}}_{1:k-1}\ \ \mathbf{p}_{k+1:D}])\right) \quad \text{for } k = (D-1) \text{ to } 1
\end{aligned}$$

where $\hat{\mathbf{p}}$ is an intermediate variable used to implement the sequential updating of the momentum.

A.2.2. Overall Symplectic Integrator in Multi-Dimensional Case

Finally the overall time-reversible symplectic evolution operator can be obtained to second-order by the following splitting of the non-separable Hamiltonian

$$\exp(\epsilon\hat{F}/2)\exp(\epsilon\hat{A}/2)\exp(\epsilon\hat{T})\exp(\epsilon\hat{A}/2)\exp(\epsilon\hat{F}/2)$$

which yields the overall operator $(\mathbf{p}_0, \boldsymbol{\theta}_0) \to (\mathbf{p}, \boldsymbol{\theta})$ where $\mathbf{p}_0, \boldsymbol{\theta}_0$ are the current momentum and parameter values respectively. This is now denoted as the vector function $(\mathbf{p}, \boldsymbol{\theta}) = \mathbf{f}(\mathbf{p}_0, \boldsymbol{\theta}_0, \epsilon)$

$$\begin{aligned}
\mathbf{p}_1 &= \mathbf{p}_0 + (\epsilon/2)\boldsymbol{\varphi}(\boldsymbol{\theta}_0)\\
\mathbf{p}_2 &= \mathbf{g}(\boldsymbol{\theta}_0, \mathbf{p}_1, \epsilon/2)\\
\boldsymbol{\theta} &= \boldsymbol{\theta}_0 + \epsilon\mathbf{G}^{-1}(\boldsymbol{\theta}_0)\mathbf{p}_2\\
\mathbf{p}_3 &= \mathbf{g}(\boldsymbol{\theta}, \mathbf{p}_2, \epsilon/2)\\
\mathbf{p} &= \mathbf{p}_3 + (\epsilon/2)\boldsymbol{\varphi}(\boldsymbol{\theta})
\end{aligned}$$

where the third step may be iterated if required. This is referred to as Scheme 1. An alternative split is possible yielding the alternative Scheme 2 given below.

a second order factorisation of $\exp(\epsilon\hat{V})$ follows as

$$\exp(\epsilon\hat{V}) = \exp\left(\frac{1}{2}\epsilon\gamma\frac{\partial}{\partial p}\right)\exp\left(\frac{1}{2}\epsilon\alpha p^2\frac{\partial}{\partial p}\right)\exp\left(\epsilon\beta p\frac{\partial}{\partial p}\right)\exp\left(\frac{1}{2}\epsilon\alpha p^2\frac{\partial}{\partial p}\right)\exp\left(\frac{1}{2}\epsilon\gamma\frac{\partial}{\partial p}\right)$$

from which the following general algorithm emerges

$$\begin{aligned} p_1 &= p_0 + \epsilon\gamma/2 && (36)\\ p_2 &= \frac{p_1}{1-\epsilon\alpha p_1/2} && (37)\\ p_3 &= \exp(\epsilon\beta)p_2 && (38)\\ p_4 &= \frac{p_3}{1-\epsilon\alpha p_3/2} && (39)\\ p &= p_4 + \epsilon\gamma/2 && (40)\end{aligned}$$

which will be referred to as $p = g(p_0, \epsilon, \alpha, \beta, \gamma)$. Now the full multi-dimensional case will be considered in developing the eventual symplectic integrator required for RM-HMC.

A.2. Multi-Dimensional Case

In this case the Hamiltonian equations of motion follow as

$$\frac{\partial\theta_k}{\partial t} = \frac{\partial H}{\partial p_k} = \left(\mathbf{G}(\boldsymbol{\theta})^{-1}\mathbf{p}\right)_k, \qquad \frac{\partial p_k}{\partial t} = -\frac{\partial H}{\partial\theta_k} = \varphi_k(\boldsymbol{\theta}) + \mathbf{p}^\mathsf{T}\mathbf{A}^k(\boldsymbol{\theta})\mathbf{p} \tag{41}$$

where $\varphi_k(\boldsymbol{\theta}) = -\partial\phi(\boldsymbol{\theta})/\partial\theta_k$, $\mathbf{A}^k(\boldsymbol{\theta}) = -\frac{1}{2}\partial\mathbf{G}(\boldsymbol{\theta})^{-1}/\partial\theta_k$, and $(\cdot)_k$ denotes the k^{th} element of the corresponding vector. The Poisson operators are then defined as

$$\hat{T} = \left(\mathbf{p}\mathbf{G}(\boldsymbol{\theta})^{-1}\right)_k\frac{\partial}{\partial\theta_k} \quad \hat{V} = \left(\varphi_k(\boldsymbol{\theta}) + \mathbf{p}^\mathsf{T}\mathbf{A}^k(\boldsymbol{\theta})\mathbf{p}\right)\frac{\partial}{\partial p_k} \equiv \hat{F} + \hat{A}$$

As in the one dimensional case we define $Q_i(\boldsymbol{\theta}) = \sum_j\int g_{ij}(\boldsymbol{\theta})d\theta_j$, where g_{ij} denotes the ij'th element of the metric tensor $\mathbf{G}(\boldsymbol{\theta})$ then it follows that $\partial/\partial\theta_k = (\partial Q_l/\partial\theta_k)(\partial/\partial Q_l) = \sum_l g_{kl}(\boldsymbol{\theta})\partial/\partial Q_l$, thus simplifying the $\hat{T}$ operator to $\hat{T} = p_k\partial/\partial Q_k$. The metric tensor is symmetric so

$$\exp(\epsilon\hat{T})\mathbf{Q} = (1 + \epsilon\mathbf{p}\cdot\partial/\partial\mathbf{Q} + \frac{1}{2}(\epsilon\mathbf{p}\cdot\partial/\partial\mathbf{Q})^2 + \cdots)\mathbf{Q} = \mathbf{Q} + \epsilon\mathbf{p} \tag{42}$$

where $\cdot$ denotes the dot product. The update $\boldsymbol{\theta}_0 \rightarrow \boldsymbol{\theta}$ follows via $\mathbf{Q}(\boldsymbol{\theta}) = \mathbf{Q}(\boldsymbol{\theta}_0) + \epsilon\mathbf{p}$. Note this system of equations can be solved via a Newton iteration $\boldsymbol{\theta} = \boldsymbol{\theta}_0 + (\partial\mathbf{Q}(\boldsymbol{\theta}_0)/\partial\boldsymbol{\theta}_0)^{-1}\epsilon\mathbf{p} = \boldsymbol{\theta}_0 + \epsilon\mathbf{G}^{-1}(\boldsymbol{\theta}_0)\mathbf{p}$ as in the uni-dimensional case. The operator $\exp(\epsilon\hat{F})$ is simply a forward time shift of the momentum $\mathbf{p}$ and as such $\mathbf{p} = \mathbf{p}_0 + \epsilon\boldsymbol{\varphi}(\boldsymbol{\theta}_0)$.

A.2.1. Implementing the $\exp(\epsilon\hat{F})$ Operator

Splitting the operator $\exp(\epsilon\hat{A})$ such that $\hat{A} = \sum_{i=1}^D\hat{A}_i$ where D is the dimension of the vector space and each $\hat{A}_i = \mathbf{p}^\mathsf{T}\mathbf{A}^i(\boldsymbol{\theta})\mathbf{p}\frac{\partial}{\partial p_i}$ then to second order

metric tensor. Intuitively, comparing (4) and (12) shows that the constant mass matrix $\mathbf{M}$, defining a globally constant metric, is now replaced with the position specific metric thus removing the requirement to tune the values of the elements of $\mathbf{M}$, which so dramatically affects the performance of HMC. Since the integration scheme detailed above is symplectic, it satisfies the requirements of both volume preservation and reversibility. Thus employing this as a proposal process provides a correct MCMC scheme satisfying detailed balance and convergence to the desired target density. It should be noted that this symplectic integrator is a generalisation of the Leapfrog method; if the metric tensor is globally constant and independent of position then equations (10) to (14) reduce exactly to the leapfrog equations (3) to (5). Pseudo-code describing the overall RM-HMC sampling scheme is described below.

Algorithm 1 Overall RM-HMC Sampling Scheme

Require: ϵ, $N_{samples}$, N_1, N_2, $\boldsymbol{\theta}(0)$, $\mathbf{p}(0)$
  for $i = 1$ to $N_{samples}$ **do**
    $\mathbf{p} \sim \mathcal{N}(\mathbf{0}, \mathbf{G}(\boldsymbol{\theta}(i-1)))$
    $(\mathbf{p}^*(0), \boldsymbol{\theta}^*(0)) \leftarrow (\mathbf{p}, \boldsymbol{\theta}(i-1))$
    for $j = 1$ to N_1 **do**
      $(\mathbf{p}^*(j), \boldsymbol{\theta}^*(j)) \leftarrow \mathbf{f}(\mathbf{p}^*(j-1), \boldsymbol{\theta}^*(j-1), \epsilon, N_2)$
    end for
    $(\mathbf{p}^*, \boldsymbol{\theta}^*) \leftarrow (\mathbf{p}(N_{steps}), \boldsymbol{\theta}(N_{steps}))$
    draw $\alpha \sim Uniform(0,1)$
    if $\alpha < \min\{1, \exp(-H(\boldsymbol{\theta}^*, \mathbf{p}^*) + H(\boldsymbol{\theta}(i-1), \mathbf{p}))\}$ **then**
      $\boldsymbol{\theta}(i) \leftarrow \boldsymbol{\theta}^*$
    else
      $\boldsymbol{\theta}(i) \leftarrow \boldsymbol{\theta}(i-1)$
    end if
  end for

Figures 1 and 2 provide an intuitive visual demonstration of the differences in HMC and RM-HMC when converging to and sampling from a target density. Matlab code for RM-HMC is available at `http://www.dcs.gla.ac.uk/inference/rmhmc`. To illustrate the RM-HMC sampling scheme and evaluate performance against alternative MCMC methods, a number of example applications are presented. We begin with posterior sampling for Logistic Regression models.

4. RM-HMC for Bayesian Logistic Regression

Consider an $N \times D$ design matrix $\mathbf{X}$ comprising N samples each with D covariates and a binary response variable $\mathbf{t} \in \{0,1\}^N$. Denoting the logistic link function as $\sigma(\cdot)$, a Bayesian logistic regression model of the binary response (Gelman *et al.*, 2004; Liu, 2001) is obtained by the introduction of regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^D$ with an appropriate prior, which for illustrative purposes is given as $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \alpha\mathbf{I})$ where α is given. Neglecting constants, the log joint-likelihood follows in standard form as

$$\mathcal{L}(\boldsymbol{\beta}) - \frac{1}{2\alpha}\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\beta} = \boldsymbol{\beta}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{t} - \sum_{n=1}^N\log(1 + \exp(\boldsymbol{\beta}^\mathsf{T}\mathbf{X}_{n,\cdot}^\mathsf{T})) - \frac{1}{2\alpha}\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\beta} \tag{15}$$

where $\mathbf{X}_{n,\cdot}$ denotes the vector that is the n^{th} row of the $N \times D$ matrix $\mathbf{X}$. The derivative of the log joint-likelihood is $\nabla\mathcal{L}(\boldsymbol{\beta}) - \alpha^{-1}\boldsymbol{\beta}$ with Fisher Information $E\{\nabla\mathcal{L}(\boldsymbol{\beta})\nabla\mathcal{L}(\boldsymbol{\beta})^\mathsf{T}\} = \mathbf{X}^\mathsf{T}\boldsymbol{\Lambda}\mathbf{X}$.

Table 1. Summary of datasets for logistic regression

Name	Covariates (D)	Data Points (N)	Dimension of $\boldsymbol{\beta}$ (b)
Pima Indian	7	532	8
Australian Credit	14	690	15
German Credit	24	1000	25
Heart	13	270	14
Ripley	2	250	7

The diagonal $N \times N$ matrix $\boldsymbol{\Lambda}$ has elements $\boldsymbol{\Lambda}_{n,n} = \sigma(\boldsymbol{\beta}^{\mathsf{T}}\mathbf{X}_{n,\cdot}^{\mathsf{T}})(1-\sigma(\boldsymbol{\beta}^{\mathsf{T}}\mathbf{X}_{n,\cdot}^{\mathsf{T}}))$. To capture prior informativeness in the metric tensor we follow (Tsutakawa, 1972; Ferreira, 1981) and sum the Fisher Information with the prior precision to define the overall metric tensor for the model as $\mathbf{G}(\boldsymbol{\beta}) = \mathbf{X}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{X} + \alpha^{-1}\mathbf{I}$ and finally the derivative matrices of the metric tensor take the form $\partial\mathbf{G}(\boldsymbol{\beta})/\partial\beta_i = \mathbf{X}^{\mathsf{T}}\boldsymbol{\Lambda}\mathbf{V}^i\mathbf{X}$ where the $N\times N$ diagonal matrix $\mathbf{V}^i$ has elements $(1-2\sigma(\boldsymbol{\beta}^{\mathsf{T}}\mathbf{X}_{n,\cdot}^{\mathsf{T}}))X_{ni}$. The above identities are all that are required to define both HMC and RM-HMC sampling methods, which will be illustrated in the following experimental section.

4.1. Experimental Results for Bayesian Logistic Regression

We present results from the analysis of 5 datasets (Michie *et al.*, 1994; Ripley, 1996), summarised in Table 1. These datasets exhibit a wide range of characteristics which provides a challenging test for any applied sampling method; the number of covariates ranges from 2 to 24, the number of data points ranges from 250 to 1000, and the standard deviations of the induced marginal posterior distributions range from 0.0004 to 3. We investigate the use of RM-HMC applied to this problem and also implement the following sampling methods for comparison:-

(a) Component-Wise Adaptive Metropolis-Hastings (Robert, 2004)
(b) Joint Updating Gibbs Sampler (Holmes and Held, 2005)
(c) Metropolis Adjusted Langevin Algorithm (Roberts and Stramer, 2003)
(d) Hybrid Monte Carlo (Duane *et al.*, 1987; Neal, 1993a; Liu, 2001)

Given each dataset we wish to sample from the posterior distribution over the regression coefficients $\boldsymbol{\beta}$, and in each experiment wide Gaussian prior distributions were employed such that $\pi(\beta_i) \sim \mathcal{N}(0, 100)$. A linear logistic regression model with intercept was used for each of the datasets with the exception of the Ripley dataset, for which a cubic polynomial regression model was employed.

Each method was run 10 times with every dataset and the average results were recorded. We reproduce the results of Holmes and Held (2005) by allowing 5000 burn in iterations so that each sampler reaches the stationary distribution and has time to adapt as necessary. The next 5000 iterations were used to collect posterior samples for each of the methods and the CPU time required to collect these samples was recorded. Each method was implemented in the interpreted language Matlab to ensure fair comparison. We compared the relative efficiency of these methods by calculating the effective sample size (ESS) using the posterior samples for each covariate, $ESS = N(1+2\sum_k \gamma(k))^{-1}$ where N is the number of posterior samples and $\sum_k \gamma(k)$ is the sum of the K monotone sample autocorrelations as estimated by the initial monotone sequence estimator (see Geyer (1992)). The standard error around the mean ESS was less than 2×10^{-2} for all results. Such an approach was also taken by Holmes and Held (2005), in which they report the *mean* ESS, averaged over each of the covariates. However, we feel this could give a rather inflated measure of

provided that the effect of each component operator can be computed exactly and coefficients t_i, v_i are determined by the necessary order. For example

$$\begin{aligned}
\exp\left(\epsilon\hat{V}\right)\theta &= \left(1+\left(\alpha(\theta)p^2\frac{\partial}{\partial p}\right)+\frac{1}{2}\left(\alpha(\theta)p^2\frac{\partial}{\partial p}\right)^2+\frac{1}{3!}\left(\alpha(\theta)p^2\frac{\partial}{\partial p}\right)^3+\cdots\right)\theta \\
&= \theta \\
\exp\left(\epsilon\hat{V}\right)p &= \left(1+\left(\alpha(\theta)p^2\frac{\partial}{\partial p}\right)+\frac{1}{2}\left(\alpha(\theta)p^2\frac{\partial}{\partial p}\right)^2+\frac{1}{3!}\left(\alpha(\theta)p^2\frac{\partial}{\partial p}\right)^3+\cdots\right)p \\
&= \frac{p}{1-\epsilon\alpha(\theta)p}
\end{aligned}$$

where the latter corresponds to the integration of dp/dt while holding θ constant. Now considering the $\exp(\epsilon\hat{T})$ operator, it is straightforward to see that $\exp(\epsilon\hat{T})p = p$. As $\hat{T} = pg(\theta)^{-1}\frac{\partial}{\partial\theta}$, the polynomial expansion will act repeatedly on $g(\theta)^{-1}$ and an analytic closed form solution will not emerge. This issue can be circumvented however by introducing a change of variable from $\theta \mapsto q(\theta)$ such that $g(\theta)\partial\theta = \partial q \implies q = \int g(\theta)d\theta$. The $\hat{T}$ operator now simplifies to $\hat{T} = p\frac{\partial}{\partial q}$ and one has exactly

$$\exp(\epsilon\hat{T})q = \left(1+\epsilon p\frac{\partial}{\partial q}+\frac{1}{2}\left(\epsilon p\frac{\partial}{\partial q}\right)^2+\cdots\right)q = q+\epsilon p \tag{30}$$

We are now in a position to define a second order symplectic integrator by employing the following factorisation,

$$\exp\left(\epsilon\left(\hat{T}+\hat{V}\right)\right) \approx \exp\left(\epsilon\hat{V}/2\right)\exp\left(\epsilon\hat{T}\right)\exp\left(\epsilon\hat{V}/2\right) \tag{31}$$

Gathering the expressions for each of the two operators acting on both θ and p yields the following algorithm to provide the mapping $(\theta_0, p_0) \mapsto (\theta, p)$, i.e in terms of time evolution $(\theta_t, p_t) \mapsto (\theta_{t+\epsilon}, p_{t+\epsilon})$.

$$p_1 = \frac{p_0}{1-\frac{1}{2}\epsilon\alpha(\theta_0)p_0} \tag{32}$$

$$q(\theta) = q(\theta_0)+\epsilon p_1 \tag{33}$$

$$p = \frac{p_1}{1-\frac{1}{2}\epsilon\alpha(\theta)p_1} \tag{34}$$

To obtain θ from equation (33), one can regard it as the root of $f(\theta) = q(\theta) - q(\theta_0) - \epsilon p_1 = 0$ and solve for it via a Newton iteration, $\theta = \theta_0 - f(\theta_0)/f'(\theta_0) = \theta_0 + g(\theta_0)^{-1}\epsilon p_1$, repeating with $\theta \mapsto \theta_0$ as necessary for convergence. Whilst this step of the integrator is defined iteratively it should be stressed that the overall symplectic integrator that is presented in this paper is, computationally, an enormous improvement to fully implicit methods such as the Generalised Leapfrog, see e.g. Leimkuhler and Reich (2004).

Prior to considering the full multivariate case a generalisation of the operator $\hat{V}$ is presented as

$$\hat{V} = \left(\alpha(\theta)p^2+\beta(\theta)p+\gamma(\theta)\right)\frac{\partial}{\partial p} \tag{35}$$

Now since the following equalities hold,

$$\exp\left(\epsilon\alpha p^2\frac{\partial}{\partial p}\right)p = \frac{p}{1-\epsilon\alpha p}, \quad \exp\left(\epsilon\beta p\frac{\partial}{\partial p}\right)p = \exp(\epsilon\beta)p, \quad \exp\left(\epsilon\gamma\frac{\partial}{\partial p}\right)p = p+\epsilon\gamma$$

Schervish. M.J. (1995) *Theory of Statistics*, Springer. New York.

Tsutakawa. R.K. (1972) Design of Experiment for Bioassay *Journal of the American Statistical Association*. 67(339). pp 584–590.

Rasmussen. C.E and Williams. C.K.I (2006) *Gaussian Processes for Machine Learning*, The MIT Press.

Vyshemirsky. V. and Girolami. M. (2008) Bayesian Ranking of Biochemical System Models, *Bioinformatics* 24, (2008), pp 833–839.

Zlochin. M. and Baram. Y. (2001) Manifold Stochastic Dynamics for Bayesian Learning. *Neural Computation*, 13. pp 2549–2572.

A. Symplectic Integrator for Non-Seperable Hamiltonians

To develop the Riemannian Manifold Hamiltonian Monte Carlo (RM-HMC) method an non-implicit symplectic integrator is required for non-separable Hamiltonians of the form

$$H(\boldsymbol{\theta}, \mathbf{p}) = \phi(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^{\mathsf{T}}\mathbf{G}(\boldsymbol{\theta})^{-1}\mathbf{p} \tag{24}$$

A.1. One Dimensional Case

To clearly illustrate the manner in which the symplectic integrator is derived the simplest case of univariate random variables is considered where $\phi(\theta) = 0$ and the Hamiltonian is

$$H(\theta, p) = \frac{1}{2}\frac{p^2}{g(\theta)} \tag{25}$$

with associated equations of motion

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial p} = \frac{p}{g(\theta)}, \quad \frac{dp}{dt} = -\frac{\partial H}{\partial \theta} = \left(-\frac{1}{2}\frac{\partial g(\theta)^{-1}}{\partial \theta}\right)p^2 = \alpha(\theta)p^2 \tag{26}$$

Introducing a general dynamical variable $W(\theta, p)$ whose time evolution is determined by the above equations of motion then the following can be considered as a Poisson bracket operator equation for $W(\theta, p)$ (Hairer *et al.*, 2002)

$$\frac{dW}{dt} = \frac{\partial W}{\partial \theta}\frac{d\theta}{dt} + \frac{\partial W}{\partial p}\frac{dp}{dt} = \left(pg(\theta)^{-1}\frac{d}{d\theta} + \alpha(\theta)p^2\frac{d}{dp}\right)W(\theta, p) = (\hat{T} + \hat{V})W(\theta, p) \tag{27}$$

with solution

$$W_{t+\epsilon} = \exp\left(\epsilon\left(\hat{T} + \hat{V}\right)\right)W_t \tag{28}$$

Symplectic integrators are derived by approximating the short time evolution operator in product form such that

$$\exp\left(\epsilon\left(\hat{T} + \hat{V}\right)\right) \approx \prod_{i=1}^{N_o}\exp\left(t_i\epsilon\hat{T}\right)\exp\left(v_i\epsilon\hat{V}\right) \tag{29}$$

the true ESS, since ideally we want a measure of the number of samples which are uncorrelated over *all* covariates. In this paper we therefore report the *minimum* ESS of the sampled covariates. This minimum ESS is then normalised relative to the CPU time by calculating the time taken to obtain 1 sample which is effectively uncorrelated across all covariates.

4.2. Comparative MCMC Sampling Methods

We employed an adaptive Metropolis-Hastings (M-H) scheme, such that each covariate was updated individually with its stepsize being adapted in every 100 iterations during burn-in to achieve an acceptance rate of between 20% and 40%. The stepsize was then fixed when sampling from the posterior distribution.

The auxiliary variable Gibbs sampler of Holmes and Held (2005) was implemented with a joint update of $\{\mathbf{z}, \boldsymbol{\beta}\}$, where $\mathbf{z} \in \mathbb{R}^N$ is the auxiliary variable designed to improve mixing of the covariate samples. We implemented the algorithm based on the very detailed pseudo-code given in the appendix of their paper, and in contrast to the M-H algorithm this method has the advantage of requiring no tuning of parameters. The main computational expense however is in the repeated sampling from truncated normal distributions, for which we implemented code based on the efficient method defined in Johnson *et al.* (1999).

We implemented a MALA sampler with proposed covariates being drawn from the multivariate normal distribution $\mathcal{N}(\boldsymbol{\beta} + \nabla\log(\pi\{\boldsymbol{\beta}\})h/2, h\mathbf{I}_D)$, where $\mathbf{I}_D$ is the D-dimensional identity matrix and h controls the scaling of the proposal variance. We follow the advice of Roberts and Rosenthal (1998) by scaling h like $O(D^{-\frac{1}{3}})$, where D is the number of covariates, such that we achieve an acceptance rate of between 40% and 60%.

Hybrid Monte Carlo has promised to offer more efficient sampling from high dimensional probability distributions by effectively reducing the amount of random walk present in the parameter values being proposed. This has indeed been shown to be the case for relatively simple, although high-dimensional, multivariate normal distributions, however there has been little application to more complex data models. We believe the reason for this lies in the amount of tuning required to obtain reasonable mixing and rates of acceptance, as will be highlighted in the following section. The two main parameters which require tuning are the number of leapfrog steps, N_1, and the size of each leapfrog step, ϵ. It has been suggested that choosing the leapfrog stepsize to be inversely proportional to the marginal standard deviation of the target distribution along each dimension drastically improves mixing, particularly when such marginals are of greatly varying orders of magnitude. Setting different leapfrog stepsizes along different directions can be equivalently encoded in the so-called mass matrix (Neal, 1993a, 1996).

This approach clearly requires advance knowledge of the distribution being sampled from, and in a practical setting this information is very rarely available. The use of exploratory runs of a Metropolis sampler to obtain initial estimates of the target distribution has been suggested (Hajian, 2007), however there is the obvious associated computational cost and the fact that this may not be feasible for very complex distributions.

Following the advice of Neal (1993a, 1996), we fix the size of each leapfrog step ϵ to a value slightly smaller than the smallest marginal standard deviation of the model parameter posteriors, and set the number of leapfrog steps N_1 such that the maximum distance that can be travelled in a single move, ϵN_1, is larger than the largest standard deviation of the marginal parameter distributions. A larger step size would result in large rejection rates, while a smaller number of steps would result in very slow exploration of the target distribution.

In our experiments we assume this information is known when implementing HMC, presumably after a number of exploratory runs of the algorithm, and set ϵ small enough to obtain a high

Table 2. RM-HMC with integration scheme 1 - investigating the effect of parameter settings on sampling efficiency

ϵN_1	Max ϵ	N_1	N_2	Mean Time (s)	Min ESS	s/Min ESS
1	1/5	5	1	385.8	381	1.01
1	1/5	5	2	393.2	350	1.12
2	1/10	20	1	1332.5	1555	0.86
2	1/10	20	2	1344.2	1525	0.88
3	1/10	30	1	1949.7	2795	0.70
3	1/10	30	2	1983.3	2389	0.83

Table 3. RM-HMC with integration scheme 2 - investigating the effect of parameter settings on sampling efficiency

ϵN_1	Max ϵ	N_1	N_2	Mean Time (s)	Min ESS	s/Min ESS
1	1/10	10	1	378.9	278	1.36
1	1/10	10	2	387.9	239	1.62
2	1/10	20	1	702.2	774	0.91
2	1/10	20	2	732.6	457	1.60
3	1/15	45	1	1567.9	1624	0.97
3	1/15	45	2	1580.3	894	1.77

acceptance rate ($> 70\%$) and $\epsilon N_1 \approx 3$ allowing the chain to traverse a distance larger than the standard deviation of the largest marginal posterior for all datasets, see Table 9. This approach works well for distributions in which the marginal standard deviations are of a similar magnitude, however the algorithm soon becomes computationally very expensive to run in situations where they greatly differ and the number of leapfrog steps required for adequate mixing consequently becomes very large.

4.3. Investigating RM-HMC

We begin by investigating the RM-HMC method in detail for the most challenging of our five datasets, German Credit, which consists of 24 covariates and 1000 datapoints. We then compare the results for all five datasets employing the alternative sampling methods described in the previous section.

As previously mentioned, the evolution operator of the RM-HMC method may be obtained to second order by splitting the non-separable Hamiltonian (see Appendix A), and Scheme 1 has already been presented. The alternative way of splitting the Hamiltonian as detailed in the appendix yields a slightly different but equally valid integration scheme, Scheme 2. The main computational burden of both integration schemes is incurred in calling the function $\mathbf{p} = \mathbf{g}(\boldsymbol{\theta}, \mathbf{p}_0, \epsilon)$. Scheme 1 calls the function $\mathbf{g}$ twice per iteration with stepsize $\epsilon/2$, whereas Scheme 2 calls $\mathbf{g}$ only once per iteration, but with larger stepsize, ϵ. We investigate both integration schemes to determine which is computationally more efficient in terms of time taken per (effectively) independent sample, calculated as Time/(Min ESS). The three parameters that may be altered in the RM-HMC algorithm are the integration stepsize, ϵ, the number of steps per integration, N_1 and the number of inner steps per integration, N_2 which update $\boldsymbol{\theta}$. The maximum total distance which a chain may travel in a single proposed move is given by ϵN_1, and for any given value of ϵN_1 we chose ϵ small enough such that the acceptance ratio was above 70% and then adjusted N_1 appropriately. Tables 2 and 3 show the results of the two schemes using a variety of choices for these parameters, which allow us to make the following observations.

Holmes. C.C. and Held. L. (2005). Bayesian Auxiliary Variable Models for Binary and Multinomial Regression, *Bayesian Analysis*, 1(1), pp. 145–168.

Ishwaran. H. (1999) Applications of Hybrid Monte Carlo to Bayesian Generalised Linear Models: Quasicomplete Separation and Neural Networks. *Journal of Computational and Graphical Statistics*. 8, pp 779 – 799.

Johnson. V. E. Krantz. S. G. and Albert. J. H. (1999) *Ordinal Data Modeling*. Springer Verlag.

Kass. R.E. (1989) The Geometry of Asymptotic Inference. *Statistical Science*. 4(3), pp 188–234.

Lambert. P. and Eilers. P.H.C. (2009) Bayesian Density Estimation from Grouped Continuous Data *Computational Statistics and Data Analysis*. 53(4), pp 1388–1399.

Leimkuhler. B. and Reich. S. (2004) *Simulating Hamiltonian Dynamics*, Cambridge University Press.

Liu. J.S. (2001) *Monte Carlo Strategies in Scientific Computing*. New York: Springer

Metropolis. M. Rosenbluth. A.W. Rosenbluth. M.N. Teller.A.H. and Teller. E. (1953) Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*. 21, pp 1087–1092.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*. Prentice Hall, Englewood Cliffs, N.J.

Murray. M.K. and Rice. J.W. (1993) *Differential Geometry and Statistics* Chapman and Hall, CRC.

Neal. R.M. (1993a). *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Technical. Report, University of Toronto, Canada.

Neal. R.M. (1996). *Bayesian Learing for Neural Networks*. Springer, Lecture Notes in Statistics, New York.

Neal. R.M. (1993b) Bayesian Learning via Stochastic Dynamics. *Advances in Neural Information Processing Systems*, 5. pp. 475–482.

Ramsay. J. Hooker. G. Campbell. D. J. and Cao. J. (2007) Parameter Estimation for Differential Equations: A Generalized Smoothing Approach, *Journal of the Royal Statistical Society: Series B*, 69 (5). pp 741–796.

Rao. C. R. Information and Accuracy Attainable in the Estimation of Statistical Parameters. *Bulletin of the Calcutta Mathematical Society*. 37. pp 81 – 91.

Ripley. B. D. (1996) *Pattern Recognition and Neural Networks*. Cambridge University press.

Robert. C. (2004). *Monte Carlo Statistical Methods*. Springer Verlag.

Roberts. G. and Rosenthal. J. S. (1998) Optimal Scaling of Discrete Approximations to Langevin Diffusions. *Journal of Royal Statistical Society, B*. 60. pp 255 –268.

Roberts. G. and Stramer. O. (2003) Langevin Diffusions and Metropolis-Hastings Algorithms. *Methodology and Computing in Applied Probability*. 4. pp 337–358.

9. Acknowledgements

M. Girolami is supported by an Engineering and Physical Sciences Research Council (EPSRC) Advanced Research Fellowship EP/E052029/1, and the Biotechnology and Biological Sciences Research Council (BBSRC) project grant BB/G006997/1. B. Calderhead is supported by a Microsoft Research European PhD Scholarship.

References

Amari. S. (1990). *Differential-Geometrical Methods in Statistics*. Lecture Notes in Statistics 28. Springer Verlag.

Amari. S. (1997). Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10, pp. 251–276.

Andrieu. C. and Thoms. J. (2008). A Tutorial on Adaptive MCMC. *Statistics and Computing*, 18, pp. 343–373.

Beichl. I. and Sullivan. F. (2000). The Metropolis Algorithm. *Computing in Science and Engineering*. 2(1). pp 65–69.

Calderhead. B. Girolami. M and Lawrence. N. D. (2009). Accelerating Bayesian Inference over Nonlinear Differential Equations with Gaussian Processes, *22nd Conference on Neural Information Processing Systems*. MIT Press.

Chavel. I. (1993). *Riemannian Geometry: A Modern Introduction*. Cambridge University Press.

Christensen. O.F. Roberts. G.O. and Rosenthal. J.S. (2005). Scaling Limits for the Transient Phase of Local Metropolis-Hastings Algorithms. *Journal of the Royal Statistical Society: Series B.* 67(2), pp. 253–268.

Duane. S. Kennedy. A. D. Pendleton. B. J. and Roweth. D. (1987) Hybrid Monte Carlo, *Physics Letters. B.*, 55, pp. 2774–2777.

Ferreira. P.E. (1981). Extending Fisher's Measure of Information. *Biometrika*. 68(3), pp. 695-698.

Gelman. A. Carlin. J.B. Stern. H.S. and Rubin. D.B. (2004) *Bayesian Data Analysis*, Chapman & Hall.

Geyer. C. J. (1992) Practical Markov Chain Monte Carlo. *Statistical Science*. 7. pp 473 – 483.

Gilks. W.R. Richardson. S. and Spiegelhalter. D.J. (1996) *Markov Chain Monte Carlo in Practice*. CRC Press.

Gustafson. P. (1997) Large Hierarchical Bayesian Analysis of Multivariate Survival Data. *Biometrics*. 53, pp 230 – 242.

Hairer. E. Lubich. C. and Wanner. G. (2002) *Geometric Numerical Integration*, Springer-Verlag.

Hajian. A. (2007) Efficient Cosmological Parameter Estimation with Hamiltonian Monte Carlo Technique. *Phys. Rev. D.* 75. 083525 – 1– 11.

Hastings. W.K. (1970) Monte Carlo Sampling Methods using Markov Chains and their Applications. *Biometrika* 57, pp 97-109.

Firstly, it is clear that using more than 1 inner step provides very little improvement in sampling efficiency, and may even be detrimental in some instances (we conjecture this is due to overshooting, which commonly occurs when the log-density is locally non-quadratic). Secondly, we found that sampling generally became more efficient as the maximum total distance travelled by a chain, ϵN_1, was increased, i.e. when the chain was able to traverse a distance greater than the width of each marginal distribution. Note that Scheme 2 required a smaller step size for $\epsilon N_1 = 3$, which impacted negatively on efficiency. Thirdly, using Scheme 1 it was sometimes possible to perform the integration using a larger step size, which meant that a smaller number of integration steps was required to move the chain a set distance, compared to Scheme 2 requiring a smaller stepsize and larger number of iterations. Therefore, even though Scheme 1 requires two calls to the function **g** per iteration, it appears to be computationally more efficient per independent sample than Scheme 2. This efficiency is likely also a result of the integration scheme computing the rate of change of the metric tensor more accurately, since it calls the function **g** twice per iteration but with half a stepsize.

4.4. Comparison of RM-HMC

Following the guidelines given in the previous section, we find that the RM-HMC sampling method works very well for a variety of datasets and is fairly robust to the choice of algorithm parameters. For comparison with the alternative sampling methods, we chose the settings for RM-HMC based on the above analysis. We employed Scheme 1 with 1 inner step, setting ϵ for each dataset equal to the smallest stepsize for which the acceptance rate was reasonably high ($> 70\%$), and the number of integration steps such that $\epsilon N_1 \approx 3$. We repeated the sampling experiments 10 times and averaged the results, which are shown for each of the datasets in Tables 4 to 8. It is interesting to see that MALA generally performs poorly. Whereas all other methods converge within 5000 burn-in iterations for all datasets, MALA needs as many as 2 million iterations to converge due to the very small stepsize required to achieve an acceptance ratio above 40%. This is particularly the case for the Australian Credit and Heart datasets, which exhibit very large differences in scale between the largest and smallest marginal standard deviations (see Table 9), resulting in extremely slow exploration of the target distribution, indeed even after 2 million iterations the Langevin guided chains had still not reached their stationary distributions. Clearly some method of scaling the regression coefficients would improve the mixing, however this is again unfeasible unless information regarding the marginal posterior distributions is known in advance. Similarly the standard HMC method fails to converge for the Australian Credit dataset, since the stepsize is so small that the number of integration steps required becomes computationally impractical to implement. Figure 3 shows the trace and autocorrelation plots for 1000 posterior samples using the Heart dataset. The difference in autocorrelation is quite striking, both from inspection of the traces and from examination of the autocorrelation plots themselves. The autocorrelation of the RM-HMC samples drop towards zero far quicker than for any of the other methods.

In our simulations, RM-HMC outperforms all of the other methods using every dataset. It is interesting to note that due to the dense matrix form of the metric tensor and its inverse computational cost of RM-HMC on this example will not scale favourably and it can be seen it outperforms by the smallest margin on the German Credit dataset, which has largest number of regression coefficients ($b = 25$) and the largest number of data points ($N = 1000$). A further example based on a stochastic volatility model is now considered where the metric tensor and its inverse are sparse permitting scaling of RM-HMC to very high dimensions.

Table 4. Australian Credit Dataset, $D = 14$, $N = 690$, 15 regression coefficients - Comparison of sampling methods

Method	Time	ESS (Min, Med, Max)	s/Min ESS	Rel. Speed
Metropolis	16.5	(15, 199, 698)	1.10	×10.7
Aux. Var.	562.9	(48, 1087, 1457)	11.73	×1
MALA	No Convergence	(-, -, -)	-	-
HMC	No Convergence	(-, -, -)	-	-
RM-HMC	82.6	(4769, 5000, 5000)	0.0173	×678

Table 5. German Credit Dataset, $D = 24$, $N = 1000$, 25 regression coefficients - Comparison of sampling methods

Method	Time	ESS (Min, Med, Max)	s/Min ESS	Rel. Speed
Metropolis	44.9	(10, 81, 604)	4.49	×1
Aux. Var.	831.2	(1089, 2164, 2655)	0.76	×5.9
MALA	7.7	(3, 5, 175)	2.57	×1.7
HMC	3161.6	(2707, 4201, 5000)	1.17	×3.8
RM-HMC	892.3	(2264, 3084, 3717)	0.39	×11.5

Table 6. Pima Indian Dataset, $D = 7$, $N = 532$, 8 regression coefficients - Comparison of sampling methods

Method	Time	ESS (Min, Med, Max)	s/Min ESS	Rel. Speed
Metropolis	6.5	(14, 35, 181)	0.46	×21.7
Aux. Var.	468.6	(1138, 1957, 2397)	0.41	×24.3
MALA	29.9	(3, 10, 39)	9.97	×1
HMC	1499.1	(3149, 3657, 3941)	0.48	×20.8
RM-HMC	29.3	(4981, 5000, 5000)	0.006	×1662

Table 7. Heart Dataset, $D = 13$, $N = 270$, 14 regression coefficients - Comparison of sampling methods

Method	Time	ESS (Min, Med, Max)	s/Min ESS	Rel. Speed
Metropolis	10.4	(7, 63, 516)	1.49	×3.7
Aux. Var.	215.7	(722, 1275, 1719)	0.30	×18.3
MALA	No Convergence	(-, -, -)	-	-
HMC	2018	(368, 2740, 2938)	5.48	×1
RM-HMC	85.9	(3371, 4031, 4519)	0.025	×219

Table 8. Ripley Dataset, $D = 2$, $N = 250$, 7 regression coefficients - Comparison of sampling methods

Method	Time	ESS (Min, Med, Max)	s/Min ESS	Rel. Speed
Metropolis	4.1	(9, 18, 248)	0.46	×5.9
Aux. Var.	175.9	(68, 373, 2008)	2.59	×1
MALA	1.8	(4, 7, 27)	0.45	×5.7
HMC	52.8	(1365, 1596, 1754)	0.039	×66.4
RM-HMC	58.3	(3586, 4106, 4522)	0.016	×162

8. Conclusions and Discussion

In this paper a Riemannian Manifold Hamiltonian Monte Carlo sampler has been developed in an attempt to improve upon existing MCMC methodology when sampling from target densities that may be of high dimension and exhibit strong correlations. It is argued that the method is fully automated in terms of tuning the overall proposal mechanism to accommodate target densities which may exhibit strong correlations, widely varying scales in each dimension, and significant changes in the geometry of the manifold between the transitional and stationary phases of the Markov chain.

By exploiting the natural Riemannian structure of the parameter space of statistical models the proposed method can be seen to be a generalisation of both HMC and MALA methods and as such overcomes the oftentimes complex manual tuning required of both methods. In high dimensional problems such as inferring the 4096 dimensional latent Gaussian field MALA and HMC fail completely due to the high levels of spatial correlation in the latent field and can only proceed after a transformation is used to break those correlations. In contrast RM-HMC proceeds without the need for such a transformation or indeed phase specific tuning.

A novel semi-explicit symplectic integrator is developed for the non-seperable Hamiltonian that emerges due to the appearance of the metric tensor in the log joint-likelihood. Of course the Generalised Leapfrog method for non-separable Hamiltonians is available see e.g. Leimkuhler and Reich (2004) however the updates for parameters and auxiliary variables (momentum) are defined fully implicitly requiring further nonlinear solutions for the iterations. The Sundman transformation could be considered but it is unclear how this would be at all practical for the general methodology which was being developed. What is important is that the second order convergence of the Newton step has, empirically, been found to require a single step in all of the examples that have been considered in this paper when simulating paths across the manifold. The overall RM-HMC method employing this symplectic integrator has been shown to provide highly efficient convergence and exploration of the target density for the range of models considered.

Clearly there are two main overheads when employing RM-HMC, the development of the analytical expressions for the metric tensor and the associated derivatives as well as the $\mathcal{O}(N^3)$ scaling of solving the linear systems when updating the parameter vectors i.e. inverting the metric tensor. In all but the nonlinear differential equation example, exact analytical expressions for the Fisher Information could be obtained. It remains to be seen what other classes of statistical models may have this same issue, nevertheless even with this approximation RM-HMC remains superior to HMC and MALA in time normalised sampling efficiency in this example.

The issue of the $\mathcal{O}(N^3)$ scaling is something which deserves further consideration. In some statistical models there is a natural sparsity in the metric tensor, the SVM is a case in point where due to this structure RM-HMC was computationally more efficient than HMC. In other models this is not the case for example the logistic regression model and the Log-Gaussian Cox model. It should be noted that adaptive MCMC methods, see e.g. Andrieu and Thoms (2008), also incur the same level of cubic scaling. At the very high dimensional end of the scale a decorrelating transformation is required for MALA and HMC and this will also incur an $\mathcal{O}(N^3)$ scaling however further work to characterise the incurred computational costs at the intermediate dimensionality regime will be of value.

In summary the RM-HMC method provides a novel MCMC algorithm whose performance has been assessed on a diverse range of statistical models and in all cases has been shown to be superior to similar MCMC methods. We finally note, as has been highlighted previously, (Neal, 1993a, 1996; Liu, 2001), that RM-HMC can be embedded within a population MCMC procedure when the posterior has distinct separated modes, see e.g. Calderhead *et al.* (2009) .

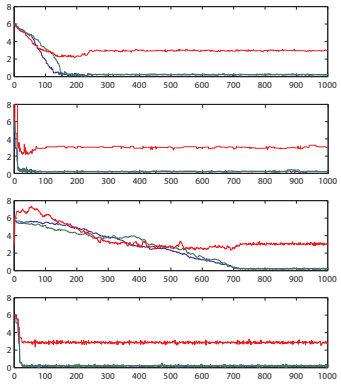

Fig. 8. Trace of the first 1000 model parameter samples for Fitzhugh Nagumo with Metropolis (top), HMC with unit masses (second top), HMC with tuned masses (second bottom) and RM-HMC (bottom)

Table 14. Fitzhugh Nagumo: Summary of results for 100 runs of the full sampling scheme with 2000 posterior samples

Sampling Method	Burn-in Time (s)	Posterior Time (s)	Mean ESS (a, b, c)	Total Time/ (Min mean ESS)	Relative Speed
Metropolis	201.8	165.7	397, 417, 311	1.181	$\times 1.04$
HMC	460	607	1399, 865, 1254	1.234	$\times 1$
RM-HMC	52.5	252.4	1978, 1931, 1824	0.317	$\times 3.89$

advantage of the RM-HMC scheme; it is self-tuning in that the metric tensor automatically adapts the mass matrix to the local topology of the parameter space, allowing it to take bigger steps through parameter space when required. This is often the case during the burn in period, particularly for this type of inference problem where, unlike in the logistic regression example, a nonlinear likelihood is induced by the nonlinearities of the differential equation model. The results of our simulations are shown in Table (14). Although Metropolis is quickest at drawing 2000 posterior samples, its mean ESS is around five times smaller than that of RM-HMC and, importantly, takes much longer than RM-HMC to converge to the target distribution. As a result, Metropolis requires around 3 times longer than RM-HMC to produce an effectively independent sample. The HMC scheme fares the worst of the three methods. As we have discussed, this is due to the rather sensitive effect that tuning the masses has on the ESS. In addition, since the masses are fixed throughout the simulation, much smaller integration stepsizes are required, and consequently a larger number of integration steps. This HMC approach then becomes computationally very intensive for this problem and is seen to be less efficient than Metropolis as a direct result. The fast convergence of RM-HMC combined with its high ESS scores results in this approach being the most efficient of the three methods, even normalising against the computational effort required.

Table 9. Summary of standard deviations of the marginal posterior distributions for each dataset

Dataset	Smallest Marg. S.D.	Largest Marg. S.D.	Ratio
Pima Indian	0.0043	0.9646	225
Australian Credit	0.00017	1.0667	6404
German Credit	0.0038	1.1492	303
Heart	0.004	2.9221	739
Ripley	1.2575	7.556	6

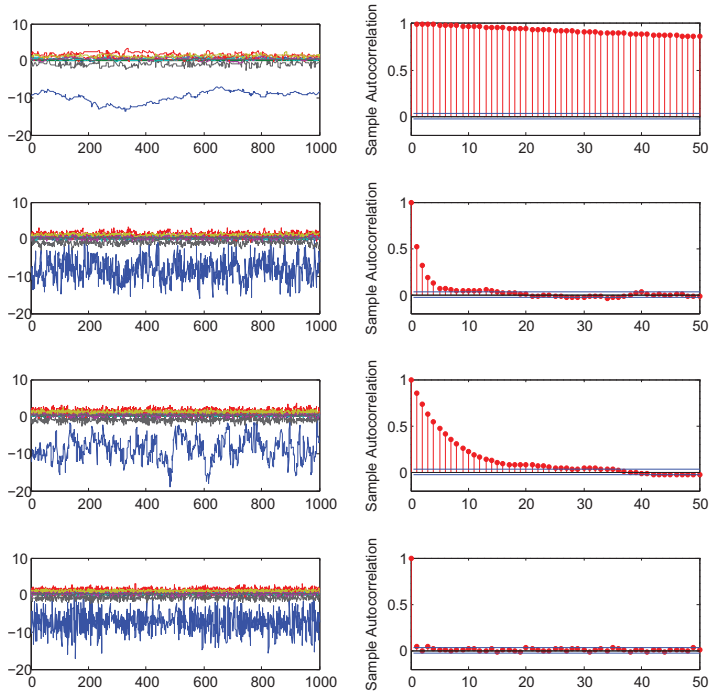

Fig. 3. Trace plots for 1000 posterior samples with the Heart dataset using Metropolis (top), auxiliary variable sampler (second top), standard HMC (second bottom), and RM-HMC (bottom). Autocorrelation plots are also shown for one of its parameters, which may be seen in the trace plots to have a mean of around -7.

5. RM-HMC for a Stochastic Volatility Model

A stochastic volatility model (SVM) studied in Liu (2001) is defined with the latent volatilities taking the form of an AR(1) process such that $y_t = \epsilon_t \beta \exp(x_t/2)$ with $x_{t+1} = \phi x_t + \eta_{t+1}$ where $\epsilon_t \sim \mathcal{N}(0,1)$, $\eta_t \sim \mathcal{N}(0,\sigma^2)$ and $x_1 \sim \mathcal{N}(0, \sigma^2/(1-\phi^2))$ having joint likelihood

$$p(\mathbf{y}, \mathbf{x}, \beta, \phi, \sigma) = \prod_{t=1}^{T} p(y_t|x_t,\beta)p(x_1) \prod_{t=2}^{T} p(x_t|x_{t-1},\phi,\sigma)\pi(\beta)\pi(\phi)\pi(\sigma). \quad (16)$$

We may conveniently split up the sampling procedure into two steps, which as we shall see allows the implementation of RM-HMC in a computationally efficient manner. Firstly we may simulate ϕ, σ, β from $p(\beta, \phi, \sigma|\mathbf{y}, \mathbf{x})$, where the priors, as in Liu (2001), are chosen to be $p(\beta^2) \propto \beta^{-2}$, $\sigma^2 \sim$ Inv-$\chi^2(10, 0.05)$ and $(\phi - 1)/2 \sim$ Beta$(20, 1.5)$. Secondly we may sample the latent volatilities by simulating from the conditional $p(\mathbf{x}|\mathbf{y}, \beta, \phi, \sigma)$. We shall consider the use of RM-HMC, HMC and MALA for the purpose of sampling both the parameters and latent volatilities.

5.1. RM-HMC for SVM Parameters

We introduce a transformation of the parameters to ensure that they are suitably bounded, $\sigma = \exp(\gamma)$ and $\phi = \tanh(\alpha)$, and we obtain expressions for the transformed priors and log-joint likelihood accordingly. We now require the partial derivatives of the log-joint likelihood with respect to the parameters, as well expressions for the metric tensor and its partial derivatives, in order to implement the RM-HMC, HMC and MALA methods. All of these quantities may be obtain straightforwardly (see Appendix B for details). In particular, the Fisher Information is given by

$$\begin{bmatrix} \frac{2T}{\beta^2} & 0 & 0 \\ 0 & T+1 & 2\phi \\ 0 & 2\phi & \phi^2(3-T)+(T-1) \end{bmatrix}$$

where T is the number of observations. The metric tensor follows by adding this Fisher Information to the prior precision. Having transformed the priors on β, σ and ϕ into valid priors on β, γ and α, we may now use any of these methods to draw samples from the conditional posterior $p(\beta, \gamma, \alpha|\mathbf{y}, \mathbf{x},)$, and transform the posterior samples to obtain $\beta, \sigma = \exp(\gamma)$ and $\phi = \tanh(\alpha)$.

5.2. RM-HMC for SVM Latent Volatilities

For all three gradient based methods, we require the gradient of the joint-log likelihood with respect to each of the latent volatilities. Defining the vectors $\mathbf{u} = (x_3, \cdots, x_T)^\mathsf{T}$, $\mathbf{v} = (x_2, \cdots, x_{T-1})^\mathsf{T}$, $\mathbf{w} = \frac{\phi}{\sigma^2}(\mathbf{u} - \phi\mathbf{v})$, $\mathbf{s} = (s_1, \cdots, s_T)^\mathsf{T}$ such that $s_i = 0.5(1 - y_i^2\beta^{-2}\exp(-x_i))$, $\delta_1 = -\sigma^{-2}(x_1 - \phi x_2)$, and $\delta_T = -\sigma^{-2}(x_T - \phi x_{T-1})$, we define the vector $\mathbf{r} = (\delta_1, \mathbf{w}^\mathsf{T}, \delta_2)^\mathsf{T}$ and the required gradient is $\nabla_{\mathbf{x}} \log p(\mathbf{y}, \mathbf{x}|\beta, \phi, \sigma) \equiv \nabla_{\mathbf{x}}\mathcal{L} = \mathbf{s} - \mathbf{r}$.

To devise an RM-HMC sampler for the latent volatilities, $\mathbf{x}$, we also require an expression for the metric tensor and its partial derivatives with respect to the latent volatilities. For the data likelihood of the model, $p(\mathbf{y}|\mathbf{x}, \beta)$, the Fisher Information is a diagonal matrix with 0.5 for each element denoted as $\mathbf{I}_{0.5}$. The latent volatility is an AR(1) process having covariance matrix $\mathbf{C}$ with elements $E\{x_{t+n}x_t\} = \phi^{|n|}\sigma^2/(1-\phi^2)$ and as in the previous examples the metric tensor is defined as the sum of the Fisher Information and prior precision, $\mathbf{G} = \mathbf{I}_{0.5} + \mathbf{C}^{-1}$, conditional on current values of σ, ϕ, β. Now the expression for the covariance matrix is completely dense and is therefore computationally expensive to manipulate. Fortunately, this AR(1) process admits a simple analytic expression for the precision matrix in the form of a sparse tridiagonal matrix, such that the diagonal

Table 13. Fitzhugh Nagumo Species R: Summary of results for 100 runs of GP regression for 2000 posterior samples

Sampling Method	Time (s)	Mean ESS ($\varphi_1^R, \varphi_2^R, \sigma^R$)	Time/(Min mean ESS)	Rel. Speed
Metropolis	20.7	128, 146, 251	0.164	$\times 18.2$
MALA	14.6	4.9, 5.2, 79.2	2.980	$\times 1$
HMC	77.5 + 20.7	343, 559, 942	0.286	$\times 10.4$
RM-HMC	126.5	1767, 1692, 1941	0.075	$\times 39.7$

distribution given in equation (21),

$$\frac{\partial \dot{V}}{\partial a} = \frac{\partial \dot{V}}{\partial b} = 0, \frac{\partial \dot{V}}{\partial c} = \left(V - \frac{V^3}{3} + R\right), \frac{\partial \dot{R}}{\partial a} = \frac{1}{c}, \frac{\partial \dot{R}}{\partial b} = \frac{-R}{c}, \frac{\partial \dot{R}}{\partial c} = \left(\frac{V - a + bR}{c^2}\right)$$

All of the second derivatives of $\dot{V}$ with respect to the model parameters are equal to zero, and the five non-zero second partial derivatives of $\dot{R}$ are as follows,

$$\frac{\partial^2 \dot{R}}{\partial a \partial c} = -\frac{1}{c^2}, \frac{\partial^2 \dot{R}}{\partial b \partial c} = \frac{R}{c^2}, \frac{\partial^2 \dot{R}}{\partial c \partial a} = -\frac{1}{c^2}, \frac{\partial^2 \dot{R}}{\partial c \partial b} = \frac{R}{c^2}, \frac{\partial^2 \dot{R}}{\partial c^2} = 2\left(\frac{-V + a - bR}{c^3}\right)$$

We now compare the performance of M-H, HMC and RM-HMC. We note that the performance of the HMC scheme was very sensitive to the tuning of its mass parameters, and the number and size of the integration steps required for a reasonable acceptance rate were considerably higher and lower respectively than those needed when using RM-HMC. We see that the extra computational expense incurred in computing the metric tensor and its derivatives is offset by the fact that fewer integration steps are needed for each new parameter proposal since larger stepsizes may be employed.

Additionally, the tuned HMC scheme sometimes took much longer to converge than RM-HMC, particularly if the initial model parameter values were far from the true values, since the integration parameters were tuned to optimise sampling from the true posterior and different integration parameters are often required to sample efficiently in different regions of parameter space. Figure (8) shows the trace of a Markov chain generated by each of the methods with the initial parameter values $a = 6$, $b = 6$, $c = 6$. The difference in the rate of convergence is striking, with RM-HMC converging the quickest. While the M-H scheme converges slower than RM-HMC, it converges much quicker than HMC with tuned masses, although it does then require another few hundred iterations to adapt to the posterior mode until reasonable acceptance rates are achieved. Once the tuned HMC scheme does finally reach the posterior mode it visibly samples much more effectively than Metropolis.

On the other hand, if we consider an HMC scheme with the same size and number of integration steps but with unit masses, instead of carefully tuned masses, we see the Markov chain traverse the parameter space much more quickly but then sample very poorly from the posterior mode, since the stepsize is too large relative to the width of the mode. The transient and stationary phases of the HMC algorithm clearly require different level of scaling of the masses, see Christensen *et al.* (2005) for a theoretical analysis related to MALA, whereas RM-HMC overcomes this issue automatically.

In summary, the masses in HMC may be set approximately equal to the marginal variance of each parameter to ensure efficient sampling from the posterior. However, if these masses are very small relative to the size of space being explored, then the chain cannot travel far in one iteration resulting in slow convergence to the posterior mode. In this example we again see the clear

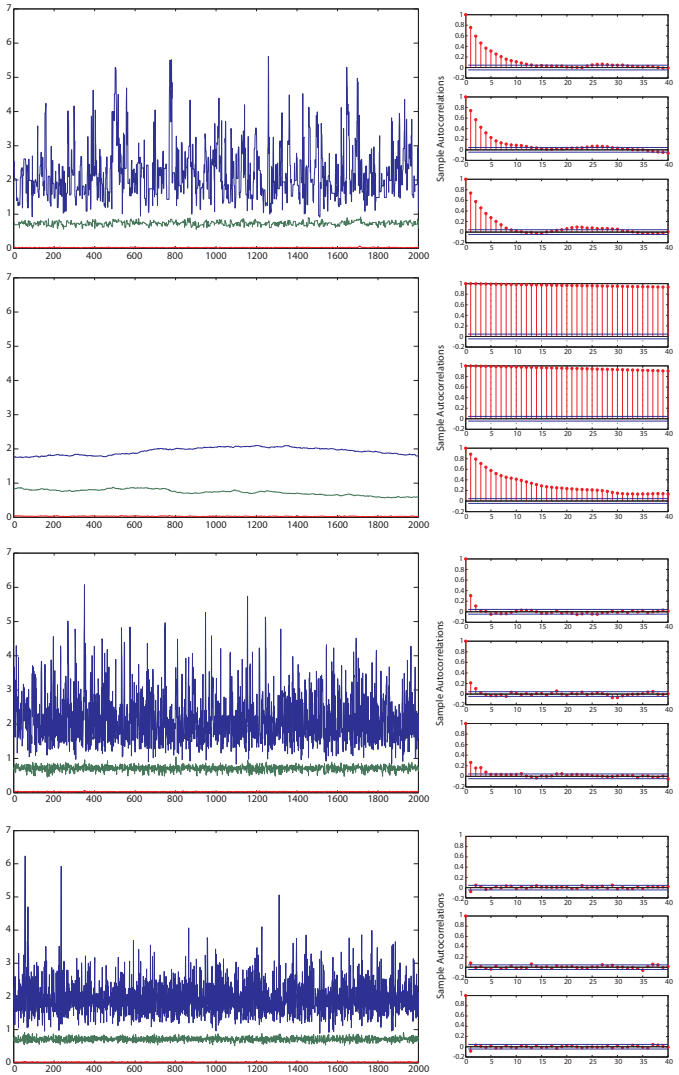

Fig. 7. Trace and corresponding autocorrelation plots for 2000 posterior samples of the 3 RBF hyperparameters (φ_1^V, φ_2^V and σ^V, top to bottom respectively) for species V of the FHN model using Metropolis (top), MALA (second top), standard HMC (second bottom), and RM-HMC (bottom)

Table 10. 2000 simulated observations with $\beta = 0.65$, $\sigma = 0.15$ and $\phi = 0.98$ - Comparison of sampling the parameters β, σ and ϕ after 20,000 posterior samples averaged over 10 runs

Method	Mean Time	ESS (β,σ,ϕ)	S.E. (β,σ,ϕ)	s/(Min ESS)	Rel. Speed
MALA	58.2	(12.7, 10.0, 33.2)	(3.2,1.3,5.2)	5.8	$\times$1.7
HMC	996	(101,104,212)	(11.7,8.1,20.4)	9.9	$\times$1
RM-HMC	368	(224, 127, 300)	(19,8.8,31)	2.9	$\times$3.4

Table 11. 2000 simulated observations with $\beta = 0.65$, $\sigma = 0.15$ and $\phi = 0.98$ - Comparison of sampling the latent volatilities after 20,000 posterior samples averaged over 10 runs

Method	Mean Time	ESS (min, median, max)	s/(Min ESS)	Rel. Speed
MALA	58.2	(10.3,15.7,26.1)	5.7	$\times$1
HMC	996	(278,400,904)	3.6	$\times$1.6
RM-HMC	368	(558,877,1698)	0.66	$\times$8.6

elements are equal to $(1+\phi^2)/\sigma^2$, with the exception of the first and last diagonal elements which are equal to $1/\sigma^2$, and the super and sub diagonal elements are equal to $-\phi/\sigma^2$. Thus the metric tensor also has a tridiagonal form. For large numbers of observations this sparse structure allows great gains in computational efficiency, since the inverse of this tridiagonal metric tensor may be computed in $\mathcal{O}(n)$ as opposed to the usual $\mathcal{O}(n^3)$. We note that computationally efficient methods for manipulating tridiagonal matrices are automatically implemented by the standard routines in Matlab.

We notice that the metric tensor in this case is not a function of $\mathbf{x}$ and so the associated partial derivatives with respect to the latent volatilities are zero. In this case a one step RM-HMC integration scheme collapses to

$$\mathbf{x} = \mathbf{x}_0 + \frac{\epsilon^2}{2}\mathbf{G}^{-1}\nabla_{\mathbf{x}}\mathcal{L} + \epsilon\sqrt{\mathbf{G}^{-1}}\mathbf{p} \tag{17}$$

where $\mathbf{p} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ which is a discrete Langevin iteration that is preconditioned by the constant matrix $\mathbf{G}^{-1}$. It is clear that this preconditioning will improve both the mixing and overall ESS, see (Lambert and Eilers, 2009) for a recent application of this type of preconditioning in MALA. We point out that in the case of RM-HMC the preconditioning matrix emerges naturally from the underlying geometric principles of RM-HMC.

5.3. *Experimental Results for Stochastic Volatility Model*

We now compare the computational efficiency of RM-HMC, HMC and MALA for sampling both the parameters and the latent variables of the stochastic volatility model as previously defined. 2000 observations were simulated from the model with the parameter values $\beta = 0.65, \sigma = 0.15$ and $\phi = 0.98$ as given in Liu (2001). Using this data, 20000 posterior samples were collected after a burn-in period. This sampling procedure was repeated 10 times. The efficiency was compared in terms of time normalised ESS, as in the previous section, for the parameters and the latent volatilities. MALA was tuned such that the acceptance ratio was between 40% and 60%, and it was necessary to use a different tuning for the transient phase than for the stationary phase. HMC was implemented using a step size of 0.015 and 200 integration steps per proposal. RM-HMC was implemented using a stepsize of 0.5 and 10 integration steps per parameter proposal, and a stepsize of 0.1 and 50 integration steps per volatility proposal.

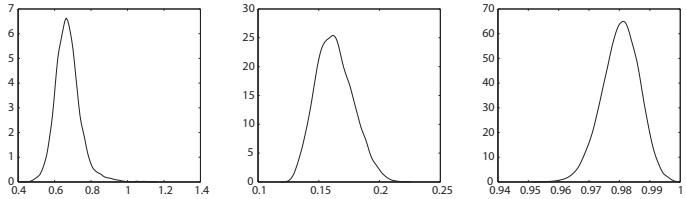

Fig. 4. Posterior histograms for β, σ and ϕ respectively, employing RM-HMC to draw 20,000 samples of the parameters and latent volatilities using a simulated dataset consisting of 2000 observations. The true values are $\beta = 0.65$, $\sigma = 0.15$ and $\phi = 0.98$.

RM-HMC gives the best performance both in terms of sampling the parameters and also the latent volatilities. In particular it runs faster than HMC, partly because of the computationally efficient tridiagonal structure of the metric tensor and partly because RM-HMC follows the contravariant tensor gradient through the parameter space and explores regions of the target density more quickly than HMC, refer to Figure 1 and 2 for an illustration of the contrast between HMC and RM-HMC sampling of the parameters of this model. MALA exhibits a very poor ESS, however the computation time is also extremely small compared to the other two methods and so, based on the normalised ESS, MALA does not perform quite as badly as the unnormalised ESS values alone might suggest. It should again be noted that in addition to RM-HMC outperforming HMC and MALA, RM-HMC requires very little tuning compared to the other methods; unlike MALA it does not require different tuning in different parts of the parameter space, and unlike HMC it requires no manual setting of a mass matrix.

We now consider an example where the target density is extremely high dimensional, which is encountered when performing inference using spatial data modeled by a log-Gaussian Cox process.

6. RM-HMC for Log-Gaussian Cox Point Processes

RM-HMC is further studied using the example of inference in a log-Gaussian Cox point process as detailed in (Christensen *et al.*, 2005). This is a particularly useful example in that the target density is of high dimension with strong correlations and provides a severe test of MCMC capability. The data, model and experimental protocol as described in (Christensen *et al.*, 2005) is adopted here. A 64×64 grid is overlayed on the area $[0, 1]^2$ with the number of points in each grid cell denoted by the random variables $\mathbf{Y} = \{Y_{i,j}\}$ which are assumed conditionally independent, given a latent intensity process $\Lambda(\cdot) = \{\Lambda(i, j)\}$, and are Poisson distributed with means $m\Lambda(i, j) = m \exp(X_{i,j})$, where m = 1/4096. The random variable $\mathbf{X} = \{X_{i,j}\}$ is a Gaussian process with mean $E\{\mathbf{x}\} = \mu\mathbf{1}$ and covariance function $\mathbf{\Sigma}_{(i,j),(i',j')} = \sigma^2 \exp(-\delta(i, i', j, j')/64\beta)$, where $\delta(i, i', j, j') = \sqrt{(i - i')^2 + (j - j')^2}$. The joint density is

$$p(\mathbf{y}, \mathbf{x}|\mu, \sigma, \beta) \propto \prod_{i,j}^{64} \exp\{y_{i,j}x_{i,j} - m \exp(x_{i,j})\} \exp(-(\mathbf{x} - \mu\mathbf{1})^{\mathsf{T}}\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu\mathbf{1})/2) \quad (18)$$

Denoting $\mathcal{L} \equiv \log p(\mathbf{y}|\mathbf{x}, \mu, \sigma, \beta)$, $\mathbf{e} = \{m \exp(x_{i,j})\}$ then $\nabla_{\mathbf{x}}\mathcal{L} = \mathbf{y} - \mathbf{e}$. The Fisher Information matrix is $\mathbf{E} = \mathsf{diag}(\mathbf{e})$, with the addition of the prior precision matrix the metric tensor and its

Table 12. Fitzhugh Nagumo Species V: Summary of results for 100 runs of GP regression for 2000 posterior samples

Sampling Method	Time (s)	Mean ESS (φ_1^V, φ_2^V, σ^V)	Time/(Min mean ESS)	Rel. Speed
Metropolis	17.8	214, 194, 282	0.091	$\times 32.5$
MALA	16.1	5.4, 5.9, 50.3	2.960	$\times 1$
HMC	56.2 + 17.8	797, 1311, 1425	0.093	$\times 31.8$
RM-HMC	101.8	1987, 1581, 1821	0.064	$\times 46.3$

for comparison, calculating the time per effectively independent sample. 100 simulations were run for each method, and for each simulation new experimental data was generated, thus the marginal distributions were slightly different in each run. All methods were implemented in the interpreted language Matlab for consistency of comparison.

In order to implement our RM-HMC scheme for sampling the GP hyperparameters, we must compute the metric tensor, given by the Fisher Information matrix, as well as its partial derivatives with respect to each of the hyperparameters of the GP. These can be derived straightforwardly as a function of the GP covariance function and its first and second partial derivatives (see Appendix E for full details). In this example due to the smoothness of the dynamics induced by the system of equations we employ a stationary radial basis covariance function, although other covariance functions may also be employed to better capture the characteristics of the specific data being modeled.

For setting the masses in the HMC scheme, we employed estimates using the average marginal variances obtained from running the Metropolis-Hastings scheme and chose the stepsize and number of integration steps such that the acceptance rate was greater than 70%. The setting of the masses using such exploratory Metropolis runs was necessary to achieve a reasonable acceptance rate, since the marginal distributions of the hyperparameters were of different orders of magnitude. In our results we therefore add the average time taken for a Metropolis run to the average time taken for an HMC run, although in practice extra time is required to implement this necessary tuning.

The results of our simulations are given in Tables 12 and 13, and the posterior samples from a typical run for each method are given in Figure 7. RM-HMC samples about twice as effectively as the M-H scheme in terms of normalised ESS, and the difference in the correlation between consecutive samples is clearly visible in the trace plots. We see that MALA fares very badly when sampling the hyperparameters, and this is due to the different orders of magnitude of the marginal distribution in each dimension. The standard implementation (Roberts and Rosenthal, 1998; Roberts and Stramer, 2003) employs a symmetric proposal distribution, thus when the algorithm adapts its stepsize (according to the acceptance rate) it is limited by the dimension with the smallest marginal variance. It therefore generally samples one parameter much more efficiently than the others, as may be seen from the ESS values reported in Tables 12 and 13. We could of course tune MALA by premultiplying by a mass matrix, and so the approach becomes equivalent to an HMC scheme with just one integration step when proposing new hyperparameters. The difference between the two methods then is that HMC suppresses the random walk aspect of MALA, which is generally desirable for improving the speed of exploration of an unknown probability distribution. Of these two methods, we therefore consider only HMC in the following sections, given that it is a generalisation of the Langevin approach.

7.3.2. Full RM-HMC Sampling Scheme

We now also require the first and second partial derivatives of the Fitzhugh Nagumo equations in order to calculate the metric tensor (see Appendix D) for employing RM-HMC to sample from the

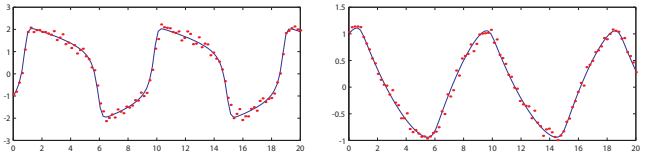

Fig. 6. Output for species V (left) and species R (right) of the Fitzhugh Nagumo model with parameters $a = 0.2$, $b = 0.2$, $c = 3$. An example noisy dataset is shown by the red points.

where $\kappa_n^{s,s'}$ denotes the s, s' element of $(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}$, and all the first and second order partial derivatives of the vector field can be obtained analytically from the system of differential equations.

From details given in the appendix, the metric for each of the error variance terms $\sqrt{\gamma_n}$ is the Fisher Information $g(\sqrt{\gamma_n}) = \mathcal{I}_{\sqrt{\gamma_n}} \approx 2\gamma_n \mathsf{trace}\left(\mathbf{H}_n^{-1}\mathbf{H}_n^{-1}\right)$ where $\mathbf{H}_n = (\mathbf{K}_n + \mathbf{I}\gamma_n)$, and the corresponding derivative $\partial g(\sqrt{\gamma_n})/\partial\sqrt{\gamma_n} = 2\mathsf{trace}\left(2\sqrt{\gamma_n}\mathbf{H}_n^{-1}\mathbf{H}_n^{-1}(\mathbf{I} - \gamma_n\mathbf{H}_n^{-1})\right)$.

Finally, Appendix E provides the details for the RM-HMC procedure for (19). We now have everything required to implement an RM-HMC sampling scheme for dynamical system models defined by systems of nonlinear differential equations.

7.3. *Experimental Results for Nonlinear Differential Equations*

We now present results comparing the sampling efficiency for the parameters of the Fitzhugh Nagumo differential equations (Ramsay *et al.*, 2007),

$$\dot{V} = c\left(V - \frac{V^3}{3} + R\right), \quad \dot{R} = -\left(\frac{V - a + bR}{c}\right) \tag{23}$$

We may follow the sampling scheme given by equations (19) to (21) to obtain samples from the joint posterior $p(\boldsymbol{\theta}, \mathbf{X}, \boldsymbol{\varphi}, \boldsymbol{\sigma}, \boldsymbol{\gamma}|\mathbf{Y})$, and so in this example $\mathbf{X}_{1,\cdot} = \mathbf{V}$ and $\mathbf{X}_{2,\cdot} = \mathbf{R}$. We note that we can sample exactly from equation (20) and so we investigate the efficiency of sampling methods for the GP hyperparameters, the model mismatch parameters and the structural model parameters. Indeed, we split the sampling step given by equation (21) into two steps for ease of computation, firstly sampling $p(\boldsymbol{\gamma}|\boldsymbol{\theta}, \mathbf{X}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$, then sampling $p(\boldsymbol{\theta}|\boldsymbol{\gamma}, \mathbf{X}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$. We employ a Metropolis-Hastings scheme, MALA, HMC and RM-HMC, using Scheme 1, as described for the logistic regression simulations. We again compare the simulations by calculating the effective sample size (ESS) normalised by the computational time required to produce the samples.

7.3.1. *Sampling GP Hyperparameters*

We first examine the sampling of the hyperparameters for the GP (equation (19)). We investigate the performance of each method with data generated from the Fitzhugh Nagumo ODE model. We used 100 data points generated between $t = 0$ and $t = 20$ with the model parameters $a = 0.2$, $b = 0.2$, $c = 3$. Gaussian distributed noise was then added to the data, with standard deviation equal to 10% of the standard deviation of each species respectively, see Figure 6.

All of the methods converged to the three dimensional target distribution within 2000 burn in iterations (according to the acceptance ratios and Gelman's $\hat{R}$ statistic), after which 2000 posterior samples were collected. We calculated the ESS for each parameter and used the minimum value

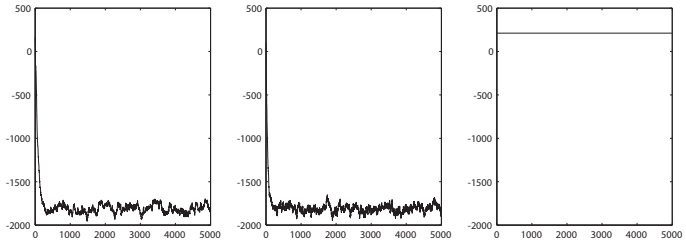

Fig. 5. Trace plots of the log joint-likelihood for the first 5000 samples of the latent variables of a log-Gaussian Cox process. The left hand plot shows the convergence of the RM-HMC scheme which is able to directly sample the latent variables **x** without the need for ad-hoc reparameterisations and pilot runs for fine-tuning. The middle plot shows the log joint-likelihood for samples drawn by MALA using a reparameterisation of the latent variables. The scaling was carefully tuned to allow traversal of the parameter space to the posterior mode. The right hand plot shows the trace of the MALA sampler tuned for optimally sampling from the posterior mode. We note that the algorithm is now unable to traverse the parameter space when initialised away from this mode. Such fine-tuning and reparameterisation is frequently necessary when employing MALA and the HMC sampler.

partial derivatives follow as $\mathbf{G}(\mathbf{x}) = \mathbf{E} + \boldsymbol{\Sigma}^{-1}$ and $\partial\mathbf{G}(\mathbf{x})/\partial x_{i,j} = E'$ where E' is a zero matrix with $m\exp(x_{i,j})$ as the $(64 \times (i-1) + j)^{th}$ element on the diagonal.

Noting that the metric tensor has dimension 4096×4096 the $\mathcal{O}(N^3)$ operations required in the RM-HMC scheme are clearly going to be computationally costly. However, it should be noted that in previous studies of this Log-Gaussian Cox process, (Christensen *et al.*, 2005), a transformation of the latent Gaussian field is necessary based on the Cholesky decomposition of $\boldsymbol{\Sigma}^{-1} + \mathsf{diag}(\mathbf{x})$, which will therefore also scale as $\mathcal{O}(N^3)$.

6.1. *Experimental Results for Log-Gaussian Cox Processes*

Following the example given by Christensen *et al.* (2005), we fix the parameters $\beta = 1/33$, $\sigma^2 = 1.91$ and $\mu = \log(126) - \sigma^2/2$. We generate a latent Gaussian field, **x**, from the Gaussian process and use these values to generate count data **y** from the latent intensity process $\boldsymbol{\Lambda}$. Given the generated data and the fixed parameters, we infer **x** using RM-HMC and the Langevin method as in Christensen *et al.* (2005).

The Langevin based method (MALA) is particularly sensitive to the choice of scaling, as is HMC, and very often a reparameterisation of the target density is required for these methods to be effective. Indeed this is seen to be the case with this particular example, where MALA is unable to sample **x** directly. We therefore follow Christensen *et al.* (2005) and employ the transformation $\mathbf{X} = \boldsymbol{\mu}\mathbf{1} + \mathbf{L}\boldsymbol{\Gamma}$, where $\mathbf{L}$ is obtained by Cholesky factorisation such that $\{\boldsymbol{\Sigma} - \mathrm{diag}(\mathbf{x})\}^{-1} = \mathbf{L}\mathbf{L}^{\mathsf{T}}$. Even after this re-parameterisation, it is still necessary to carefully tune the scaling factor for this method to work at all. This challenging aspect of employing MALA has been investigated in detail by Christensen *et al.* (2005) who characterise the problem very well, advise great care in its implementation, but ultimately are unable to offer any panacea. In contrast to the necessary transformation and fine-tuning required by MALA, RM-HMC allows us to directly sample the latent variables **x** *without* reparameterising the target density. Additionally, no manual tuning via pilot runs

is required since the stepsize of the symplectic integrator may be adjusted automatically based on the acceptance rate.

Figure 5 shows the traces of the log joint-likelihood for both methods using the starting position $x_{i,j} = \mu$ for $i, j = 1, \ldots, 64$. Note that for MALA these starting positions must be transformed into corresponding values for $\mathbf{\Gamma}$. The RM-HMC sampler quickly converges to the true mode after very minimal automatic tuning of the integration stepsize based on the acceptance rate. MALA converges in a similar number of iterations, but only for a suitable choice of scaling factor. The middle plot in Figure 5 shows convergence when the scaling factor is carefully tuned for the transient phase of the Markov chain, however the right hand plot demonstrates how it fails to converge at all given a scaling factor which is tuned for stationarity. In this example the RM-HMC method required 10 seconds per sample compared to the 6 seconds needed by MALA, however this does not take into account the often considerable time and effort required to tune MALA. The algorithms were run on a single AMD Opteron processor with 8GB of memory and were coded in Matlab.

Inferring the latent field of a log-Gaussian Cox process with a finely grained discretisation is clearly a very challenging problem due to the high dimensionality and strong spatial correlations present between the latent variables. The major challenges associated with employing MALA are firstly finding a suitable reparameterisation of the target density, and secondly making a suitable choice for the scaling factor according to whether the Markov chain is in a transient or stationary regime. In contrast, RM-HMC does not exhibit such extreme technical difficulties. We have demonstrated that RM-HMC is able to sample directly from the original target distribution with minimal automatic tuning and effort, albeit with a slightly increased computational cost. We will now turn our attention to the very topical application of statistical inference to nonlinear differential equations.

7. RM-HMC for Nonlinear Differential Equation Models

An important class of problems recently gaining attention is the statistical analysis of uncertainty in dynamical systems defined by a system of nonlinear differential equations (Ramsay *et al.*, 2007; Calderhead *et al.*, 2009; Vyshemirsky and Girolami, 2008). A dynamical system may be described by a collection of N nonlinear ordinary differential equations and model parameters $\boldsymbol{\theta}$ which define a functional relationship between the process state, $\mathbf{x}(\mathbf{t})$, and its time derivative such that $\dot{\mathbf{x}}(\mathbf{t}) = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{t})$. A sequence of process observations, $\mathbf{y}(\mathbf{t})$, are usually contaminated with some measurement error, which is modeled as $\mathbf{y}(\mathbf{t}) = \mathbf{x}(\mathbf{t}) + \boldsymbol{\epsilon}(\mathbf{t})$, where $\boldsymbol{\epsilon}(\mathbf{t})$ defines an appropriate multivariate noise process, e.g. a zero-mean Gaussian with variance σ_n^2 for each of the N states. If observations are made at T distinct time points, the $N \times T$ matrices summarise the overall observed system as $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. In order to obtain values for $\mathbf{X}$, the system of ODEs must be solved, so that in the case of an initial value problem $\mathbf{X}(\boldsymbol{\theta}, \mathbf{x}_0)$ denotes the solution of the system of equations at the specified time points for the parameters $\boldsymbol{\theta}$ and initial conditions $\mathbf{x}_0$. The posterior density follows by employing appropriate priors such that $p(\boldsymbol{\theta}, \mathbf{x}_0, \boldsymbol{\sigma}|\mathbf{Y}) \propto \pi(\boldsymbol{\theta})\pi(\mathbf{x}_0)\pi(\boldsymbol{\sigma}) \prod_n \mathcal{N}(\mathbf{Y}_{n,\cdot}|\mathbf{X}(\boldsymbol{\theta}, \mathbf{x}_0)_{n,\cdot}, \mathbf{I}\sigma_n^2)$. The desired marginal $p(\boldsymbol{\theta}|\mathbf{Y})$ can be obtained from this joint posterior.

Various sampling schemes can be devised to sample from the joint posterior. However, regardless of the sampling method, each proposal requires the specific solution of the system of differential equations. This is the main computational bottleneck in running an MCMC scheme for models based on differential equations. The computational complexity of numerically solving such a system cannot be easily quantified since it depends on many factors such as the type of model and its stiffness, which in turn depends on the specific parameter values used. In Calderhead *et al.* (2009) an MCMC methodology was proposed, similar in spirit to the work of Ramsay *et al.* (2007), which sidesteps the issue of solving the system of equations within an MCMC routine by introducing auxiliary Gaussian Process (GP) functions (Williams and Rasmussen, 2006) to define distributions over the evolution of each state and their associated time derivatives.

7.1. Gaussian Process Regression on State Variables

Let us first introduce a statistical model based on GP regression to describe the time evolution of the observed dynamical system. For notational convenience assume independent GP priors on the state variables such that $p(\mathbf{X}_{n,\cdot}|\boldsymbol{\varphi}_n) = \mathcal{N}(\mathbf{X}_{n,\cdot}|\mathbf{0}, \mathbf{C}_{\boldsymbol{\varphi}_n})$, where $\mathbf{C}_{\boldsymbol{\varphi}_n}$ denotes the matrix of covariance function values with hyperparameters $\boldsymbol{\varphi}_n$. With noise $\boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I}_T)$, the state posterior, $p(\mathbf{X}_{n,\cdot}|\mathbf{Y}_{n,\cdot}, \sigma_n, \boldsymbol{\varphi}_n)$ follows as $\mathcal{N}(\mathbf{X}_{n,\cdot}|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, where $\boldsymbol{\mu}_n = \mathbf{C}_{\boldsymbol{\varphi}_n}(\mathbf{C}_{\boldsymbol{\varphi}_n} + \sigma_n^2\mathbf{I})^{-1}\mathbf{Y}_{n,\cdot}$ and $\boldsymbol{\Sigma}_n = \sigma_n^2\mathbf{C}_{\boldsymbol{\varphi}_n}(\mathbf{C}_{\boldsymbol{\varphi}_n} + \sigma_n^2\mathbf{I})^{-1}$. Given priors $\pi(\sigma_n)$ and $\pi(\boldsymbol{\varphi}_n)$ over the hyper-parameters their posterior is $p(\boldsymbol{\varphi}_n, \sigma_n|\mathbf{Y}_{n,\cdot}) \propto \pi(\sigma_n)\pi(\boldsymbol{\varphi}_n)\mathcal{N}(\mathbf{Y}_{n,\cdot}|\mathbf{0}, \sigma_n^2\mathbf{I} + \mathbf{C}_{\boldsymbol{\varphi}_n})$. The conditional prior for the state-derivatives follows as $p(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{X}_{n,\cdot}, \boldsymbol{\varphi}_n) = \mathcal{N}(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{m}_n, \mathbf{K}_n)$, where the mean and covariance are given by $\mathbf{m}_n = {}'\mathbf{C}_{\boldsymbol{\varphi}_n}(\mathbf{C}_{\boldsymbol{\varphi}_n})^{-1}\mathbf{X}_{n,\cdot}$ and $\mathbf{K}_n = \mathbf{C}''_{\boldsymbol{\varphi}_n} - {}'\mathbf{C}_{\boldsymbol{\varphi}_n}(\mathbf{C}_{\boldsymbol{\varphi}_n})^{-1}\mathbf{C}'_{\boldsymbol{\varphi}_n}$ with $\mathbf{C}''_{\boldsymbol{\varphi}_n}$ denoting the auto-covariance for each state-derivative, and $\mathbf{C}'_{\boldsymbol{\varphi}_n}$ and ${}'\mathbf{C}_{\boldsymbol{\varphi}_n}$ denoting the cross-covariances between the state and its derivative (Williams and Rasmussen, 2006). The GP specifies a jointly Gaussian distribution over the regression function modeling the system states and their time derivatives.

In Calderhead *et al.* (2009) a second statistical model of the state derivatives is obtained by assuming normal errors with variance γ_n between the state derivatives and the value of the vector field obtained when plugging in the GP posterior samples of the state values, $p(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) = \mathcal{N}(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{f}_n(\mathbf{X}, \boldsymbol{\theta}, \mathbf{t}), \gamma_n\mathbf{I})$. Both the GP-based regression model, $p(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{X}_{n,\cdot}, \boldsymbol{\varphi}_n)$, and the model representing the state derivatives in terms of the induced vector field, $p(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})$, are combined in product form to model $p(\dot{\mathbf{X}}_{n,\cdot}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})$. By analytically marginalising the state derivatives the following sampling scheme provides samples from $p(\boldsymbol{\theta}, \mathbf{X}, \boldsymbol{\varphi}, \boldsymbol{\sigma}, \boldsymbol{\gamma}|\mathbf{Y})$ (see Appendix C),

$$p(\boldsymbol{\varphi}_n, \sigma_n|\mathbf{Y}_{n,\cdot}) \propto \pi(\sigma_n)\pi(\boldsymbol{\varphi}_n)\mathcal{N}(\mathbf{Y}_{n,\cdot}|\mathbf{0}, \sigma_n^2\mathbf{I} + \mathbf{C}_{\boldsymbol{\varphi}_n}) \quad \forall \; n = 1 \cdots N \tag{19}$$

$$p(\mathbf{X}_{n,\cdot}|\mathbf{Y}_{n,\cdot}, \sigma_n, \boldsymbol{\varphi}_n) = \mathcal{N}(\mathbf{X}_{n,\cdot}|\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) \quad \forall \; n = 1 \cdots N \tag{20}$$

$$p(\boldsymbol{\theta}, \boldsymbol{\gamma}|\mathbf{X}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) \propto \exp\left(-U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma})/2\right)\pi(\boldsymbol{\gamma})\pi(\boldsymbol{\theta}) \tag{21}$$

where $U(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{\sigma}) = \sum_{n=1}^{N}(\mathbf{f}_n - \mathbf{m}_n)^{\mathsf{T}}(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}(\mathbf{f}_n - \mathbf{m}_n)$, with $\mathbf{f}_n$ denoting $\mathbf{f}_n(\mathbf{X}, \boldsymbol{\theta}, \mathbf{t})$. It is clear that Metropolis sampling is required for (19) and (21). Given the complexity of the induced likelihood function for nonlinear differential equations with respect to the structural parameters $\boldsymbol{\theta}$, see Ramsay *et al.* (2007) and Calderhead *et al.* (2009), an RM-HMC sampling scheme is now considered for both (19) and (21).

7.2. Metric Tensor and Derivatives for Systems of Nonlinear Differential Equations

To implement RM-HMC for (21) we require the metric tensor and its derivatives with respect to the target parameters. In Appendix D it is shown that an approximate form for the Fisher Information for the parameters $\boldsymbol{\theta}$ and hence the metric tensor is $\mathbf{G}(\boldsymbol{\theta}) \approxeq \sum_{n=1}^{N} \mathbf{F}_n(\mathbf{K}_n + \mathbf{I}\gamma_n)^{-1}\mathbf{F}_n^{\mathsf{T}}$ and the required elements of the derivative of the above metric tensor follow

$$\frac{\partial g_{d,d'}}{\partial \theta_i} = \sum_{n,s,s'} \kappa_n^{s,s'} \left(\frac{\partial^2 \mathbf{f}_{n,s}}{\partial \theta_i \partial \theta_d} \frac{\partial \mathbf{f}_{n,s'}}{\partial \theta_{d'}} + \frac{\partial \mathbf{f}_{n,s'}}{\partial \theta_d} \frac{\partial^2 \mathbf{f}_{n,s}}{\partial \theta_i \partial \theta_{d'}} \right) \tag{22}$$