Tesis Doctoral / Doktorego Tesia

# Enhancing Sampling in Computational Statistics Using Modified Hamiltonians


*Autora / Egilea:*

Tijana Radivojević

*Directores / Zuzendariek:*

Prof. Elena Akhmatskaya

Prof. Enrico Scalas



eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

2016

Doctoral Thesis

# Enhancing Sampling in Computational Statistics Using Modified Hamiltonians

*Author:*
Tijana Radivojević

*Supervisors:*
Prof. Elena Akhmatskaya
Prof. Enrico Scalas


2016

This research was carried out at the Basque Center for Applied Mathematics (BCAM) within the Group Modelling and Simulation in Life and Materials Sciences. The research was supported by the Spanish Ministry of Education, Culture and Sport (MECD) within the FPU-2012 program (Formación del Profesorado Universitario) under Grant FPU12/05209 and also by the Basque Government through the BERC 2014-2017 program and by the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SEV-2013-0323 and Grant *Retos en integración numérica: de las estructuras algebráicas a simulaciones Montecarlo* MTM2013–46553–C3–1–P.

# Abstract

The Hamiltonian Monte Carlo (HMC) method has been recognized as a powerful sampling tool in computational statistics. In this thesis, we show that performance of HMC can be dramatically improved by replacing Hamiltonians in the Metropolis test with modified Hamiltonians, and a complete momentum update with a partial momentum refreshment. The resulting generalized HMC importance sampler, which we called Mix & Match Hamiltonian Monte Carlo (MMHMC), arose as an extension of the Generalized Shadow Hybrid Monte Carlo (GSHMC) method, previously proposed for molecular simulation. The MMHMC method adapts GSHMC specifically to computational statistics and enriches it with new essential features: (i) the efficient algorithms for computation of modified Hamiltonians; (ii) the implicit momentum update procedure and (iii) the two-stage splitting integration schemes specially derived for the methods sampling with modified Hamiltonians. In addition, different optional strategies for momentum update and flipping are introduced as well as algorithms for adaptive tuning of parameters and efficient sampling of multimodal distributions are developed. MMHMC has been implemented in the in-house software package HaiCS (**Ha**miltonians **i**n **C**omputational **S**tatistics) written in C, tested on the popular statistical models and compared in sampling efficiency with HMC, Generalized Hybrid Monte Carlo, Riemann Manifold Hamiltonian Monte Carlo, Metropolis Adjusted Langevin Algorithm and Random Walk Metropolis-Hastings. The analysis of time-normalized effective sample size reveals the superiority of MMHMC over popular sampling techniques, especially in solving high-dimensional problems.

# Summary

Both academia and industry have been witnessing exponential accumulation of data, which, if carefully analyzed, potentially can provide valuable insights about the underlying processes. The main feature of the resulting problems is uncertainty, which appears in many forms, either in data or assumed data models, and causes great challenges. Therefore, we need more complex models and advanced analysis tools to deal with the size and complexity of data.

The Bayesian approach offers a rigorous and consistent manner of dealing with uncertainty and provides a means of quantifying the uncertainty in our predictions. It also allows us to objectively discriminate between competing model hypotheses, through the evaluation of Bayes factors (Gelman et al., 2003). Nevertheless, the application of Bayesian framework to complex problems runs into a computational bottleneck that needs to be addressed with efficient inference methods. The challenges arise in the e.g. evaluation of intractable quantities or large-scale inverse problems linked to computationally demanding forward problems, and especially in high dimensional settings. Theoretical and algorithmic developments in computational statistics have been leading to the possibility of undertaking more complex applications by scientists and practitioners, but sampling in high dimensional problems and complex distributions is still challenging.

Various methods are being used to practically address these difficulties, which technically reduce to the calculation of integrals, as the core of a Bayesian inference procedure. These integrals appear either in evaluating an expected value over some complex posterior distribution or in determining the marginal likelihood of a distribution. For example, we are interested in computing an expected value (or some other moment) of a function $f$ with respect to a distribution $\pi$

$$I = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{1}$$

Deterministic methods aim to find analytically tractable approximations of the distributions $\pi$. We are interested however in Monte Carlo (stochastic) methods (Metropolis and Ulam, 1949), which can sample from the desired distribution, are exact in the limit of infinite number of samples, and can achieve an arbitrary level of accuracy by drawing as many samples as one requires. The integral (1) in this case is estimated as

$$\hat{I} = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{\theta}^n),$$

where random samples $\{\boldsymbol{\theta}^n\}_{n=1}^N$ are drawn from $\pi(\boldsymbol{\theta})$.

Bayesian statistics have been revolutionized by ever-growing computational capacities and development of Markov chain Monte Carlo (MCMC) techniques (Brooks et al., 2011). In this thesis, we focus on an advanced MCMC methodology, namely Hamiltonian (or hybrid) Monte Carlo (HMC) (Duane et al., 1987; Neal, 2011), which arose as a good candidate for dealing with high-dimensional and complex problems. Interestingly, both MCMC and HMC were originally developed in statistical mechanics; however, they made a significant impact on the statistical community almost 35 and 10 years later, respectively.

The HMC method has proved to be a successful and valuable technique for a range of problems in computational statistics. It belongs to the class of auxiliary variable samplers, which defines an augmented target distribution through the use of a Hamiltonian function. HMC incorporates the gradient information of the target $\pi(\boldsymbol{\theta})$ and can follow this gradient over considerable distances. This is achieved by means of generating Hamiltonian trajectories (integrating Hamiltonian equations of motion), which are rejected or accepted based on the Metropolis test. As a result, HMC suppresses the random walk behavior typical of the Metropolis-Hastings Monte Carlo method (Metropolis et al., 1953; Hastings, 1970).

On the other hand, the performance of HMC deteriorates exponentially, in terms of acceptance rates, with respect to the system's size and the step size due to errors introduced by numerical approximations (Izaguirre and Hampton, 2004). Many rejections induce high correlations between samples and reduce the efficiency of the estimator of (1). Thus, in systems with large numbers of parameters, or latent parameters, or when the data set of observations is very large, efficient sampling might require a substantial number of evaluations of the posterior distribution and its gradient. This may be computationally too demanding for HMC. To maintain the acceptance rate for larger systems at a high level, one should either decrease the step size or use a higher order numerical integrator, which is usually impractical for large systems.

Ideally, one would like to have a sampler that increases acceptance rates, converges fast, improves sampling efficiency and whose optimal simulation parameters are not difficult to determine.

In this thesis we develop, test and analyze the HMC based methodology that enhances the sampling performance of the HMC method. We introduce a new approach, called Mix & Match Hamiltonian Monte Carlo (MMHMC), which arose as an extension of the Generalized Shadow Hybrid Monte Carlo (GSHMC) method by Akhmatskaya and Reich (2008). GSHMC was proposed for molecular simulation and has been published, patented and successfully tested on complex biological systems. As GSHMC, the MMHMC method samples with modified Hamiltonians, but it enriches GSHMC with the new essential features and adapts it specifically to computational statistics.

To the best of our knowledge, this is the first time that the method sampling with modified Hamiltonians has been implemented and applied to Bayesian inference problems in computational statistics.

The MMHMC method can be defined as a generalized HMC importance sampler. It

offers an update of momentum variables in a general form and samples with respect to a modified distribution that is determined through modified Hamiltonians. In particular, the method involves two major steps, the Partial Momentum Monte Carlo step, and the Hamiltonian Dynamics Monte Carlo (HDMC) step. The partial momentum update adds a random noise, controlled by an additional parameter, to the current momentum variable and accepts this update through the modified Metropolis test. As is the case with HMC, in the HDMC step a proposal state is generated by integrating Hamiltonian equations of motion and accepted according to the Metropolis test. The only difference in the HDMC step of MMHMC from the one in HMC is that in the Metropolis test the modified Hamiltonian is used instead of the true Hamiltonian. This leads to higher acceptance rates of MMHMC, as symplectic numerical integrators preserve modified Hamiltonians to a higher accuracy than the true Hamiltonian. Since sampling is performed with respect to the modified distribution, the importance weights are taken into account when estimating integral (1).

Within this thesis, we provide new formulations of modified Hamiltonians of 4th and 6th order for the splitting integrating schemes, which include families of two-, three- and four-stage integrators, recently proposed in the literature for improving the accuracy of numerical integration. The newly derived modified Hamiltonians are defined either through analytical derivatives of the potential function or numerical time derivatives of its gradient, which are computed from the quantities accessible during the simulation. We consider the former formulation being appropriate for sparse Hessian matrices of the potential and the latter, although including additional integration steps, are beneficial for cases where higher order derivatives are computationally demanding.

The novel numerical integrators from the two- and three-stage families of splitting integrators and specific to sampling with modified Hamiltonians are derived. We design new integrators by minimizing either error in modified Hamiltonian introduced due to numerical integration or its expected value, taken with respect to the modified density. With a high dimensional Gaussian model problem, two-stage integrators demonstrate a remarkable improvement over the commonly used Verlet integrator, both in terms of acceptance rates and sampling efficiency, over a range of simulation parameters. Moreover, the improvement increases with dimension and comes at no additional computational cost. Our recommendation is to use the new two-stage integrators instead of Verlet for high dimensional problems.

We also propose a computationally effective Metropolis test for momentum update and show that its use can potentially reduce computational time by 60%. In addition, different alternative strategies for momentum update, including transformation of momenta variables and several repetitive momentum update schemes are investigated. We implement, test and analyze these strategies but do not find any benefit from these formulations whatsoever.

Further on, we adapt the reduced momenta flipping technique (Wagoner and Pande,

2012) to MMHMC, which potentially can improve sampling. While in molecular simulations a momentum flip can indeed have a negative impact on dynamics, in computational statistics there is no clear evidence regarding a harmful influence on the sampling performance. Nevertheless, having implemented the statistically rigorous though an optional tool for reduced flipping can help to collect the information on the role of a momentum flip in MMHMC.

Considering ideas used for designing the MMHMC method, one could expect the following advantages over HMC: (i) high acceptance rates (due to better conservation of modified Hamiltonians by symplectic integrators than true Hamiltonian); (ii) access to second-order information about the target distribution and (iii) an extra parameter for improving the performance. These advantages come with an expense in terms of (i) a reduced efficiency of an estimator of the integral (1) due to importance sampling and (ii) a larger computational cost, consisting of the computation of modified Hamiltonian for each proposal (higher orders being even more expensive) and extra Metropolis test for momentum update.

Several extensions to the MMHMC method are proposed in this thesis. We first adapt MMHMC to sampling of constrained variables. We then devise two algorithms for automatic adaptation of MMHMC simulation parameters using Bayesian optimization approach in order to reduce the efforts of manual tuning. We also formulate the parallel tempering MMHMC method, whose benefits are twofold. Firstly, due to the use of an ensemble of chains it improves mixing and enables sampling from the multimodal probability distributions. Secondly, it provides samples from all required power posteriors simultaneously, which then can be used for estimation of the marginal likelihood, as we also describe.

We develop the user-friendly software package written in C HaiCS (**Ha**miltonians **i**n **C**omputational **S**tatistics) targeted to computers running UNIX certified operating systems. The code is intended for statistical sampling of high dimensional and complex distributions and parameter estimation in different models through Bayesian inference using Hamiltonian Monte Carlo based methods. The currently available sampling techniques include HMC, Generalized Hamiltonian Monte Carlo (GHMC), Metropolis Adjusted Langevin Algorithm (MALA), second order Langevin Monte Carlo (L2MC) and Mix & Match Hamiltonian Monte Carlo, the method developed in this thesis.

The package benefits from efficient implementation of modified Hamiltonians, the accurate multi-stage splitting integration schemes (as previously proposed as novel), the analysis tools compatible with CODA toolkit for MCMC diagnostics as well as the interface for implementing complex statistical models. The popular statistical models multivariate Gaussian distribution, Bayesian Logistic Regression (BLR) and Stochastic Volatility (SV) are implemented in HaiCS.

The MMHMC method has been carefully tested and compared with the traditional and advanced sampling techniques for computational statistics such as Random Walk Metropolis-Hastings, HMC, GHMC, MALA and Riemann Manifold Hamiltonian Monte Carlo (RMHMC). We examine the performance of these methods on a set of standard benchmark statistical models.

We inspect space exploration using an illustrative banana-shaped distribution. MMHMC accepts more proposals that result in better coverage of the space than with HMC. Although it uses the second-order information on the posterior, MMHMC does not follow its local curvature as obviously as it does RMHMC.

Acceptance rate is higher for MMHMC than for other methods consistently for all experiments.

Being a method that generates both correlated and weighted samples, MMHMC requires a metric for sampling efficiency different from the one commonly used for MCMC. Here we suggest a new metric for ESS estimation for samples drawn by MMHMC, which can also be employed for any MCMC importance sampling based method.

Our tests demonstrate that in terms of sampling efficiency MMHMC, HMC and GHMC perform comparably for small dimensional problems. In high dimensional problems however, when compared to HMC and GHMC, the MMHMC method demonstrates superior performance, in terms of bigger time-normalized ESS, for a range of applications, a range of dimensions and choice of simulation parameters. It allows for bigger step sizes to be used without decreasing acceptance rate; moreover, it achieves better performance for larger step sizes. The improvements increase with dimension – for a multivariate Gaussian problem MMHMC shows an improvement over HMC of up to remarkable 40 times and for the BLR model up to 4 times. We expect even higher enhancement for problems of higher dimensions, as the new integrators specifically designed for MMHMC are particularly beneficial for high dimensional problems. An additional advantage of MMHMC lays in the fact that it is less sensitive than HMC to the choice of a number of integration steps. The SV model experiments demonstrate the clear superiority of MMHMC and RMHMC over the HMC and GHMC methods. The sampling performance of MMHMC and RMHMC is comparable for this benchmark. Nevertheless, in contrast to the original RMHMC, MMHMC does not require higher order derivative and inverse of the metric and thus is computationally less expensive. This issue becomes particularly important for high-dimensional problems with dense Hessian matrix. Besides, choices of integrators for RMHMC are limited due to the use of non-separable Hamiltonians, whereas MMHMC allows for the use of the novel efficient numerical integrators.

**The structure of the thesis** We begin with introducing our motivation for the development of efficient sampling techniques in computational statistics and reviewing some basic methods in Chapter 1. Chapter 2 provides details of the HMC methodology and an outlook on the further developments in computational statistics and computational sciences. In Chapter 3 we present the novel Mix & Match Hamiltonian Monte Carlo (MMHMC) method and a number of different strategies that can be employed within. These include (i) new formulations of modified Hamiltonians; (ii) novel multi-stage numerical integrators, as alternatives to the Verlet integrator; (iii) different strategies for momenta update and flipping. Several extensions to MMHMC are designed in Chapter 4. In particular, we formulate a parallel tempering algorithm for efficient multimodal sampling that utilizes MMHMC as an underlying sampler. An algorithm for Bayesian adaptation of MMHMC parameters is also

proposed. In addition, we discuss the estimation of the marginal likelihood using MMHMC and formulate sampling of constrained parameters in the context of the MMHMC method. In Chapter 5 we describe the software package developed along this thesis in which the novel MMHMC has been implemented. Testing and comparison of MMHMC with popular sampling techniques are provided in Chapter 6. Chapter 7 summarizes contributions made by this thesis and outlines some future directions of research that can follow from the thesis. Finally, Appendix provides a list of contributions in model and algorithm development that I have made during my Ph.D. program.

# Resumen

Tanto el ámbito académico como el industrial han sido testigos de una acumulación exponencial de datos que, analizándolos con atención, pueden proporcionar valiosas aportaciones sobre los procesos subyacentes. La principal característica de los problemas derivados es la incertidumbre en sus diferentes variantes, bien en forma de datos o bien en supuestos modelos de datos, planteando grandes retos. Por lo tanto, se necesitan modelos más complejos y herramientas de análisis avanzadas para gestionar el tamaño y la complejidad de los datos.

El enfoque bayesiano permite afrontar la incertidumbre de forma rigurosa y coherente, tratándose de una herramienta para cuantificar dicha incertidumbre en nuestras predicciones. Nos permite también discriminar objetivamente hipótesis de modelos competidores a través de la evaluación de los factores de Bayes (Gelman et al., 2003). Sin embargo, la aplicación del marco bayesiano a problemas complejos deriva en un cuello de botella computacional que hace que sea necesario abordarlos utilizando métodos eficaces de inferencia. Los retos aparecen p. ej. en la evaluación de cantidades difíciles de tratar o problemas inversos a gran escala ligados a problemas prospectivos exigentes desde el punto de vista computacional, y especialmente en los ajustes dimensionales altos. Los desarrollos teóricos y algorítmicos en estadística computacional han permitido a científicos y profesionales asumir aplicaciones más complejas, aunque todavía sigue siendo un reto el muestreo en problemas de grandes dimensiones y distribuciones complejas.

Se utilizan varios métodos para resolver de manera práctica estas dificultades que técnicamente se reducen al cálculo de integrales, como núcleo del procedimiento de inferencia bayesiano. Estas integrales aparecen tanto al evaluar un valor esperado con respecto a algunas distribuciones posteriores complejas, como también al determinar la probabilidad marginal de una distribución. Por ejemplo, estamos interesados en computar un valor esperado (u otro momento) de una función $f$ con respecto a una distribución $\pi$

$$I = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{2}$$

El objetivo de los métodos determinísticos es buscar aproximaciones analíticamente manejables de las distribuciones $\pi$. No obstante, nuestro interés se centra en los métodos Monte Carlo (estocásticos) (Metropolis and Ulam, 1949), que permiten el muestreo desde la distribución deseada, son exactos en el limite de un numero infinito de muestras y logran un

nivel arbitrario de precisión tomando tantas muestras como sea necesario. La integral (2) en este caso se estima como sigue

$$\hat{I} = \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{\theta}^n),$$

donde las muestras aleatorias $\{\boldsymbol{\theta}^n\}_{n=1}^N$ se toman desde $\pi(\boldsymbol{\theta})$.

La estadística bayesiana ha experimentado una revolución gracias a unas capacidades computacionales cada vez mayores y al desarrollo de técnicas Markov chain Monte Carlo (MCMC) (Brooks et al., 2011). En esta tesis nos centramos en la metodología MCMC avanzada, concretamente en el método hamiltoniano (o híbrido) de Monte Carlo (HMC, por sus siglas en inglés) (Duane et al., 1987; Neal, 2011), que nació como un buen candidato para abordar problemas complejos de grandes dimensiones. Curiosamente, los métodos MCMC y HMC se desarrollaron originalmente en la mecánica estadística aunque no tuvieron gran impacto en la comunidad estadística hasta 35 y 10 años después, respectivamente.

El método HMC ha demostrado ser exitoso además de una técnica valiosa para una serie de problemas en la estadística computacional. Pertenece a la clase de muestreadores de variables auxiliares, que definen una distribución proyectada aumentada mediante la utilización de una función hamiltoniana. El HMC incorpora la información de gradiente del $\pi(\boldsymbol{\theta})$ objetivo y es capaz de seguir este gradiente a grandes distancias. Esto se logra mediante la generación de trayectorias hamiltonianas (integrando ecuaciones de movimiento de Hamilton), que son rechazadas o aceptadas en base a un test de Metropolis. Como resultado, el HMC suprime el comportamiento de recorrido aleatorio típico del método Metropolis-Hastings Monte Carlo (Metropolis et al., 1953; Hastings, 1970).

Por otra parte, el rendimiento del HMC se deteriora exponencialmente en términos de tasas de aceptación con respecto al tamaño del sistema y al tamaño del paso, debido a los errores introducidos por las aproximaciones numéricas (Izaguirre and Hampton, 2004). En caso de muchos rechazos se producen altas correlaciones entre las muestras y se reduce la eficiencia del estimador de (2). Por consiguiente, en sistemas con muchos parámetros, o parámetros latentes, o cuando el conjunto de datos de las observaciones es muy grande, para que el muestreo sea eficiente puede que se requiera un número sustancial de evaluaciones de la distribución posterior y de su gradiente. Desde el punto de vista computacional, dichas evaluaciones pueden resultar demasiado exigentes para el HMC. Para mantener el alto nivel de la tasa de aceptación en sistemas más grandes, se debería disminuir el tamaño del paso o utilizar un integrador numérico de orden superior, hecho que no suele resultar práctico en sistemas grandes.

Lo ideal sería disponer de un muestreador que aumente las tasas de aceptación, converja rápido, mejore la eficiencia de muestreo y cuyos parámetros de simulación opcionales no sean difíciles de determinar.

En la presente tesis hemos desarrollado, comprobado y analizado la metodología basada en HMC que aumenta el rendimiento de muestreo del método HMC. Introducimos un nuevo

enfoque, el denominado Mix & Match Hamiltonian Monte Carlo o MMHMC (Hamiltoniano Combinado de Monte Carlo), que surgió como ampliación del método Generalized Shadow Hybrid Monte Carlo o GSHMC (Monte Carlo Híbrido generalizado con hamiltoniano *shadow*) por parte de Akhmatskaya and Reich (2008). El GSHMC se propuso para la simulación molecular y se ha publicado, patentado y comprobado con éxito en sistemas biológicos complejos. Al igual que el GSHMC, el método MMHMC realiza el muestreo con hamiltonianos modificados (o *shadow*) pero enriquece el GSHMC con nuevas características fundamentales, adaptándolo específicamente a la estadística computacional.

Según nuestros datos, se trata de la primera vez en la que se ha implementado y aplicado el método de muestreo con hamiltonianos modificados a problemas de inferencia bayesianos en estadística computacional.

El método MMHMC puede definirse como muestreador de importancia (*importance sampling*) HMC generalizado. Ofrece la actualización general de las variables de cantidad de movimiento y realiza muestreos con respecto a la distribución modificada que se define a través de los hamiltonianos modificados. Este método contempla concretamente dos pasos principales: el paso de Cantidad de Movimiento Parcial de Monte Carlo y el paso de Dinámica Hamiltoniana de Monte Carlo (HDMC). La actualización de la cantidad de movimiento parcial añade un ruido aleatorio, controlado por un parámetro adicional, a la variable actual de cantidad de movimiento y acepta esta actualización a través de la prueba Metropolis modificada. Al igual que con el HMC, en el paso HDMC se genera un estado de propuesta integrando ecuaciones de movimiento hamiltonianas, aceptado a su vez de acuerdo con el test de Metropolis. La única diferencia en el paso HDMC del MMHMC con respecto al del HMC es que en la prueba Metropolis se utiliza el hamiltoniano modificado en vez del hamiltoniano real. En consecuencia, el MMHMC ofrece unas tasas de aceptación superiores porque los integradores numéricos conservan los hamiltonianos modificados con más precisión que en el caso del hamiltoniano real. Como el muestreo se realiza con respecto a la distribución modificada, los pesos de importancia se tienen en cuenta al estimar la integral (2).

En esta tesis presentamos nuevas formulaciones de los hamiltonianos modificados de 4.º y 6.º orden para los esquemas de integración de división, que incluyen familias de integradores de dos, tres y cuatro etapas, propuestas recientemente en la bibliografía para mejorar la precisión de la integración numérica. Los nuevos hamiltonianos modificados derivados están bien definidos a través de derivadas analíticas de la función potencial o de derivadas temporales numéricas de su gradiente, que se computan a partir de las cantidades accesibles durante la simulación. Consideramos que la anterior formulación es apropiada para matrices hessianas dispersas del potencial; las siguientes, aunque incluyen pasos de integración adicionales, son favorables para casos en los que las derivadas de orden superior son exigentes desde el punto de vista computacional.

Nuevos integradores numéricos de las familias de dos y tres etapas de integradores de división y específicos para muestreos con hamiltonianos modificados son derivados. Hemos

diseñado nuevos integradores minimizando el error en el hamiltoniano modificado introducido debido a la integración numérica o minimizando su valor esperado, tomado con respecto a la densidad modificada. Ante un problema de modelo gaussiano de grandes dimensiones, los integradores de dos etapas demostraron mejoras considerables en comparación con el integrador de Verlet comúnmente utilizado, tanto en términos de tasas de aceptación como en eficiencia de muestreo, sobre una amplia gama de parámetros de simulación. Asimismo, esta mejora aumenta junto con la dimensión y no produce costes computacionales adicionales. Para problemas de grandes dimensiones, recomendamos utilizar los nuevos integradores de dos etapas en lugar del integrador de Verlet.

Proponemos también una prueba Metropolis eficaz en términos computacionales para la actualización de la cantidad de movimiento y demostramos que su uso puede reducir potencialmente el tiempo computacional en un 60%. Además, se están investigando diferentes estrategias alternativas para la actualización de la cantidad de movimiento, incluída la transformación de variables de cantidad de movimiento y varios esquemas repetitivos para la actualización de la cantidad de movimiento. Hemos implementado, comprobado y analizado estas estrategias, pero no hemos detectado ninguna mejora con respecto a la formulación original.

Posteriormente, hemos adaptado la técnica de inversión de la cantidad de movimiento (Wagoner and Pande, 2012) al MMHMC, que potencialmente puede mejorar el muestreo. Mientras que en las simulaciones moleculares una inversión de la cantidad de movimiento puede tener un impacto negativo sobre la dinámica, en la estadística computacional no hay evidencia clara de la influencia negativa sobre el rendimiento de muestreo. Sin embargo, habiendo implementado el rigor estadístico, una herramienta opcional para inversión reducida puede ayudar a recopilar información sobre el rol de la inversión de la cantidad de movimiento en el MMHMC.

Teniendo en cuenta las ideas a partir de las cuales se ha diseñado el método MMHMC, se podrían esperar las siguientes ventajas con respecto al HMC: (i) altas tasas de aceptación (debido a que los integradores simplécticos conservan mejor los hamiltonianos en comparación al hamiltoniano real); (ii) acceso a información de segundo orden sobre la distribución objetivo y (iii) un parámetro adicional para mejorar el rendimiento. Estas ventajas conllevan un gasto en términos de (i) eficiencia reducida de un estimador de la integral (2) debido al *importance sampling* (ii) y un coste computacional más alto que consiste en la computación del hamiltoniano modificado para cada propuesta (siendo los órdenes superiores incluso más caros) y en una prueba Metropolis adicional para la actualización de la cantidad de movimiento.

En esta tesis se proponen varias ampliaciones con respecto al método MMHMC. En primer lugar hemos adaptado el MMHMC para el muestreo de variables dependientes. Posteriormente hemos ideado dos algoritmos para la adaptación automática de los parámetros de simulación del MMHMC utilizando un enfoque de optimización bayesiano para reducir los esfuerzos de la sintonización manual. Hemos formulado también el método MMHMC de temple paralelo que ofrece una doble ventaja. En primer lugar, como se utiliza un conjunto

de cadenas, mejora la mezcla y permite el muestreo desde distribuciones de probabilidad multimodal. En segundo lugar, ofrece simultáneamente muestras desde todas las posteriores de potencia requeridas, que después pueden ser utilizadas para la estimación de la probabilidad marginal, tal y como describimos.

Hemos desarrollado un paquete de software intuitivo y fácil de utilizar en C HaiCS (**Ha**miltonians **i**n **C**omputational **S**tatistics) dirigido a ordenadores con sistemas operativos certificados UNIX. El código está previsto para el muestreo estadístico de distribuciones complejas y de grandes dimensiones, así como la estimación de parámetros en diferentes modelos a través de la inferencia bayesiana utilizando métodos basados en el Hamiltoniano de Monte Carlo. Las técnicas de muestreo actualmente disponibles incluyen el HMC, el Generalized Hamiltonian Monte Carlo o GHMC (Hamiltoniano Generalizado de Monte Carlo), Metropolis Adjusted Langevin Algorithm o MALA (Algoritmo de Langevin Ajustado de Metropolis), el Langevin Monte Carlo de segundo orden (L2MC) y el Mix & Match Hamiltonian Monte Carlo (Hamiltoniano Combinado de Monte Carlo), el método desarrollado en esta tesis.

El paquete incluye ventajas como la implantación eficiente de hamiltonianos modificados, los precisos esquemas de integración de división multietapa (propuestos anteriormente como novedosos), las herramientas de análisis compatibles con la serie de herramientas CODA para diagnósticos MCMC, así como la interfaz para implementar modelos estadísticos complejos. La popular distribución gaussiana multivariante de modelos estadísticos, la regresión logística bayesiana (BLR) y la volatilidad estocástica (SV) se implementan en HaiCS.

El método MMHMC ha sido exhaustivamente comprobado y comparado con técnicas de medición tradicionales y avanzadas para la estadística computacional, tales como el Random Walk Metropolis-Hastings (Recorrido Aleatorio de Metropolis-Hastings), HMC, GHMC, MALA y Riemann Manifold Hamiltonian Monte Carlo o RMHMC (Hamiltoniano de Monte Carlo con variedades de Riemann). El rendimiento de estos métodos ha sido examinado en un conjunto de modelos estadísticos de referencia estándar.

Inspeccionamos la exploración del espacio utilizando una distribución ilustrativa en forma de plátano. El MMHMC acepta más propuestas, obteniendo una mejor cobertura del espacio en comparación con el HMC. Aunque emplea información de segundo orden en la probabilidad a posteriori, el MMHMC no sigue su curvatura local con tanta evidencia como lo hace el RMHMC.

Sistemáticamente las tasas de aceptación son más altas para el MMHMC que para otros métodos para todos los experimentos.

Al tratarse de un método que genera muestras tanto correlativas como ponderadas, el MMHMC requiere una métrica para la eficiencia de muestreo diferente a la utilizada comúnmente para el MCMC. Sugerimos una nueva métrica para la estimación ESS para muestras tomadas mediante el MMHMC, que también puede utilizarse para cualquier método basado en el muestreo por importancia MCMC.

Nuestra tesis demuestra que en términos de eficiencia de muestreo, los métodos MMHMC,

HMC y GHMC actúan de forma comparable en caso de pequeños problemas dimensionales. No obstante, en grandes problemas dimensionales, en comparación con los métodos HMC y GHMC, el método MMHMC demuestra un rendimiento superior en términos de ESS normalizado en tiempo más alto para una amplia gama de aplicaciones y dimensiones, así como para la elección de parámetros de simulación. Permite utilizar tamaños más grandes de paso sin disminuir la tasa de aceptación, mejorando a su vez el rendimiento en pasos más largos. Las mejoras aumentan con la dimensión: para un problema gaussiano multivariante, el MMHMC muestra una mejora con respecto al HMC de hasta 40 veces; para el modelo BLR es de hasta 4 veces. Esperamos mejoras incluso superiores en el caso de problemas de grandes dimensiones, ya que los nuevos integradores diseñados específicamente para el MMHMC resultan especialmente adecuados para problemas de grandes dimensiones. Una ventaja adicional del MMHMC radica en que es menos sensible que el HMC al elegir el número de pasos de integración. Los experimentos con modelos SV demuestran claramente la superioridad del MMHMC y del RMHMC con respecto a los métodos HMC y GHMC. El rendimiento de muestreo del MMHMC y del RMHMC es comparable para esta cota de referencia. Sin embargo, en comparación con el RMHMC original, el MMHMC no requiere ninguna derivada de orden superior ni inversión de la métrica, por lo que es menos caro desde el punto de vista computacional. Este aspecto adquiere especial relevancia en grandes problemas dimensionales con matriz hessiana densa. Además, las opciones para incorporar integradores en el RMHMC son limitadas ya que se utilizan hamiltonianos que no se pueden separar; el MMHMC, por su parte, permite el uso de los novedosos y eficientes integradores numéricos.

**Estructura de la tesis** En primer lugar presentamos los motivos que nos han llevado a desarrollar técnicas de muestreo eficientes en la estadística computacional y revisamos algunos métodos básicos en el capítulo 1. En el capítulo 2 se detalla la metodología HMC y se ofrece una perspectiva de los desarrollos posteriores llevados a cabo en la estadística computacional y en las ciencias computacionales. En el capítulo 3 presentamos el novedoso método Hamiltoniano Combinado de Monte Carlo (MMHMC), junto con diferentes estrategias que pueden utilizarse, entre las que se incluyen (i) nuevas formulaciones de hamiltonianos modificados; (ii) novedosos integradores numéricos multietapa, como alternativa al integrador de Verlet; (iii) diferentes estrategias para la inversión y actualización de la cantidad de movimiento. En el capítulo 4 se diseñan varias extensiones para el MMHMC. En particular, formulamos un algoritmo de temple paralelo para el muestreo multimodal eficiente que utiliza el MMHMC como muestreador subyacente. Se propone también un algoritmo para la adaptación bayesiana de los parámetros del MMHMC. Asimismo, también abordamos la estimación de la probabilidad marginal utilizando el MMHMC y formulamos el muestreo de parámetros dependientes en el contexto del método MMHMC. En el capítulo 5 describimos el paquete de software desarrollado a lo largo de esta tesis, en el que se ha implementado el novedoso MMHMC. Las pruebas y comparaciones del método MMHMC con otras técnicas populares de muestreo se reflejan en el capítulo 6. El capítulo 7 resume las contribuciones de esta tesis y describe las futuras vías de investigación que pueden dar

continuidad a la presente. Finalmente, en el Anexo se incluye una lista de contribuciones realizadas al desarrollo de modelos y algoritmos que he confeccionado durante mi doctorado.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **MC** | **M**onte **C**arlo |
| **MCMC** | **M**arkov **C**hain **M**onte **C**arlo |
| **HMC** | **H**ybrid/**H**amiltonian **M**onte **C**arlo |
| **p.d.f** | **p**robability **d**ensity **f**unction |
| **BCH** | **B**aker-**C**ampbell-**H**ausfdorff |
| **ME** | **M**inimum **E**rror |
| **BCSS** | **B**lanes-**C**asas-**S**anz-**S**erna |
| **RMHMC** | **R**iemann **M**anifold **H**amiltonian **M**onte **C**arlo |
| **GHMC** | **G**eneralized **H**ybrid **M**onte **C**arlo |
| **MDMC** | **M**olecular **D**ynamics **M**onte **C**arlo |
| **PMU** | **P**artial **M**omenta **U**pdate |
| **MD** | **M**olecular **D**ynamics |
| **GSHMC** | **G**eneralized **S**hadow **H**ybrid **M**onte **C**arlo |
| **PMMC** | **P**artial **M**omenta **M**onte **C**arlo |
| **LD** | **L**angevin **D**ynamics |
| **GSHmMC** | **G**eneralized **S**hadow **Ham**iltonian **M**onte **C**arlo |
| **MMHMC** | **M**ix (&) **M**atch **H**amiltonian **M**onte **C**arlo |
| **HD** | **H**amiltonian **D**ynamics |
| **AR** | **A**cceptance **R**ate |
| **ESS** | **E**ffective **S**ample **S**ize |
| **EF** | **E**fficiency **F**actor |
| **BLR** | **B**ayesian **L**ogistic **R**egression |
| **SV** | **S**tochastic **V**olatility |

# List of Symbols

| | |
|---|---|
| $\pi$ | target distribution (density) |
| $\boldsymbol{\theta}$ | $D$-dimensional vector of model parameters |
| $\mathbf{y}$ | $K$-dimensional vector of observed data |
| $\mathbf{p}$ | $D$-dimensional momenta vector |
| $\mathbf{x}$ | $D$-dimensional position vector |
| $H$ | Hamiltonian function |
| $U$ | potential function |
| $K$ | kinetic function |
| $M$ | 'mass' matrix / covariance matrix of momenta variables |
| $\mathcal{F}$ | flip operator |
| $h$ | step size |
| $L$ | number of integration steps |
| $\Phi_h, \varphi_h$ | exact flows |
| $\Psi_h$ | approximate flow |
| $\tilde{H}$ | modified Hamiltonian |
| $Z$ | normalizing constant / model evidence / marginal likelihood |
| $\Delta$ | energy error |
| $\tilde{\pi}$ | modified target distribution (density) |
| $\varphi$ | noise parameter |
| $w$ | importance weights |

*To my family – my sister Andrijana and my parents Nada and Cojo*

# 1

# Introduction

## 1.1 Motivation: Learning from data

The proliferation of innovations across various industries and their incorporation in our daily lives, together with advancements in technology and communication have enlightened a whole new arena for the buzzing business environment of the 21st century. What once was a computational system accessible only to scientists at prestigious universities and laboratories (e.g. in Pennsylvania, Princeton, Los Alamos, Manchester) or corporations (e.g. IBM, RAND)[1] is nowadays around 240 million[2] pieces of personal computers sold worldwide in 2015 only; what once was a status symbol of super-rich is nowadays around 70 million[3] cars produced in 2015 and what once was military-based communication system is nowadays the Internet with 3.5 billion[4] people able to connect to it.

These trends did not only affect improvement in the quality of life, but they have significantly influenced exponential accumulation of both scientific and commercial data. The data that surround us come from sources such as sensors (temperature, pressure), mobile devices (location, activity tracking), financial activity (transactions, stock market fluctuations), market scans (user preferences), Internet searches and many others. Fields such as aerospace, manufacturing, retail, pharmaceuticals, insurance and finance, public sector administration and academia have a common trait: they are all in the business of accumulating and analyzing the data.

Although the measuring techniques for capturing data have become more accurate, the wealth of data does come with *uncertainty*, arising for example from incompleteness

[1] http://www.computerhistory.org/timeline/computers/
[2] http://www.statisticbrain.com/computer-sales-statistics/
[3] http://www.statisticbrain.com/cars-produced-in-the-world/
[4] http://www.statista.com/statistics/273018/number-of-internet-users-worldwide/

due to a frequency of measurements, inaccuracy of devices or even impossibility to obtain measurements (e.g. clinical trials). Therefore, mining the valuable information from the overwhelming gigabytes of observed data and translating those findings into decision-making has become the imperative to secure competitive efficiency, desired profits and innovative prestige or advancement in science. Consequently, the new multi-disciplinary field of *data science* emerged, with the main focus to deal with uncertain data and the underlying processes. This core focus is equally applicable to problems in physics, genetics, econometrics, statistics, robotics, signal processing, geophysical and space exploration, (bio)engineering, astronomy, weather forecast, clinical trials, personalized medicine, risk prediction/management, uncertainty quantification, etc. which led to data scientists being in high demand across the broad spectrum of industries (Davenport and Patil, 2012).

With the main objective to *learn from the data* (meaning to extract information from the observations to learn about the underlying process that generated the data, *and* use that knowledge to make predictions for yet unobserved scenarios) experts are faced with many challenges.

The difficulties in modern data analysis lay in the fact that complex models are needed for data representation, and more and more sophisticated and robust algorithms that scale well both with those kinds of models and the size of the data are necessary to be developed. A natural way of dealing with the uncertain world of data is by employing *statistical modeling and algorithms*, which provide insights through the probabilistic approach. Those tools characterize uncertainty, both in the observed data and proposed data model, in a mathematically consistent and rigorous manner, but also allow us to quantify the uncertainty in our predictions. In particular, if we aim to infer parameters of a model, with this approach we can assess the variance and covariance structure of the estimated values.

As Green et al. (2015) put it

> We must retain a sense of the stochastic elements in data collection, data analysis, and inference, recognising uncertainty in data and models, to preserve the inductive strength of data science–seeing beyond the data we have to what it might have been, what it be next time, and where it came from.

Mathematically, the problem of learning about the data-generating process comes down to inferring unobserved variables given data. This problem is used under different terminology across fields: inverse problem, data assimilation, data mining, calibration, system identification, parameter estimation, statistical inference (frequentist or Bayesian), etc. A true data-generating process (model) is assumed to exist. In practice, we consider a small set of models which adequately approximate the true model, and do not necessarily include the true model. Inference is then the identification of the set of models in agreement with a particular set of observed data. Although the classical (frequentist) inference has been used extensively, in this thesis we adopt the Bayesian approach.

### 1.1.1 Bayesian statistics

Bayesian inference is a powerful methodology for dealing with complex statistical models used in a wide range of problems. It provides a consistent and principled way of incorporating uncertainties in observations, prior information or beliefs, and assumptions about the data model.

The groundwork for the modern Bayesian statistics had been set as early as in the 18th and 19th century by Bayes and Laplace. The cornerstone of the Bayesian methodology is the *Bayes' theorem*, named after Reverend Thomas Bayes, due to his essay (Bayes and Price, 1763)[5] on how humans can learn from experience by updating their beliefs as more data become available. It was Richard Price who discovered Bayes' unpublished work, recognized its importance and contributed to the publication. Laplace independently reinvented Bayes' principle (Laplace, 1820) and made probability theory applicable to many scientific and practical problems. Despite its early discovery, the Bayesian methodology was on hold almost for two centuries. Its uprising started with increasing computing power and development of the Markov chain Monte Carlo methods. Nowadays, when one can perform Bayesian data analysis on laptops, this approach has become common across many applied fields.

The Bayesian approach includes the knowledge before observing data through *prior distributions* and incorporates the observations through a *likelihood* to calculate *posterior* and *predictive probabilities* of different outcomes as refinements of our beliefs in light of those newly observed data. More concretely, for a considered model $m$, the Bayes' theorem computes the posterior (conditional) distribution of the (hidden) model parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$ given the (observed) data $\mathbf{y} = (y_1, \ldots, y_K)$

$$\underbrace{p(\boldsymbol{\theta}|\mathbf{y}, m)}_{\text{Posterior}} = \frac{\overbrace{p(\mathbf{y}|\boldsymbol{\theta}, m)}^{\text{Likelihood}}\overbrace{p(\boldsymbol{\theta}|m)}^{\text{Prior}}}{\underbrace{p(\mathbf{y}|m)}_{\text{Marginal Likelihood}}}. \tag{1.1}$$

The *posterior* distribution expresses the variability or uncertainty within model parameters after taking both the prior belief and observations into account. The *likelihood* (or likelihood function) of the data given model parameters accounts for errors in e.g. measurements and/or underlying models. Usually, it is assumed that errors are independent and identically distributed (i.i.d.) and therefore the likelihood appears as a product over all data points. The likelihood is a function (not a distribution) in which all variables are related in a full statistical model. The *prior* distribution incorporates prior beliefs and quantifies uncertainty about each model parameter. It is chosen before seeing the observed data and generally, may be categorized as informative or uninformative. The *marginal likelihood* normalizes the posterior distribution, ensuring it is a proper probability distribution and integrates to one. It is the probability of the data given the model and is obtained by

[5] Published posthumously by Price.

integrating over the parameter space

$$p(\mathbf{y}|m) = \int p(\mathbf{y}|\boldsymbol{\theta}, m)p(\boldsymbol{\theta}|m)\mathrm{d}\boldsymbol{\theta}. \tag{1.2}$$

For most statistical models of interest, the marginal likelihood involves an intractable integration problem, and therefore, it must be approximated.

Usually, one wishes to understand the uncertainty associated with each statistical model, or to use this uncertainty for estimation of quantities of interest or prediction of unobserved scenarios, or eventually, to discriminate between the proposed models. Bayesian inference is used for one or more of the following tasks.

**Marginalization** Given joint posterior of parameters $(\boldsymbol{\theta}, \phi)$, where $\phi$ can be a set of nuisance parameters included in the model, we may wish to marginalize parameters we are not interested in

$$p(\boldsymbol{\theta}|\mathbf{y}) = \int p(\boldsymbol{\theta}, \phi|\mathbf{y})\mathrm{d}\phi.$$

**Prediction** The likelihood of the predicted data $\mathbf{y}'$ can be evaluated using the *posterior predictive distribution*

$$p(\mathbf{y}'|\mathbf{y}) = \int p(\mathbf{y}'|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})\mathrm{d}\boldsymbol{\theta},$$

where the integral is taken over the posterior distribution. Data $\mathbf{y}'$ can play the role of any potential new data or an attempt to replicate the observed data $\mathbf{y}$ and make a comparison (Gelman et al., 2003). This task can be generalized as a calculation of the expected value of a function $f$ with respect to the posterior distribution given data $\mathbf{y}$

$$\mathbb{E}[f] = \int f(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})\mathrm{d}\boldsymbol{\theta}.$$

**Model selection** The task of selecting a model which explains data better than another model is performed by evaluating Bayes factor, a Bayesian alternative to hypothesis testing in frequentist statistics. Bayes factor is the ratio of model evidences for two competing models $m_1$ and $m_2$ given a particular dataset $\mathbf{y}$ (Kass and Raftery, 1995)

$$\frac{p(\mathbf{y}|m_1)}{p(\mathbf{y}|m_2)}.$$

The terms in Bayes factor are marginal likelihoods (1.2), also known as model evidence or normalizing constant (of a posterior distribution function).

Conclusions from Bayes factor of the model $m_1$ over model $m_2$ (as introduced by Jeffreys (1961)), given in terms of strength of evidence in favor of the model $m_1$, are summarized in Table 1.1.

| Bayes Factor | Strength of Evidence for $m_1$ |
|---|---|
| <1 | negative (supports $m_2$) |
| 1 to 3 | barely worth mentioning |
| 3 to 10 | substantial |
| 10 to 30 | strong |
| 30 to 100 | very strong |
| >100 | decisive |

Table 1.1: Interpretation of Bayes factor.

A common issue in these three tasks of Bayesian inference is the computation of high-dimensional and usually analytically intractable integrals. Our focus now moves to computational statistics, an active area in which efficient techniques for dealing with the challenges related to these tasks are being developed. For more details on Bayesian inference we refer the reader to e.g. Bernardo and Smith (1994) and Gelman et al. (2003) and the references therein.

## 1.2 Computational statistics

The core of a Bayesian inference procedure is the calculation of integrals, appearing either in evaluating an expected value over some posterior distribution or in determining the marginal likelihood of a distribution. For most problems of interest, these integrals are high dimensional and intractable and therefore, efficient techniques are required. In general, these integrals can be treated either using *deterministic* or *stochastic* (or Monte Carlo) methods. The former ones include techniques such as Laplace Approximation, Variational Bayes, etc. and the latter ones e.g. Approximate Bayesian Computation, Rejection Sampling, Importance Sampling, Markov chain Monte Carlo methods. We focus on the Monte Carlo methods that rely on statistical sampling – drawing samples from the desired distribution to evaluate integrals by estimators that converge to true solutions. Some problems may involve complex distributions that are extremely difficult to sample from, and thus, sophisticated methods are needed.

For an excellent review and significance of different methods of Bayesian computation in the era of data science, we refer the reader to Green et al. (2015).

### 1.2.1 Monte Carlo

Building the first electronic general-purpose computer ENIAC in the 1940s caused a rebirth of experimental mathematics and in particular, of statistical sampling through the use of the Monte Carlo (MC) method. Metropolis and Ulam (1949) published the MC method for the first time although Enrico Fermi invented it and used it in the 1930s independently but never published it (Metropolis, 1987).

Monte Carlo methods represent a large class of algorithms that repeatedly draw random samples to estimate an integral of interest by a sample average estimator. More concretely, we aim at generating samples from the desired distribution $\pi$, or in general, computing an expected value (or some other moment) of a function $f$ with respect to the distribution $\pi$

$$I = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{1.3}$$

We assume $\pi$ is some *posterior* distribution, though the same approach applies to any distribution. The basic idea of the *conventional Monte Carlo* method is to draw a set of $N$ i.i.d. random samples $\{\boldsymbol{\theta}^n\}_{n=1}^N$ from the desired distribution $\pi(\boldsymbol{\theta})$ and obtain an unbiased Monte Carlo estimate of the integral (1.3) as

$$\hat{I}_N = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{\theta}^n). \tag{1.4}$$

By the Strong Law of Large Numbers (SLLN), the estimator $\hat{I}_N$ converges almost surely to $I$ as the number of samples $N$ tends to infinity, i.e. $\mathbb{P}(\hat{I}_N = I) = 1$, as $N \to \infty$. If the variance of $f(\boldsymbol{\theta})$, $\sigma_f^2$, is finite, then the Central Limit Theorem states that the error $\sqrt{N}(\hat{I}_N - I)$ converges in distribution to a $\mathcal{N}(0, \sigma_f^2)$ random variable. Therefore, the variance of the Monte Carlo estimator $\hat{I}_N$ is

$$\mathrm{Var}(\hat{I}_N) = \frac{\sigma_f^2}{N}.$$

In most of the problems, however, it is not possible to draw independent samples directly from the desired distribution in order to find the Monte Carlo estimate (1.4). Many techniques have been developed over the past decades (see e.g. Robert and Casella, 2005). In this thesis, we focus on the methodology which combines several advanced Monte Carlo based techniques in a rigorous way. Among them are importance sampling and Markov chain Monte Carlo methods for which we give more details in the following sections.

### 1.2.2 Importance Sampling

Rather than sampling directly from the desired distribution $\pi$, the importance sampling method (Kahn and Marshall, 1953) generates samples from a different but easy-to-sample-from distribution $\varrho$. The samples then need to be reweighted to account for differences in probabilities. Both the target and importance distribution may be known only up to normalizing constants. Let $Z$ and $Z_\varrho$ be the normalizing constants of $\pi$ and $\varrho$, respectively, such that $\pi = p/Z$ and $\varrho = q/Z_\varrho$. Using a few simple tricks, one can prove that the expected value of a function $f$ with respect to the distribution $\pi$ can be obtained as an expectation with respect to the distribution $\varrho$, as follows

$$\mathbb{E}_\pi[f] \quad = \quad I = \int f(\boldsymbol{\theta})\frac{p(\boldsymbol{\theta})}{Z}\mathrm{d}\boldsymbol{\theta} = \frac{\int f(\boldsymbol{\theta})\frac{p(\boldsymbol{\theta})}{Z}\mathrm{d}\boldsymbol{\theta}}{\int \frac{p(\boldsymbol{\theta})}{Z}\mathrm{d}\boldsymbol{\theta}} = \frac{\frac{Z}{Z_\varrho}\int f(\boldsymbol{\theta})\frac{p(\boldsymbol{\theta})}{Z}\mathrm{d}\boldsymbol{\theta}}{\frac{Z}{Z_\varrho}\int \frac{p(\boldsymbol{\theta})}{Z}\mathrm{d}\boldsymbol{\theta}}$$

$$= \frac{\int f(\boldsymbol{\theta}) \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \frac{q(\boldsymbol{\theta})}{Z_\varrho} \mathrm{d}\boldsymbol{\theta}}{\int \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \frac{q(\boldsymbol{\theta})}{Z_\varrho} \mathrm{d}\boldsymbol{\theta}} = \frac{\mathbb{E}_\varrho[wf]}{\mathbb{E}_\varrho[w]}, \tag{1.5}$$

where

$$w(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}$$

is the *importance weight* function and accounts for differences in the desired probability $\pi$ and probability $\varrho$ used for generating a sample.

The Monte Carlo estimate of $I$ follows as

$$\hat{I}_N = \frac{\sum_{n=1}^N w(\boldsymbol{\theta}^n) f(\boldsymbol{\theta}^n)}{\sum_{n=1}^N w(\boldsymbol{\theta}^n)}, \quad \boldsymbol{\theta}^n \sim \varrho(\boldsymbol{\theta}) \tag{1.6}$$

and SLLN holds, that is $\hat{I}_N \xrightarrow[N\to\infty]{\text{a.s.}} I$.

Efficiency of an importance sampler depends on the choice of the importance function $\varrho(\boldsymbol{\theta})$, which should satisfy the following:

(i) $\varrho(\boldsymbol{\theta})$ is a fairly good approximation of the desired distribution. In particular, $\varrho(\boldsymbol{\theta}) > 0$ whenever $\pi(\boldsymbol{\theta}) > 0$.

(ii) It is easy to simulate samples from $\varrho(\boldsymbol{\theta})$.

Problems with importance samplers may occur e.g. when a small number of weights are much larger than others, which means that there are effectively only few samples generated. Also, high variability in weights might increase the variance of $\hat{I}_N$, therefore resulting in an inefficient Monte Carlo estimator of $I$.

Importance samplers especially suffer from severe limitations in high dimensional spaces. For such problems it is difficult, if not impossible, to find a distribution $\varrho$ that fulfills (i) and (ii) and we consider more sophisticated methods that introduce correlation among samples through Markov chains.

### 1.2.3 Markov chain Monte Carlo

A widely used alternative to generating independent samples from the desired distribution is to draw random samples by evolving a Markov chain on parameter space. This approach was developed starting from a statistical mechanics perspective, and it was introduced by Metropolis et al. (1953). The resulting Markov chain Monte Carlo (MCMC) method is a commonly used sampling technique for Bayesian computation. Actually, Bayesian uprising took place when Geman and Geman (1984) and Gelfand and Smith (1990) adopted MCMC and also due to increased computing power. Thanks to this powerful methodology, we can now assess the uncertainties in a Bayesian analysis through a numerically calculated posterior distribution. MCMC provides a means to simulate from a complex distribution $\pi$ in high-dimensional problems, without knowing its normalizing constant.

The idea behind MCMC is to construct a Markov chain whose invariant distribution is the target distribution $\pi$ and simulate the chain for many steps. In the long run, the states of the chain follow the target distribution.

In a Markov chain, the probability of introducing a new state in the chain depends on the current state only and not on the past. In particular, a new state of the chain $\boldsymbol{\theta}^{n+1}$ is generated from a transition density that only depends on the current state $\boldsymbol{\theta}^n$

$$\boldsymbol{\theta}^{n+1} \sim K(\boldsymbol{\theta}^{n+1}|\boldsymbol{\theta}^n).$$

A distribution $\pi$ is called the *invariant (stationary)* distribution for the transition kernel $K$ if for all points $\boldsymbol{\theta}'$

$$\pi(\boldsymbol{\theta}') = \int K(\boldsymbol{\theta}'|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{1.7}$$

In other words, the transition kernel $K$ *preserves* the invariant distribution $\pi$.

The properly designed MCMC method samples the target distribution rigorously, provided that the Markov chain possesses some properties and that the number of states in the chain approaches infinity.

If a Markov chain is *irreducible*, meaning that all states can be reached with positive probability in a finite number of steps, and *aperiodic*, meaning that return to a state can occur at irregular time, then the chain has a unique invariant distribution $\pi$ and it will converge to $\pi$ for $n \to \infty$, independently of the initial distribution. Moreover, as a consequence of the SLLN, the chain is *ergodic* and time averages on a single realization converge to ensemble averages for $n \to \infty$.

The states from the beginning of the chain are highly dependent on the initial state, due to the Markovian nature of the MCMC algorithm. Therefore, those samples are usually removed as *warm-up*.

A very important condition on a Markov chain is the *detailed balance condition*

$$\pi(\boldsymbol{\theta})K(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}')K(\boldsymbol{\theta}|\boldsymbol{\theta}'),\ \forall \boldsymbol{\theta}, \boldsymbol{\theta}'. \tag{1.8}$$

If the detailed balance (DB) condition holds, the chain is called *reversible*, as the probabilities of the chain going from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}'$ and from $\boldsymbol{\theta}'$ to $\boldsymbol{\theta}$ are equal for all $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. DB is important because it implies the invariant distribution condition (1.7), which follows straightforwardly by integrating both sides of the equality of (1.8). It is a sufficient but not necessary condition for $\pi$ to be the desired invariant distribution, i.e. a chain can have an invariant distribution although not satisfying detailed balance. A way to design an MCMC algorithm with invariant distribution $\pi$ is to ensure that the DB condition is satisfied, which is easier than proving the invariance condition.

Another role of the DB condition in MCMC lays in the fact that if the chain is reversible, irreducible and all states are drawn from the invariant distribution, then the Central Limit Theorem holds for long time averages and thus the variance of the MC estimator can be estimated (Geyer, 1992).

### Metropolis-Hastings algorithm

Metropolis-Hastings was the first MCMC algorithm, initially designed by Metropolis et al. (1953) for simulation of a liquid in equilibrium with its gas phase and later extended to the more general case by Hastings (1970). The method was formulated in the year 1953, from the statistical mechanics perspective, but the mainstream community of statisticians adopted it significantly later. It was Geman and Geman (1984) who made use of it in Bayesian inference problems and sampling from posterior distributions applied to computer vision. In the year 2000, Metropolis-Hastings was ranked as one of the "10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century" (Dongarra and Sullivan, 2000).

The method is designed in a way to generate states that make a large contribution to the integral of interest. At each iteration, a parameter vector is sampled from a proposal distribution, which depends on the current state, and either accepted or rejected according to the probability of the new sample relative to the current one. The algorithm is summarized below.

**Algorithm 1** Metropolis-Hastings

**Input:** $N$: number of Monte Carlo samples
$q(\boldsymbol{\theta}'|\boldsymbol{\theta})$: proposal distribution
Initialize $\boldsymbol{\theta}^0$
**for** $n = 1, \ldots, N$ **do**
  $\boldsymbol{\theta} = \boldsymbol{\theta}^{n-1}$
  Generate a candidate state from the proposal distribution

$$\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}'|\boldsymbol{\theta})$$

  Calculate acceptance probability

$$\alpha = \min\left(1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right)$$

  Metropolis test
    Draw $u \sim \mathcal{U}(0, 1)$
    **if** $u < \alpha$
      $\boldsymbol{\theta}^n = \boldsymbol{\theta}'$ {accept}
    **else**
      $\boldsymbol{\theta}^n = \boldsymbol{\theta}$ {reject}
    **end if**
**end for**

The acceptance probability $\alpha$ can also be seen as a ratio of importance weights, of the target and proposal distribution of the current and proposed state. As mentioned above, MCMC overcomes the need of knowing the normalizing constant of the target distribution,

since it cancels out in the acceptance probability. MCMC draws samples from the target $\pi$ by calculating $\pi Z$, where $Z$ is the (unknown) normalizing constant.

In the Metropolis algorithm (Metropolis et al., 1953), the proposal distribution $q$ is symmetric, which simplifies the acceptance criteria. Metropolis method is also referred to as a *Random Walk Metropolis-Hastings* method, and it is widely used because of its simplicity.

Nevertheless, some problems deteriorate the efficiency of this algorithm. For example, due to the random walk nature, the Metropolis algorithm is slow in sampling from target distributions, i.e. the chain can be stuck in a local maximum of a distribution with multiple modes, and the convergence rate to the target distribution can be slow. It may require too many iterations to explore the state space and the samples generated are usually highly correlated. The method becomes even more inefficient when applied to high-dimensional problems and with strong correlations among parameters.

In general, the following issues affect the efficiency of an MCMC sampler: (i) *convergence* to the target distribution; (ii) *mixing* properties, i.e. level of correlation among samples; (iii) *computational cost* and (iv) *tuning* mechanisms.

A very active research is being conducted on the development of efficient MCMC methods for sampling from distributions arising from high dimensional problems and complex statistical models. Many different MCMC approaches to improve (i)-(iv) have been proposed in the literature in the last decades. Excellent reviews can be found e.g. in (Andrieu et al., 2003; Robert and Casella, 2005; Liu, 2008; Brooks et al., 2011). One way to overcome problems encountered with the Metropolis-Hastings algorithm is to sample from an augmented distribution $\pi(\boldsymbol{\theta}, \phi)$, where $\phi$ is an auxiliary variable, using so-called *auxiliary variable samplers*. We now shift our focus towards a very successful and popular method from this class, namely the Hamiltonian Monte Carlo methodology, which explores gradient information of the target distribution.

### 1.2.4 Hamiltonian Monte Carlo

As is the case with MCMC, the *Hamiltonian Monte Carlo* (HMC) method originates from statistical physics, where it is known as *Hybrid Monte Carlo*. It was initially developed by Duane, Kennedy, Pendleton, and Roweth (1987), but it was Neal (1994) who launched the application of HMC for statistical problems.

HMC is an MCMC algorithm that produces a chain whose invariant distribution is an augmented target distribution

$$\pi(\boldsymbol{\theta}, \mathbf{p}) = \pi(\boldsymbol{\theta})p(\mathbf{p}) \propto \exp(-H(\boldsymbol{\theta}, \mathbf{p})),$$

with Hamiltonian function

$$H(\boldsymbol{\theta}, \mathbf{p}) = U(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p},$$

where

$$U(\boldsymbol{\theta}) = -\log \pi(\boldsymbol{\theta}) + \text{const}$$

is the potential function and $\mathbf{p}$ is an auxiliary "momenta" variable, independent on parameters $\boldsymbol{\theta}$ and drawn at every Monte Carlo iteration from a zero-mean Gaussian distribution with a covariance matrix $M$. The resulting chain incorporates the gradient information of the target $\pi(\boldsymbol{\theta})$ and can follow this gradient over considerable distances. This is achieved by means of generating Hamiltonian trajectories (integrating Hamiltonian equations of motion), which are rejected or accepted based on the Metropolis test. The (marginal) distribution of parameters $\boldsymbol{\theta}$, $\pi(\boldsymbol{\theta})$, is then obtained from $\pi(\boldsymbol{\theta}, \mathbf{p})$ by simply marginalizing out momenta variables.

The potential efficacy of this approach lays in the fact that the value of the Hamiltonian function, and so the value of $\pi(\boldsymbol{\theta}, \mathbf{p})$, does change along a Hamiltonian trajectory only due to the inaccuracies in numerical integration of Hamiltonian equations, which leads to a likely acceptance of the proposed state with a possibly quite different value of $\pi(\boldsymbol{\theta})$. As a result, HMC suppresses the random walk behavior from traditional Monte Carlo techniques and samples high dimensional and complex distributions more efficiently than conventional MCMC.

We provide a thorough overview of the HMC methodology in Chapter 2.

## 1.3 Summary

Both academia and industry have been witnessing exponential accumulation of data, which, if carefully analyzed, potentially can provide valuable insights about the underlying processes. The main feature of the resulting problems is uncertainty, which appears in many forms, either in data or assumed data models, and causes great challenges. Therefore, we need more complex models and advanced analysis tools to deal with the size and complexity of data.

The Bayesian approach offers a rigorous and consistent manner of dealing with uncertainty and provides a means of quantifying the uncertainty in our predictions. It also allows us to objectively discriminate between competing model hypotheses, through the evaluation of Bayes factors. However, the application of the Bayesian framework to complex problems runs into a computational bottleneck that needs to be addressed with efficient inference methods. The challenges arise in the e.g. evaluation of intractable quantities or large-scale inverse problems linked to computationally demanding forward problems, and especially in high dimensional settings. Theoretical and algorithmic developments in computational statistics have been leading to the possibility of undertaking more complex applications by scientists and practitioners, but sampling in high dimensional problems and complex distributions is still challenging.

Various methods are being used to address these difficulties practically. Deterministic methods for example, aim to find analytically tractable approximations of distributions of interest. We are interested however in Monte Carlo (stochastic) methods, which can sample from the desired distribution, are exact in the limit of infinite number of samples, and can achieve an arbitrary level of accuracy by drawing as many samples as one requires.

Bayesian statistics have been revolutionized by ever-growing computational capacities and development of Markov chain Monte Carlo (MCMC) techniques. In this thesis, we focus on an advanced MCMC methodology, namely Hamiltonian Monte Carlo (HMC), which arose as a good candidate for dealing with high-dimensional and complex problems. Interestingly, both MCMC and HMC were originally developed in statistical mechanics; however, they made a significant impact on the statistical community almost 35 and 10 years later, respectively.

Our objective in this thesis is to improve the HMC method further by developing more efficient methodologies for rigorous enhanced sampling in complex problems and to analyze their performance.

We start with providing details and an outlook on the HMC methodology in Chapter 2. In Chapter 3 we present the novel Mix & Match Hamiltonian Monte Carlo (MMHMC) method and a number of different strategies that can be employed within. Several extensions to MMHMC are designed in Chapter 4. In Chapter 5 we describe the software package developed along this thesis in which the novel MMHMC has been implemented. Testing and comparison of MMHMC with popular sampling techniques is provided in Chapter 6. Finally, Chapter 7 concludes this thesis and outlines some directions for further research.

# 2

# Hamiltonian Monte Carlo Methods

## 2.1 Background essentials

We begin with reviewing the basic concepts and main ingredients of the Hamiltonian Monte Carlo method. These include Hamiltonian dynamics, numerical integration of Hamilton's equations of motion and the framework of modified Hamiltonians. Further details can be found in e.g. (Sanz-Serna and Calvo, 1994; Hairer et al., 2006; Leimkuhler and Reich, 2005).

### 2.1.1 Hamiltonian dynamics

We consider a system of $D$ particles in which the state $(\mathbf{x}, \mathbf{p}) \in \mathbf{\Omega}$ at time $t \in T \subseteq \mathbb{R}_+$ is determined by a *position* vector $\mathbf{x} = (x_1, \ldots, x_D)$ and a *momentum* vector $\mathbf{p} = (p_1, \ldots, p_D)$, where $\mathbf{\Omega} \subseteq \mathbb{R}^{2D}$ is called the *phase space*. The system is characterized by a real valued Hamiltonian function $H = H(\mathbf{x}, \mathbf{p}, t)$, which is interpreted as the total energy of the system. The associated Hamiltonian dynamics is governed by the system of ordinary differential equations

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} &= H_{\mathbf{p}}(\mathbf{x}, \mathbf{p}, t) \\ \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} &= -H_{\mathbf{x}}(\mathbf{x}, \mathbf{p}, t), \end{aligned} \tag{2.1}$$

where $H_{\mathbf{p}}$ and $H_{\mathbf{x}}$ are partial derivatives of the Hamiltonian with respect to momentum and position, respectively. We focus on a class of *separable* Hamiltonians, defined in $\mathbf{\Omega}$ as

$$H(\mathbf{x}, \mathbf{p}) = U(\mathbf{x}) + K(\mathbf{p}), \tag{2.2}$$

where $U(\mathbf{x})$ is interpreted as the potential energy and $K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}$ is interpreted as the kinetic energy with $M$ being a symmetric positive definite matrix (mass matrix of position variables). This is known as an *autonomous system*, meaning that the Hamiltonian remains constant over time. Equations (2.1) then read as

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} &= M^{-1}\mathbf{p} \\ \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} &= -U_{\mathbf{x}}(\mathbf{x}). \end{aligned} \tag{2.3}$$

Throughout this chapter the notation $\mathbf{z} = (\mathbf{x}, \mathbf{p})$ will also be used. The system (2.1) can then be rewritten as

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbf{J}H_{\mathbf{z}}(\mathbf{z}), \quad \mathbf{J} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \tag{2.4}$$

where $I$ is the identity matrix of dimension $D$. An alternative formulation,

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \{\mathbf{Id}, H\}(\mathbf{z}) \equiv \mathcal{L}_H(\mathbf{z}), \tag{2.5}$$

where $\mathbf{Id}$ denotes the identity map, uses the definition of the Poisson bracket and Lie derivative, which will be used later in this thesis.

The *Poisson bracket* of operators $F, G : \mathbb{R}^{2D} \to \mathbb{R}$ is defined as

$$\{F, G\}(\mathbf{z}) = F_{\mathbf{z}}(\mathbf{z})^T \mathbf{J} G_{\mathbf{z}}(\mathbf{z}). \tag{2.6}$$

For functions $F, G, H : \mathbb{R}^{2D} \to \mathbb{R}$, and scalars $\alpha, \beta$ it holds

$$\{F, \alpha G + \beta H\} = \alpha\{F, G\} + \beta\{F, H\}$$

and

$$\{F, G\} = -\{G, F\}.$$

The last identity implies that $\{F, F\} = 0$.

The *Lie derivative* is defined in terms of Poisson bracket as

$$\mathcal{L}_F G = \{G, F\}.$$

If the structure matrix $\mathbf{J}$ is defined as in (2.4), the Hamiltonian system is called *canonical*, whereas different generalizations of $\mathbf{J}$ correspond to *non-canonical* systems.

The solution flow associated to this system $\Phi_t : \mathbf{\Omega} \to \mathbf{\Omega}$, defined as

$$\mathbf{z}(t) = \Phi_t(\mathbf{z}(0)),$$

has some key properties that form the basis of HMC as a valid MCMC method.

**Conservation of the Hamiltonian** For systems given by the equation (2.4), Hamiltonian is a constant quantity along the solutions, i.e.

$$H \circ \Phi_t = H.$$

Change in the potential energy is balanced by the change in kinetic energy.

*Relevance to HMC*: If proposals in the Markov chain are obtained using the solution flow $\Phi_t$, the probability of all states is equal, and thus, the acceptance probability in the Metropolis test is equal to one. In practice, however, due to numerical discretization, this is not the case as the Hamiltonian is only approximately conserved.

**Conservation of volume** A volume element $\mathrm{d}\mathbf{z}$ is preserved under $\Phi_t$, i.e.

$$\mathrm{d}\mathbf{z} = \mathrm{d}\Phi_t(\mathbf{z}).$$

Alternatively, if $\Phi_t' \in \mathbf{\Omega} \times \mathbf{\Omega}$ is the Jacobian of the flow $\Phi_t$ then

$$\det(\Phi_t') = 1 \text{ for each } t.$$

This property is known as Liouville's Theorem (Arnold, 1989) and is equivalent to the divergence of the vector field defined by the system (2.4) being equal to zero. In the case $D = 1$, conservation of volume corresponds to conservation of area in the $(x, p)$ plane.

*Relevance to HMC*: In general, one should account for the change in volume introduced by the mapping for proposing new states in an MCMC method and this is done through the calculation of $\det(\Phi_t')$. A solid formulation and analysis of the technique are given by Fang et al. (2014). If Hamiltonian dynamics is used for proposals, there is no need to consider volume change.

**Symplecticness** Hamiltonian flow map $\Phi_t$ is a *symplectic* map, that is, satisfies the condition

$$\Phi_t'(\mathbf{z})^T \mathbf{J}^{-1} \Phi_t'(\mathbf{z}) = \mathbf{J}^{-1}, \text{ for every } \mathbf{z} \in \mathbf{\Omega}.$$

Symplecticness implies certain conservation laws, in particular, the conservation of volume.

*Relevance to HMC*: Numerical methods that preserve the symplectic structure are good candidates for HMC because of their good numerical stability properties and implied preservation of volume.

**Reversibility** For a map $\mathcal{F}(\mathbf{x}, \mathbf{p}) = (\mathbf{x}, -\mathbf{p})$ in phase space, which flips the sign of momenta, the flow $\Phi_t$ is reversible. This can be written as

$$\Phi_t \circ \mathcal{F} = (\Phi_t \circ \mathcal{F})^{-1}$$

or

$$\Phi_{-t} = \mathcal{F} \circ \Phi_t \circ \mathcal{F},$$

since map $\mathcal{F}$ is an involution, i.e. $\mathcal{F} \circ \mathcal{F} = \mathbf{Id}$. This means that the backward evolution is equivalent to flipping the initial momenta, evolving in time, and flipping the final momenta.

Also, the Hamiltonian is an even function of momenta, i.e.

$$H \circ \mathcal{F} = H.$$

*Relevance to HMC*: Reversibility of the flow that proposes states (i.e. Markov transitions) in an MCMC method is essential for proving the detailed balance condition, which ensures that the Markov chain leaves the target distribution invariant.

### 2.1.2 Numerical integration

In practice, the analytical expression of the flow is rarely available. Hence, the continuous Hamiltonian dynamics is approximated using a numerical scheme with a small time step $h$. This scheme gives rise to a map $\Psi_h$ that approximates the flow $\Phi_t$. We focus here on *one-step* numerical methods, which iteratively evolve the approximate solution $\mathbf{z}^n \approx \mathbf{z}(nh)$ using only the previously computed solution, i.e. $\mathbf{z}^{n+1} = \Psi_h(\mathbf{z}^n)$. The approximate solution at time $\tau = Lh$ is obtained by applying $L$ times the map $\Psi_h$

$$\Phi_\tau(\mathbf{z}) \approx \Psi_\tau(\mathbf{z}) = \Psi_{h,L}(\mathbf{z}) = \underbrace{\Psi_h \circ \cdots \circ \Psi_h}_{L \text{ times}}(\mathbf{z}).$$

These numerical schemes are also $\boldsymbol{\Omega} \to \boldsymbol{\Omega}$ mappings, and one can apply the same analysis as for flow maps, which leads to the characterization of some properties that are desirable for HMC methods.

Commonly, the integrators of choice are reversible and symplectic, i.e. preserve the symplectic structure of the Hamiltonian dynamics. Analogously to Hamiltonian flows, the numerical integrator $\Psi_h$ is symplectic if

$$\left[\frac{\partial}{\partial \mathbf{z}} \Psi_h(\mathbf{z})\right]^T \mathbf{J}^{-1} \left[\frac{\partial}{\partial \mathbf{z}} \Psi_h(\mathbf{z})\right] = \mathbf{J}^{-1}.$$

Symplecticness of the map $\Psi_h$ implies preservation of volume, but $\Psi_h$ does not exactly conserve energy due to discretization and thus, introduces an integration error. The difference between the approximated and true solution after one step of integration is called the *local error* and has an order of $\mathcal{O}(h^{p+1})$ for a $p$-order numerical method,

$$\Psi_h(\mathbf{z}) = \Phi_h(\mathbf{z}) + \mathcal{O}(h^{p+1}).$$

The *global error* is an error accumulated after integrating over a fixed time interval $\tau$ using $L$ steps. It follows

$$\Psi_\tau = \Big(\Phi_h + \mathcal{O}(h^{p+1})\Big)^L = \Phi_{Lh} + (Lh)\mathcal{O}(h^p) = \Phi_\tau + \mathcal{O}(h^p).$$

Hence, for a method of order $p$ the global integration error is $\mathcal{O}(h^p)$. A numerical integrator is said to be *stable* if all trajectories remain bounded as time goes to infinity. This means that using unstable numerical integrators within the HMC method might result in substantial differences in Hamiltonians after integration and low acceptance rates.

From now on, we focus on second-order integrators. The most popular of them is the *Verlet* method, also known as *Verlet/Störmer* or *leapfrog* method (Verlet, 1967). The Verlet integrator is a numerical approximation to the dynamics (2.3) of separable Hamiltonian systems, defined by the fully explicit three-steps procedure

$$\begin{aligned} \mathbf{p}_{\frac{h}{2}} &= \mathbf{p}_0 - \frac{h}{2} U_{\mathbf{x}}(\mathbf{x}_0) \\ \mathbf{x}_h &= \mathbf{x}_0 + hM^{-1}\mathbf{p}_{\frac{h}{2}} \\ \mathbf{p}_h &= \mathbf{p}_{\frac{h}{2}} - \frac{h}{2} U_{\mathbf{x}}(\mathbf{x}_h). \end{aligned} \tag{2.7}$$

More sophisticated symplectic methods exist, but the Verlet method is commonly used in both molecular and statistical simulations, due to its stability and preservation properties and easy implementation.

In general, a symplectic scheme can be constructed using the simple but useful technique based on *splitting* Hamiltonian $H$ as

$$H = H^1 + H^2 + \cdots + H^k.$$

The corresponding Hamiltonian vector fields

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbf{J}H^i_{\mathbf{z}}(\mathbf{z}),\ i = 1, \ldots, k$$

have exact solution flows $\varphi_t^{H^i},\ i = 1, \ldots, k$, that can be calculated explicitly. From (2.5) it can be seen that each flow is an exponential operator defined on a Lie derivative

$$\mathbf{z}(t) = \varphi_t^{H^i}(\mathbf{z}(0)) = \mathrm{e}^{t\mathcal{L}_{H^i}}\mathbf{z}(0),\ i = 1, \ldots, k.$$

The *composition* of the flows $\varphi_t^{H^i}$ then can be used for the construction of different numerical methods for integration of (2.4). As $\varphi_t^{H^i}$ are exact Hamiltonian flows and thus, symplectic, their composition is symplectic. Each flow is also reversible, and if a symmetric composition is used, the integrator is time-reversible. Moreover, it can be proved that a symmetric method is of even order.

Let us come back to the Verlet integrator (2.7) and Hamiltonian (2.2), which is defined as a composition of kinetic and potential term, i.e.

$$H(\mathbf{x}, \mathbf{p}) = K(\mathbf{p}) + U(\mathbf{x}) \equiv A + B. \tag{2.8}$$

The exact flows are

$$\varphi_h^A(\mathbf{x}, \mathbf{p}) = \begin{bmatrix} \mathbf{x} + hM^{-1}\mathbf{p} \\ \mathbf{p} \end{bmatrix}$$

and

$$\varphi_h^B(\mathbf{x}, \mathbf{p}) = \begin{bmatrix} \mathbf{x} \\ \mathbf{p} - hU_{\mathbf{x}}(\mathbf{x}) \end{bmatrix}.$$

The Verlet integrator (2.7) can now be regarded as a symmetric composition

$$\Psi_h = \varphi_{h/2}^B \circ \varphi_h^A \circ \varphi_{h/2}^B$$

associated to *Strang's* splitting

$$H^1 = \frac{1}{2}B,\ H^2 = A,\ H^3 = \frac{1}{2}B.$$

Analogously, we can construct the position variant of the Verlet integrator

$$\Psi_h = \varphi_{h/2}^A \circ \varphi_h^B \circ \varphi_{h/2}^A.$$

The Verlet integrator is time reversible due to the symmetry in the composition and reversibility of the flows $\varphi_h^A$ and $\varphi_h^B$.

If the dynamics is integrated for $L$ steps using the Verlet integrator $\Psi_{h,L}$, the conservation of Hamiltonian is violated with error

$$H(\mathbf{z}^n) = H(\mathbf{z}^0) + \mathcal{O}(h^2),$$

but the modified Hamiltonian $\tilde{H}$ is conserved as discussed below..

### 2.1.3 Modified Hamiltonians

For a discrete solution to $\mathrm{d}\mathbf{z}/\mathrm{d}t = f(\mathbf{z})$, given by a $p$-order numerical method $\Psi_h$ there exist the modified equations

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \tilde{f}(\mathbf{z}) \tag{2.9}$$

that are exactly satisfied, i.e. $\Psi_h(\mathbf{z}) = \Phi_{h,\tilde{f}}(\mathbf{z})$. Modified equations are defined by an asymptotic expansion in powers of the discretization parameter as

$$\tilde{f} = f + h^{p+1} f_{p+1} + h^{p+2} f_{p+2} + \ldots, \tag{2.10}$$

where vector fields $f_j$ can be determined by expanding the exact flow $\Phi_{h,\tilde{f}}$ and the numerical integrator $\Psi_h$ as Taylor series in terms of $h$ and matching corresponding terms in the two expansions. The asymptotic expansion (2.10) does not converge in general, except for appropriate integrators applied to linear differential equations.

Modified equations (or backward error) analysis leads to a *modified (or shadow) Hamiltonian*

$$\tilde{H} = H + h^p H_{p+1} + h^{p+1} H_{p+2} + \dots$$

that is conserved nearly exactly by a symplectic method $\Psi_h$, i.e.

$$|\tilde{H}(\Psi_h(\mathbf{z})) - \tilde{H}(\mathbf{z})| \leq c_1 h \mathrm{e}^{-\frac{c_2}{h}}, \text{ for some } c_1, c_2 \in \mathbb{R}. \tag{2.11}$$

For a $p$-order symplectic method the difference between the true and modified Hamiltonian is

$$H(\mathbf{z}) - \tilde{H}(\mathbf{z}) = \mathcal{O}(h^p). \tag{2.12}$$

It can be proved that the modified differential equations of the system (2.4) are also Hamiltonian for some modified Hamiltonian $\tilde{H}$ if and only if the integration method is symplectic (Sanz-Serna and Calvo, 1994). On the contrary, non-symplectic integrators, for which the modified equations are not Hamiltonian, will not preserve Hamiltonian properties of the system. It follows that the equations (2.9) can be written in terms of a modified Hamiltonian as

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t} = \mathbf{J}\tilde{H}_{\mathbf{z}}(\mathbf{z}).$$

In practice, one is interested in a $k$-order modified Hamiltonian, defined as a *truncation* of a modified Hamiltonian $\tilde{H}$ up to $h^{k-1}$ terms

$$\tilde{H}^{[k]} = H + h^p H_{p+1} + \dots + h^{k-1} H_k. \tag{2.13}$$

It can be proved that for a reversible integrator, the modified Hamiltonian has an expansion in even powers of $h$. This implies that e.g. for a second-order integrator the truncated modified Hamiltonian is of order $k \geq 4$. Clearly,

$$\tilde{H}^{[k]}(\mathbf{z}) = \tilde{H}(\mathbf{z}) + \mathcal{O}(h^k)$$

and hence, a symplectic method $\Psi_h$ preserves the truncated Hamiltonian up to order $h^k$, i.e.

$$\tilde{H}^{[k]}(\Psi_h(\mathbf{z})) = \tilde{H}^{[k]}(\mathbf{z}) + \mathcal{O}(h^k).$$

Precisely this property of modified Hamiltonians, i.e. better conservation with symplectic integrators versus true Hamiltonian, will be used in HMC methodologies for improving acceptance rates.

The construction of a modified Hamiltonian is always defined by choice of the integrator. In case of the Verlet integrator, we first write it down using exponential notation as

$$\Psi_h(\mathbf{z}) = \varphi^B_{h/2} \circ \varphi^A_h \circ \varphi^B_{h/2}(\mathbf{z}) = \mathrm{e}^{(h/2)\mathcal{L}_B} \mathrm{e}^{h\mathcal{L}_A} \mathrm{e}^{(h/2)\mathcal{L}_B} \mathbf{z} = \mathrm{e}^{h\mathcal{L}_{\tilde{H}}} \mathbf{z}.$$

The corresponding modified Hamiltonian can now be determined by multiple application of the Baker-Campbell-Hausdorff (BCH) formula (Sanz-Serna and Calvo, 1994) on Lie derivatives, obtaining

$$\tilde{H} = H + \frac{h^2}{24}\big(2\{A,\{A,B\}\} - \{B,\{B,A\}\}\big) + \dots$$

Modified Hamiltonians for more general composition methods might be obtained in the same fashion, or alternatively, following the approach of Murua and Sanz-Serna (1999).

## 2.2 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) method is a popular MCMC technique for sampling of high dimensional and complex systems. It combines deterministic with stochastic approaches, i.e. Hamiltonian dynamics with Metropolis Monte Carlo sampling.

The strength of HMC as a sampling technique comes from its efficient use of gradient information to reduce random walk behavior of conventional Metropolis Monte Carlo. The gradient provides information about the local behavior of the target distribution. This allows for larger moves across the state space and thus faster convergence to the target distribution. Proposals can be distant from current states but still with high acceptance probabilities.

### 2.2.1 History

Despite the complementary nature, Hamiltonian dynamics and Metropolis Monte Carlo had never been considered jointly until the *Hybrid Monte Carlo* method was formulated in the seminal paper by Duane et al. (1987). It was initially applied to lattice field theory simulations and remained unknown for statistical applications till 1994, when Neal used the method in neural network models (Neal, 1994). Since then, the common name in statistical applications is *Hamiltonian Monte Carlo* (HMC). A practitioners-friendly guide to HMC can be found in (Neal, 2011), while comprehensive geometrical foundations are provided in (Betancourt et al., 2016). Conditions under which HMC is geometrically ergodic have been established recently (Livingstone et al., 2016).

Nowadays, HMC is used in a wide range of applications – from molecular simulations to statistical problems appearing in many fields, such as ecology, cosmology, social sciences, biology, pharmacometrics, biomedicine, engineering, business. Software package Stan (Stan Development Team, 2016) has contributed to the increased popularity of the method by implementing HMC based sampling within a probabilistic modeling language in which statisticians can write their models in a familiar notation.

### 2.2.2 Formulation

Let us focus on the statistical perspective of the HMC method, namely the Hamiltonian Monte Carlo method. We are interested in sampling a random variable $\boldsymbol{\theta} \in \mathbb{R}^D$ with density $\pi(\boldsymbol{\theta})$. The target probability density function (p.d.f.) can be written as

$$\pi(\boldsymbol{\theta}) = \frac{1}{Z} \exp(-U(\boldsymbol{\theta})), \tag{2.14}$$

where the variable $\boldsymbol{\theta}$ corresponds to the position vector, $U(\boldsymbol{\theta})$ to the potential function of a Hamiltonian system and $Z$ is the normalizing constant such that $\pi(\boldsymbol{\theta})$ integrates to one. In the Bayesian framework, the target distribution $\pi(\boldsymbol{\theta})$ can be seen as the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$ of model parameters given data $\mathbf{y} = \{y_1, \ldots, y_K\}$, $K$ being the size of the data, and the potential function can be defined as

$$U(\boldsymbol{\theta}) = -\log L(\boldsymbol{\theta}|\mathbf{y}) - \log p(\boldsymbol{\theta}), \tag{2.15}$$

for the likelihood function $L(\boldsymbol{\theta}|\mathbf{y})$ and prior p.d.f. $p(\boldsymbol{\theta})$ of model parameters.

The auxiliary momentum variable $\mathbf{p} \in \mathbb{R}^D$, conjugate to and independent on the vector $\boldsymbol{\theta}$ is typically drawn from a normal distribution

$$\mathbf{p} \sim \mathcal{N}(0, M), \tag{2.16}$$

with a covariance matrix $M$, which is positive definite and often diagonal. The Hamiltonian function can be defined in terms of the target p.d.f. as the sum of the potential function and the kinetic function

$$H(\boldsymbol{\theta}, \mathbf{p}) = U(\boldsymbol{\theta}) + K(\mathbf{p}) = U(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + \frac{1}{2}\log((2\pi)^D |M|). \tag{2.17}$$

The joint p.d.f. is then

$$\begin{aligned} \pi(\boldsymbol{\theta}, \mathbf{p}) = \pi(\boldsymbol{\theta})p(\mathbf{p}) = \frac{1}{Z}\exp(-H(\boldsymbol{\theta}, \mathbf{p})) = \frac{(2\pi)^{\frac{D}{2}}|M|}{Z}\exp(-U(\boldsymbol{\theta}))\exp(-\frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}) \\ \propto \exp(-U(\boldsymbol{\theta}))\exp(-\frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}). \end{aligned} \tag{2.18}$$

By simulating a Markov chain with the invariant distribution (2.18) and marginalizing out momentum variables, one recovers the target distribution $\pi(\boldsymbol{\theta})$.

HMC samples from $\pi(\boldsymbol{\theta}, \mathbf{p})$ by alternating a step for a momentum update and a step for a joint, position and momentum, update, for each Monte Carlo iteration. In the first step, momentum is replaced by a new draw from the normal distribution (2.16). In the second step, a proposal for the new state, $(\boldsymbol{\theta}', \mathbf{p}')$, is generated by integrating Hamiltonian dynamics for $L$ steps using a symplectic integrator $\Psi_h$ with a step size $h$. Due to the numerical approximation of integration, Hamiltonian function and thus, the density (2.18), are not preserved. In order to restore this property, which ensures invariance of the target density, an accept-reject step is added through a Metropolis criteria. The acceptance probability has

a simple form

$$\alpha = \min\left\{1, \exp(-(H(\boldsymbol{\theta}', \mathbf{p}') - H(\boldsymbol{\theta}, \mathbf{p})))\right\},$$

which, due to the preservation of volume, does not include potentially difficult to compute Jacobians of the mapping. As in any MCMC method, in case of rejection, the current state is counted again in the estimation of integral (1.3). Once next sample is obtained, momentum is replaced by a new draw, so Hamiltonians have different values for consecutive samples. This means that samples are drawn along different level sets of Hamiltonians, which actually makes HMC an efficient sampler.

For a constant matrix $M$ the last term in the Hamiltonian (2.17) is a constant that cancels out in the Metropolis test. Therefore, the Hamiltonian can be defined as

$$H(\boldsymbol{\theta}, \mathbf{p}) = U(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}.$$

The algorithmic summary of the HMC method is given below.

**Algorithm 2** Hamiltonian Monte Carlo

**Input:** $N$: number of Monte Carlo samples
$h$: time step
$L$: number of integration steps
$M$: mass matrix
$\Psi_{h,L}$: numerical integrator
Initialize $\boldsymbol{\theta}^0$
**for** $n = 1, \ldots, N$ **do**
  $\boldsymbol{\theta} = \boldsymbol{\theta}^{n-1}$
  Draw momentum from Gaussian distribution

$$\mathbf{p} \sim \mathcal{N}(0, M)$$

  Generate a proposal by integrating Hamiltonian dynamics

$$(\boldsymbol{\theta}', \mathbf{p}') = \Psi_{h,L}(\boldsymbol{\theta}, \mathbf{p})$$

  Calculate the acceptance probability

$$\alpha = \min\{1, \exp(-\Delta H)\}, \quad \Delta H = H(\boldsymbol{\theta}', \mathbf{p}') - H(\boldsymbol{\theta}, \mathbf{p})$$

  Metropolis test
    Draw $u \sim \mathcal{U}(0, 1)$
    **if** $u < \alpha$
      $\boldsymbol{\theta}^n = \boldsymbol{\theta}'$ {accept}
    **else**
      $\boldsymbol{\theta}^n = \boldsymbol{\theta}$ {reject}
    **end if**
  Discard momentum $\mathbf{p}'$
**end for**

It is known that the average error between the Hamiltonians at initial and final state of a trajectory generated by a $p$th order numerical integrator satisfies

$$\mathbb{E}(\Delta H) = \mathcal{O}(Dh^{2p}), \tag{2.19}$$

where $D$ is the number of degrees of freedom (Kennedy and Pendleton, 2001). The average acceptance probability is

$$P_{acc} = \operatorname{ercf}\left(\frac{1}{2}\sqrt{\mathbb{E}(\Delta H)}\right),$$

as derived first by Gupta et al. (1990). This means that in order to maintain a reasonable acceptance rate for increasing dimension of the system, the step size should be proportional to $D^{-1/4}$.

Some work has been done for finding the optimal average acceptance rate with respect to computational cost. Beskos et al. (2013) identified the value of 0.651 as an optimal acceptance rate for distributions with independent and identically distributed variates and the Verlet integrator. This result was extended for general distributions and symplectic integrators by Betancourt et al. (2014) with the optimal interval for average acceptance rate being between 0.6 and 0.9.

### 2.2.3 Numerical integrators

As already mentioned, the computer implementation of HMC requires a numerical scheme to approximate the Hamiltonian flow (2.3). These schemes do not conserve the Hamiltonian, hence, do not exactly preserve the probability measure $\pi$, but can be used as a proposing mechanism. The invariance of $\pi$ is ensured by the Metropolis test, which uses the error in the Hamiltonian introduced by the numerical approximation. If the exact flow could be used for integration, the Metropolis probability would be

$$\exp(H(\boldsymbol{\theta}, \mathbf{p}) - H(\boldsymbol{\theta}', \mathbf{p}')) = 1.$$

Therefore, a numerical method that preserves Hamiltonian better than another implies higher acceptance rate of Monte Carlo samples.

Here we list some of the symplectic integrators suggested and used for HMC sampling.

**Verlet / leapfrog integrator (Verlet, 1967)** is the integrator of choice for most of the HMC based methods, due to its robustness and simple implementation. One integration step is defined as (2.7). The standard approach for implementation of the Verlet integrator is to merge the last step of momentum update of one integration step with the first of the next integration step so that half steps for momentum are performed only at the very beginning and very end of a trajectory. Step 6. of the Algorithm 2 becomes

$\mathbf{p} = \mathbf{p} - \frac{h}{2} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})$
**for** $i = 1, \ldots, L-1$ **do**
  $\boldsymbol{\theta} = \boldsymbol{\theta} + h\mathbf{M}^{-1}p$
  $\mathbf{p} = \mathbf{p} - hU_{\boldsymbol{\theta}}(\boldsymbol{\theta})$
**end for**
$\boldsymbol{\theta}' = \boldsymbol{\theta} + hM^{-1}\mathbf{p}$
$\mathbf{p}' = \mathbf{p} - \frac{h}{2} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})$

**Exponential integrators** These integrators are based on splitting the Hamiltonian function as

$$H = \underbrace{\frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + \frac{1}{2}\boldsymbol{\theta}^T \Sigma^{-1}\boldsymbol{\theta}}_{H_1} + \underbrace{\Phi(\boldsymbol{\theta})}_{H_2}$$

or, in other words, decomposing the linear and nonlinear parts of the dynamics

$$U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \Sigma^{-1}\boldsymbol{\theta} + f(\boldsymbol{\theta}). \tag{2.20}$$

The quadratic term of the Hamiltonian, corresponding to the Gaussian component, can then be integrated analytically.

At least three new HMC methods have emerged from this approach. Hilbert space HMC (Beskos et al., 2011) and Split HMC (Shahbaba et al., 2014) use integrator constructed as a composition

$$\varphi_{\frac{h}{2}}^{H_2} \circ \varphi_h^{H_1} \circ \varphi_{\frac{h}{2}}^{H_2}.$$

Hilbert space HMC was formulated in the framework of high-dimensional approximations of distributions on infinite-dimensional Hilbert space defined via density with respect to a Gaussian measure. A generalization of this method, designed as a non-reversible MCMC for a potential improvement of mixing properties was proposed by Ottobre et al. (2016).

In addition to splitting Hamiltonian such that the Gaussian component can be solved exactly, Split HMC method suggests splitting when some parts of the potential and its gradient do not require costly computation, contrary to the slowly varying part. They recommend to use small step size for the fast computation and bigger for costly. Another context is when Hamiltonian can be split by splitting the data. Split HMC method was extended for Gaussian process model by Lan and Shahbaba (2012).

Contrary to previous two methods, Exponential HMC (Chao et al., 2015) treats jointly quadratic and non-quadratic parts of the Hamiltonian, at the cost of introducing the filtering functions $\phi, \psi, \psi_0, \psi_1$. Exact integration within HMC for Gaussian and truncated Gaussian problems is analyzed by Pakman and Paninski (2014).

Below we summarize the reviewed exponential integration schemes using the same notation in order to demonstrate the main differences. Here $\Sigma$ accounts for the linear part of the dynamics (2.20).

| Hilbert HMC | Split HMC | Exponential HMC |
|---|---|---|
| $M = \Sigma^{-1} \equiv \Omega^2$ | $M = I, \Omega^2 \equiv \Sigma^{-1}$ | $M^{-1}, \Omega^2 \equiv M^{-\frac{1}{2}} \Sigma^{-1} M^{-\frac{1}{2}}$ |
| 1: $(\boldsymbol{\Theta}, \mathbf{P}) \leftarrow (\boldsymbol{\theta}, \Sigma \mathbf{p})$ | 1: $(\boldsymbol{\Theta}, \mathbf{P}) \leftarrow (\boldsymbol{\theta}, \mathbf{p})$ | 1: $(\boldsymbol{\Theta}, \mathbf{P}) \leftarrow (M^{\frac{1}{2}} \boldsymbol{\theta}, M^{-\frac{1}{2}} \mathbf{p})$ |
| $(C, S) = (\cos(h), \sin(h))$ | $(C_\Omega, S_\Omega) = (\cos(h\Omega), \sin(h\Omega))$ | $(C_\Omega, S_\Omega) = (\cos(h\Omega), \sin(h\Omega))$ |
| 2: $\mathbf{P} = \mathbf{P} - \frac{h}{2} \Omega^{-2} f(\boldsymbol{\Theta})$ | 2: $\mathbf{P} = \mathbf{P} - \frac{h}{2} f(\boldsymbol{\Theta})$ | $F(\boldsymbol{\Theta}) \leftarrow M^{-\frac{1}{2}} f(M^{-\frac{1}{2}} \boldsymbol{\Theta})$ |
| 3: $\boldsymbol{\Theta} = C\boldsymbol{\Theta} + S\mathbf{P}$ | 3: $\boldsymbol{\Theta} = C_\Omega \boldsymbol{\Theta} + \Omega^{-1} S_\Omega \mathbf{P}$ | 2: $\boldsymbol{\Theta} = C_\Omega \boldsymbol{\Theta} + \Omega^{-1} S_\Omega \mathbf{P} - \frac{h^2}{2} \psi F(\phi \boldsymbol{\Theta})$ |
| 4: $\mathbf{P} = -S\boldsymbol{\Theta} + C\mathbf{P}$ | 4: $\mathbf{P} = -\Omega S_\Omega \boldsymbol{\Theta} + C_\Omega \mathbf{P}$ | 3: $\mathbf{P} = -\Omega S_\Omega \boldsymbol{\Theta} + C_\Omega \mathbf{P} - \frac{h}{2} \psi_0 F(\phi \boldsymbol{\Theta})$ |
| 5: $\mathbf{P} = \mathbf{P} - \frac{h}{2} \Omega^{-2} f(\boldsymbol{\Theta})$ | 5: $\mathbf{P} = \mathbf{P} - \frac{h}{2} f(\boldsymbol{\Theta})$ | 4: $\mathbf{P} = \mathbf{P} - \frac{h}{2} \psi_1 F(\phi \boldsymbol{\Theta})$ |

Due to an exact integration of the linear part of the dynamics, these integrators are more accurate than the Verlet integrator. This implies better conservation of the Hamiltonian and thus, higher acceptance rate. Nevertheless, their performance depends on the problem at hand. In particular, if the nonlinear part dominates the dynamics or if the linear part can not be approximated accurately or in a computationally feasible manner, the performance of these integrators might degrade.

**Multi-stage integrators** One can consider more sophisticated compositions of flows $\varphi_t^A$ and $\varphi_t^B$, such as two-stage

$$\Psi_h = \varphi_{bh}^B \circ \varphi_{\frac{h}{2}}^A \circ \varphi_{(1-2b)h}^B \circ \varphi_{\frac{h}{2}}^A \circ \varphi_{bh}^B, \tag{2.21}$$

three-stage

$$\Psi_h = \varphi_{bh}^B \circ \varphi_{ah}^A \circ \varphi_{(\frac{1}{2}-b)h}^B \circ \varphi_{(1-2a)h}^A \circ \varphi_{(\frac{1}{2}-b)h}^B \circ \varphi_{ah}^A \circ \varphi_{bh}^B \tag{2.22}$$

and four-stage

$$\Psi_h = \varphi_{b_1 h}^B \circ \varphi_{ah}^A \circ \varphi_{b_2 h}^B \circ \varphi_{(\frac{1}{2}-a)h}^A \circ \varphi_{(1-2b_1-2b_2)h}^B \circ \varphi_{(\frac{1}{2}-a)h}^A \circ \varphi_{b_2 h}^B \circ \varphi_{ah}^A \circ \varphi_{b_1 h}^B \tag{2.23}$$

families of integrators, which require two, three of four gradient evaluations per time step, respectively. Analogously, one can construct the position version of these integrators, with the initial flow in the composition being $\varphi_t^A$.

We note that the concatenation of two Verlet steps of size $h_V = h/2$ are equivalent to one step of the two-stage integrator with coefficient $b = 1/4$

$$\Psi_{h_V}^V \circ \Psi_{h_V}^V = \left(\varphi_{\frac{h}{4}}^B \circ \varphi_{\frac{h}{2}}^A \circ \varphi_{\frac{h}{4}}^B\right) \circ \left(\varphi_{\frac{h}{4}}^B \circ \varphi_{\frac{h}{2}}^A \circ \varphi_{\frac{h}{4}}^B\right) = \varphi_{\frac{h}{4}}^B \circ \varphi_{\frac{h}{2}}^A \circ \varphi_{\frac{h}{2}}^B \circ \varphi_{\frac{h}{2}}^A \circ \varphi_{\frac{h}{4}}^B = \Psi_h.$$

A particular integrator from two-, three- or four-stage family is specified by freely choosing one, two or three parameters, respectively. Such parameters can be determined, for

example by minimizing an appropriate energy error function. There are different ways in which error function can be defined. Below we will review two possible ways of finding the parameters.

**Minimum error (ME) integrator**

In order to find the optimal parameters of $p$-order integrators, McLachlan and Atela (1992) have suggested to minimize the Euclidean norm of coefficients corresponding to $h^p$ terms in the *Hamiltonian truncation error*, defined as a difference between the Hamiltonian and the modified Hamiltonian, i.e.

$$H - \tilde{H}.$$

This idea was used later to derive the optimal coefficient for the two-stage integrator. In this case, the modified Hamiltonian can be calculated e.g. using BCH formula, as

$$\tilde{H} = H + h^2\alpha\{A, A, B\} + h^2\beta\{B, B, A\} + \mathcal{O}(h^4)$$

where

$$\alpha = \frac{6b - 1}{24},\ \beta = \frac{6b^2 - 6b + 1}{12}$$

and iterated Poisson brackets $\{A, \{A, B\}\}$ and $\{B, \{B, A\}\}$ are abbreviated as $\{A, A, B\}$ and $\{B, B, A\}$. By minimizing the error function

$$E = \alpha^2 + \beta^2,$$

McLachlan (1995) derived the integrator now known as the *minimum error* integrator with coefficient $b = 0.193183$. This integrator has been used e.g. by Takaishi (2014) for statistical sampling and by Takaishi and Forcrand (2006) for simulation of lattice quantum chromodynamics.

**Minimum expected error (BCSS) integrator**

Instead of considering Hamiltonian truncation error, Blanes et al. (2014) defined a measure of error through the expected value of the energy error $\Delta$ introduced due to numerical integration, i.e.

$$\Delta = H(\Psi_{h,L}(\boldsymbol{\theta}, \mathbf{p})) - H(\boldsymbol{\theta}, \mathbf{p}). \tag{2.24}$$

Their analysis is based on the standard harmonic oscillator[1] with equations of motion

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = p, \quad \frac{\mathrm{d}p}{\mathrm{d}t} = -\theta \tag{2.25}$$

and Hamiltonian

$$H(\theta, p) = \frac{1}{2}(\theta^2 + p^2). \tag{2.26}$$

[1]This model problem is equivalent to HMC sampling from a univariate Gaussian distribution for the parameter $\theta$, with zero mean and variance one.

For each family (2.21)–(2.23) they constructed a functional in terms of integrator's coefficients $\boldsymbol{\xi} = \{a, b_1, b_2\}$, which bounds the expected value of the energy error $\mathbb{E}(\Delta)$:

$$0 \leq \mathbb{E}(\Delta) \leq \rho(h).$$

Coefficients $\boldsymbol{\xi}$ are then chosen to minimize the function

$$\|\rho\|_{(\bar{h})} = \max_{0<h<\bar{h}} \rho(h, \boldsymbol{\xi}),$$

where the maximum step size is set as $\bar{h} = r$, with $r$ being the number of stages in the integrator. This choice of $\bar{h}$ follows from the fact that the Verlet integrator applied to standard harmonic oscillator performs well with $h \approx 1$, which is the half of its stability limit. Then an $r$-stage integrator is assumed to perform well with $h \approx r$.

Table 2.1 presents the values calculated for two-, three- and four-stage integrators (Blanes et al., 2014; McLachlan, 1995).

| BCSS | coefficients |
|---|---|
| 2-stage | $b = 0.21178$ |
| 3-stage | $a = 0.11888$<br>$b = 0.296195$ |
| 4-stage | $a_1 = 0.0713539$<br>$a_2 = 0.2685488$<br>$b_1 = 0.1916678$ |
| ME (2-stage) | $b = 0.193183$ |

TABLE 2.1: Multi-stage integrators. Two-stage coefficients coincide for velocity and position integrators. Coefficients for three- and four-stages are derived for position methods.

Multi-stage BCSS integrators were successfully employed by Attia and Sandu (2015) for non-Gaussian data assimilation problems.

In general, the suggested multi-stage integrators may improve acceptance rate and sampling efficiency of the HMC method, especially for high-dimensional problems that require smaller time steps. An $r$-stage integrator requires $r$ gradient evaluations per time step, which is an $r$ multiple of the time step of the Verlet integrator. Therefore, these integrators do not introduce computational overheads. On the other hand, the stability limit normalized by the number of stages is lower for multi-stage integrators than for the Verlet.

A choice of a numerical integrator for HMC sampling is not obvious. Further in this thesis, we will discuss alternative methods for our purposes and compare them with existing approaches.

### 2.2.4 Choice of parameters in HMC

HMC has three tunable parameters that affect the performance of the method – the integration step size $h$, the number of integration steps $L$, and the mass matrix $M$. These parameters may be chosen arbitrarily such that the validity of the method remains unharmed, except for some special cases when they might affect the ergodicity of the chain (e.g. combinations leading to a value that is a multiple of the period of a mode of the system). The goal is to tune free parameters such that the sampling efficiency is maximized and the computational cost is minimized.

**Step size** The acceptance rate in HMC depends critically on the value of a step size. The largest step size allowed is determined by the stability limit of the integrator. This limit is related to the most constrained variate, but it is not easy to identify, especially in high dimensional problems, as it can vary for the warm-up and stationary phases as well as among different regions of the state space.

Values that are too small may lead to slow space exploration and high computational costs if one wishes to maintain the trajectory length $\tau = hL$. On the other hand, values that are too large result in integration instabilities and low acceptance rates. Hence, there is a trade-off between the accuracy and computational cost of numerical integration. The choice of step size affects sampling efficiency through the trajectory length $\tau$. For a fixed $L$ and varying $h$, the computational cost is not altered; bigger values of $h$ induce more distant proposal and less correlation among samples, however, values that are too big might result in higher rejection rates which introduce more correlation among samples.

Common practice is to tune the step size by targeting the desired acceptance rate. This can be achieved e.g. by dual averaging (Hoffman and Gelman, 2014).

**Number of integration steps** The computational cost and sampling efficiency are affected by the number of integration steps $L$ through the trajectory length $\tau$. Trajectories that are too short might resemble random walk behavior. Taking trajectories that are too long might be computationally costly and even inefficient – trajectories might reverse direction and continue towards points that are closer to the initial state.

In the case of a stable step size, the number of integration steps does not affect the acceptance rate; therefore no additional correlation among samples is introduced due to rejections. However, for some complex models, one might encounter an increase in rejection rate for bigger values of $L$.

Ideally, $L$ should be large enough to ensure that successive MC samples are nearly independent. Nevertheless, this may differ across variates.

Some analysis on identifying the optimal integration time has been done by Betancourt (2016) through the use of *exhaustions* (families of appropriate integration times for a given problem). However, these termination criteria include a problem specific parameter; hence a good practical criterion in the general case is still to be developed.

Alternatively, $L$ can be adapted dynamically on the fly, for each MC step, as proposed by Hoffman and Gelman (2014). Moreover, it has been proved (Livingstone et al., 2016) that choosing the integration time dynamically leads to geometric convergence for a larger class of target distributions than in the case of fixed integration time.

General advice for both step size and number of integration steps is randomization, as first recommended by Mackenzie (1989). This helps to avoid some bad combinations of fixed values that might lead to slow convergence and non-ergodicity. These parameters can be selected independently from some chosen distributions $p(h)$ and $p(L)$ at each MC step. In fact, a new method called Randomized HMC extends this idea for the trajectory length (Bou-Rabee and Sanz-Serna, 2015).

**Mass matrix** In many problems that are dominated by global (non-varying) correlations, the mass matrix can improve HMC performance significantly. One recommendation is to assign smaller values to variates with larger variances so that the Hamiltonian flow can make more distant steps along those variates (Liu, 2008). This can be achieved by estimating covariances from the warm-up phase (Stan Development Team, 2016).

Local correlations, however, can be treated only with methods that have a position dependent matrix (e.g. Girolami and Calderhead, 2011b).

### 2.2.5 Modifications of HMC in computational statistics

Further modifications of the HMC method in computational statistics have been recently developed (see Figure 2.3).

**Variable mass matrix** Girolami and Calderhead (2011b) were the first to propose the method that explores geometric properties of the underlying distribution. Their method, called Riemann Manifold Hamiltonian Monte Carlo (RMHMC), employs a mass matrix that changes with position at every step of the integration performed by the implicit generalized leapfrog integrator. However, the Fisher-Rao metric used in the original implementation of RMHMC limits the applicability of this method.

The improved metric (mass matrix) that extends the class of problems to be successfully treated by RMHMC was later proposed by Betancourt (2013a).

Two alternatives, semi-explicit and explicit, to the implicit integrator in RMHMC, were suggested by Lan et al. (2015). In this case, the dynamics is driven in terms of velocity rather than momentum, thus the name Riemannian Manifold Lagrangian Monte Carlo (RMLMC).

**Adaptive methods** The No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014) implements an automated choice of trajectory lengths through the criterion of double backing of trajectories, while the step size is adapted using dual averaging. The extension of the NUTS to the RMHMC method was introduced by Betancourt (2013b).

Alternatively, the adaptation of HMC parameters can be carried out using Bayesian optimization or Bayesian parametric bandit approaches (Wang and Freitas, 2011; Wang et al., 2013), as formulated within the Adaptive (Riemann Manifold) Hamiltonian Monte Carlo (A(RM)HMC) method.

**Alternative integration approaches** In Section 2.2.3 we already introduced several alternative to Verlet integration schemes, which can be grouped into two classes: exponential integrators and multi-stage integrators. The exponential schemes were implemented in HMC and gave birth to the following HMC-based methods: Hilbert space HMC (Beskos et al., 2011), Split HMC (Shahbaba et al., 2014) and Exponential HMC (Chao et al., 2015).

**Delayed rejections** The idea of delaying rejections in MCMC in order to reduce the effect of correlated samples belonged to Mira (2001). Implementation of this idea into the HMC framework was, however, introduced first by Sohl-Dickstein et al. (2014) and Campos and Sanz-Serna (2015), leading to the Look Ahead HMC (LAHMC) and Extra Chance Generalized HMC (ECGHMC) methods, respectively.

**Accelerated computation by approximations** Several approaches for an approximation of the gradient of the posterior distribution were suggested recently for problems for which its calculation is computationally too demanding. In case of large datasets, one can use Stochastic gradient HMC (SGHMC) (Chen et al., 2014). Another method from this group, Kernel Hamiltonian / Kamiltonian Monte Carlo (KMC) (Strathmann et al., 2015) adaptively learns the gradient structure from the history of the Markov chain and uses it to simulate Hamiltonian dynamics. Zhang et al. (2015a) approximate the gradient structure using a neural network surrogate function, while Zhang et al. (2015b) precompute the gradient on a grid and interpolate those values. The Hamiltonian ABC (HABC) method (Meeds et al., 2015) incorporates approximate Bayesian computation within the HMC framework.

**Problem related HMC** The HMC method was further developed to deal with some specific problems, such as constrained target distributions (Betancourt, 2010; Brubaker et al., 2012; Lan et al., 2014b), multimodal distributions (Lan et al., 2014a; Betancourt, 2014), hierarchal models (Betancourt and Girolami, 2015; Zhang and Sutton, 2014). HMC was also extended for discrete distributions by using continuous relaxations from discrete to continuous variables (Zhang et al., 2012; Pakman and Paninski, 2013).

**Others** The tempering HMC method (Meent et al., 2014) formulates both parallel tempering and tempered transitions through recursive subsampling of observations.

The Hamiltonian Annealed Importance Sampling approach (Sohl-Dickstein and Culpepper, 2012) was designed for evidence estimation problems.

Within the framework for a generalization of the Metropolis-Hastings method Calderhead (2014) suggested the idea of improving the performance of HMC by making use

of all integration steps from the leapfrog trajectory. A similar idea was formulated under the Recycled HMC method and investigated by Nishimura and Dunson (2015).

Another popular method is Metropolis Adjusted Langevin Algorithm (MALA) (Kennedy, 1990). It can be seen as a special case of HMC for single step trajectories. The proposal can be defined as

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \frac{h^2}{2} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) + h\mathbf{u},$$

where $\mathbf{u} \sim \mathcal{N}(0, I)$.

The recently introduced Monomial Gamma HMC (Zhang et al., 2016) uses a different kinetic function, which corresponds to the monomial Gamma distribution for momentum variables. The method leads to potentially better mixing of the chain. On the other hand, it has an additional parameter to tune, and it might lead to numerical instabilities for high dimensions.

### 2.2.6 Modifications of HMC in computational sciences

Among the modifications introduced in computational sciences, the most important ones are partial momentum update and sampling with modified energies.

Before going into further details, we notice that as far as HMC is concerned, there are often some differences in notations and interpretations between computational statistics and computational sciences. For example in computational sciences, the mass matrix $M$ is not considered a preconditioning simulation parameter but rather it is determined by the simulated system. The target distribution incorporates a real temperature through

$$\exp(-\beta H(\mathbf{x}, \mathbf{p})),$$

where $\beta = 1/k_B T$ is the inverse temperature with the Boltzmann constant $k_B$. The total energy of the system corresponds to the true Hamiltonian. The position vector in computational sciences is denoted e.g. as $\mathbf{x}$.

**The Partial Momentum Update**

The partial momentum update (in contrast to the complete momentum update) was introduced by Horowitz (1991) within Generalized guided Monte Carlo, a method that relies on a single step of Hamiltonian dynamics. This method is also known as second order Langevin Monte Carlo (L2MC). The purpose of this technique was to retain more dynamical information of the simulated system.

Kennedy and Pendleton (2001) formalized this idea in the Generalized Hybrid Monte Carlo (GHMC) method. GHMC is defined as the concatenation of two steps: Molecular Dynamics Monte Carlo (MDMC) and Partial Momentum Update (PMU).

This method differs from HMC in the momentum update step – the complete reset of the momentum for initiating a new trajectory is replaced with the partial momentum update. The current momentum is mixed with an independent and identically distributed (i.i.d.)

Gaussian noise vector $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$ to obtain

$$\begin{aligned} \mathbf{p}^* &= \cos(\phi)\mathbf{p} + \sin(\phi)\mathbf{u} \\ \mathbf{u}^* &= -\sin(\phi)\mathbf{p} + \cos(\phi)\mathbf{u}, \end{aligned} \tag{2.27}$$

where $\phi \in (0, \pi/2]$ controls the level of noise.

At this point, a Metropolis test is not needed, because the variables $\mathbf{p}^*$ and $\mathbf{u}^*$ are distributed according to the same Gaussian distribution as $\mathbf{p}$ and $\mathbf{u}$. This follows from the orthogonality of the transformation (2.27).

The parameter $\phi$ introduces extra control over the sampling efficiency of the method and may lead to the superior performance of GHMC over HMC. It updates the momentum between trajectories partially so that consecutive trajectories tend to move in more similar directions.

Since momentum is not discarded, the method incorporates a momentum flip

$$\mathcal{F}(\mathbf{x}, \mathbf{p}) = (\mathbf{x}, -\mathbf{p})$$

upon rejection, in order to ensure the detailed balance condition is satisfied.

A Molecular Dynamics Monte Carlo step is defined in the same way as in the HMC method.

The algorithm of GHMC is presented below in Algorithm 3.

**Algorithm 3** Generalized Hybrid Monte Carlo

1: **Input:** $N$: number of Monte Carlo samples
$h$: time step
$L$: number of integration steps
$M$: mass matrix
$T$: temperature ($\beta = 1/k_B T$)
$\Psi_{h,L}$: numerical integrator
$\phi \in (0, \pi/2]$: noise parameter
2: Initialize $(\mathbf{x}^0, \mathbf{p}^0)$
3: **for** $n = 1, \ldots, N$ **do**
4: $(\mathbf{x}, \mathbf{p}) = (\mathbf{x}^{n-1}, \mathbf{p}^{n-1})$
5: Partial momentum update

$$\begin{aligned} \mathbf{p}^* &= \cos(\phi)\mathbf{p} + \sin(\phi)\mathbf{u} \\ \mathbf{u}^* &= -\sin(\phi)\mathbf{p} + \cos(\phi)\mathbf{u} \end{aligned}$$

where $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$
6: Generate a proposal by integrating Hamiltonian dynamics

$$(\mathbf{x}', \mathbf{p}') = \Psi_{h,L}(\mathbf{x}, \mathbf{p}^*)$$

7: Calculate the acceptance probability

$$\alpha = \min\{1, \exp(-\beta(H(\mathbf{x}', \mathbf{p}') - H(\mathbf{x}, \mathbf{p}^*))\}$$

8: Metropolis test
  Draw $u \sim \mathcal{U}(0, 1)$
  **if** $u < \alpha$
    $(\mathbf{x}^n, \mathbf{p}^n) = (\mathbf{x}', \mathbf{p}')$ {accept proposal}
  **else**
    $(\mathbf{x}^n, \mathbf{p}^n) = \mathcal{F}(\mathbf{x}, \mathbf{p}^*)$ {reject proposal and flip momentum}
  **end if**
9: **end for**

Note that the formulation above differs from the original one (Kennedy and Pendleton, 2001) in that there is a reduced number of momentum flips performed. In the original formulation, the momentum flip is applied before partial momentum refreshment and once again upon acceptance, instead of rejection; thus more momentum flips are needed in this case. The two formulations are equivalent, however.

Some well-known methods can be considered as special cases of GHMC:

- The standard HMC algorithm is a special case of GHMC if $\phi = \pi/2$. The momentum flips may be ignored in this case since $\mathbf{p}^* = \mathbf{u}$ and the previous momentum is entirely discarded.
- If additionally $L = 1$, this method corresponds to the MALA method.
- The Generalized guided MC or Langevin Monte Carlo algorithm corresponds to a single MD step ($L = 1$) and an arbitrary $\phi$.
- In the case of all MD proposals being accepted and $\phi = \sqrt{2\gamma\Delta t} \ll 1$ the method coincides with stochastic Langevin Dynamics (LD), where $\gamma > 0$ plays the role of the friction coefficient.
- If $\phi = 0$ and all trajectories are accepted, meaning that one long trajectory is produced, the Molecular Dynamics method is recovered.

The special cases of GHMC are summarized in Table 2.2.

Applications of the GHMC method to date include mainly molecular simulations. The behavior of non-special cases of GHMC is not well studied in statistical simulations, with only a few exceptions, e.g. in (Sohl-Dickstein, 2012; Sohl-Dickstein et al., 2014).

**Sampling with respect to modified density**

The performance of the HMC method degrades for large systems and time steps due to errors in Hamiltonians resulting from numerical integration. As noted in Section 2.1.3, modified Hamiltonians are conserved with symplectic integrators to a higher accuracy than true Hamiltonians. The idea of implementing the HMC method with respect to a modified

| Metropolis test | $\phi$ | $L$ | Method |
|---|---|---|---|
| ✓ | $\pi/2$ | arbitrary | **HMC** |
| ✓ | $\pi/2$ | 1 | **MALA** |
| ✓ | arbitrary | 1 | **L2MC** |
| ✗ | $\sqrt{2\gamma\Delta t} \ll 1, \gamma > 0$ | 1 | **LD** |
| ✗ | 0 | arbitrary | **MD** |

Table 2.2: Special cases of GHMC.

density by using the modified Hamiltonian in the Metropolis test was suggested by Izaguirre and Hampton (2004). The resulting method, Shadow Hybrid Monte Carlo (SHMC), consists of the following steps. First, momentum is drawn from a Gaussian distribution until it is accepted according to the modified density $\tilde{\pi}(\mathbf{x}, \mathbf{p}) \propto \exp(-\beta \tilde{H}(\mathbf{x}, \mathbf{p}))$, where

$$\tilde{H}(\mathbf{x}, \mathbf{p}) = \max\{H(\mathbf{x}, \mathbf{p}), H_h(\mathbf{x}, \mathbf{p}) - C\},$$

$H_h$ is an approximation of the modified Hamiltonian $\tilde{H}$, calculated as suggested by Skeel and Hardy (2001), and $C$ is a tunable parameter. This might be costly, due to a number of evaluations of the modified energy. In the second step, MD is performed using a symplectic integrator augmented with a scalar variable, which is needed for the calculation of shadow Hamiltonians. A Metropolis test is then evaluated with respect to $\tilde{\pi}$. Importance sampling reweighting is required for computing averages in order to recover the canonical density. The performance of SHMC is limited by the need for fine tuning the parameter $C$ and by evaluation of a non-separable shadow Hamiltonian.

The SHMC was modified by Sweet et al. (2009) by replacing a non-separable shadow Hamiltonian with the separable shadow Hamiltonian of order four, defined as

$$\tilde{H}^{[4]} = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\mathbf{x}) + \frac{h^2}{24} U_{\mathbf{x}}^T(\mathbf{x}) M^{-1} U_{\mathbf{x}}(\mathbf{x}).$$

This method, which used $\tilde{\pi}(\mathbf{x}, \mathbf{p}) \propto \exp(-\beta \tilde{H}^{[4]}(\mathbf{x}, \mathbf{p}))$ as the target density and a corresponding processed[2] leapfrog integrator, was called the Separable Shadow Hybrid Monte Carlo (S2HMC).

The first method to incorporate both the partial momentum update and sampling with respect to a modified density was introduced by Akhmatskaya and Reich (2006) and called Targeted Shadow Hybrid Monte Carlo (TSHMC). However, the Generalized Shadow Hybrid Monte Carlo (GSHMC) method formulated by Akhmatskaya and Reich (2008) appears the most efficient among the listed methods. Full details of this method are given in the following section.

[2] This processed leapfrog integrator consists of a preprocessing and a postprocessing step, each involving a fixed point computation.

## 2.3 Generalized Shadow Hybrid Monte Carlo

### 2.3.1 History

The GSHMC method was introduced in 2008 for sampling in molecular simulation (Akhmatskaya and Reich, 2008). Its purpose was to enable sampling of large complex systems while retaining the dynamical information. This is achieved by employing the modified energy for sampling and by partially updating momentum. The former leads to lower discretization errors, which implies higher acceptance rates for large system sizes as well as a reduced negative impact of the undesired momentum flips.

The method was patented by Fujitsu in the UK (Akhmatskaya et al., 2009c) and the US (Akhmatskaya et al., 2011). Due to IPR issues, there were difficulties with the implementation of the method in open source software. This changed in November 2015, when Fujitsu issued the license giving permission to use the patented method in open source software and a permission to Elena Akhmatskaya to implement and use know-how.

GSHMC proved to be successful in simulations of complex molecular systems in Biology and Chemistry (Wee et al., 2008; Akhmatskaya and Reich, 2012; Escribano et al., 2013; Akhmatskaya et al., 2013; Fernández-Pendás et al., 2014). Initially designed for atomistic simulations in the canonical (NVT) ensemble, in which the number of atoms N, volume V and temperature T are kept constant, and the isobaric-isothermal (NPT) ensemble, in which N, T and pressure P are kept constant, but implemented for the canonical ensemble only, the method has been developed further to cover a range of problems. Multi-scale simulations can be treated with the multiple-time stepping GSHMC (MTS-GSHMC) method (Escribano et al., 2014), while coarse-grained systems with the Meso-GSHMC method (Akhmatskaya and Reich, 2011; Terterov et al., 2013). The NPT-GSHMC (Fernández-Pendás et al., 2014) ensures rigorous pressure control.

### 2.3.2 Formulation

The objective of the GSHMC method is to maintain a high acceptance rate while retaining the dynamical information in simulations. It is achieved by combining the partial momentum update, as introduced in the GHMC method, with importance sampling with respect to a modified density in the appropriate manner.

Sampling is performed with respect to a modified canonical density

$$\tilde{\pi}(\mathbf{x}, \mathbf{p}) \propto \exp(-\beta \tilde{H}^{[k]}(\mathbf{x}, \mathbf{p})),$$

where $\tilde{H}^{[k]}$ is the $k$th order modified Hamiltonian (see Section 2.1.3) that approximates the true Hamiltonian as

$$\tilde{H}^{[k]} = H + \mathcal{O}(h^p),$$

for a $p$-order numerical integrator. In the case $p = 2$ the order of the modified Hamiltonian is $k \geq 4$ (cf. Equation 2.13).

The method involves two major steps, the Partial Momentum Monte Carlo (PMMC) step, and the Molecular Dynamics Monte Carlo (MDMC) step. The partial momentum update allows for keeping the dynamical information during the simulation similar to a stochastic Langevin dynamics simulation, in which the friction coefficient restricts the noise added to the momentum. We note that momenta are no longer distributed according to the normal distribution (2.16) under the modified density $\tilde{\pi}$. The momentum update step, therefore, becomes more complex and it is combined with the modified Metropolis test. The only difference in the MDMC step of the GHMC method is that in the Metropolis test the *modified* Hamiltonian is used instead of the true Hamiltonian.

Since sampling is performed with respect to the modified distribution, the importance weights have to be taken into account when calculating averages of quantities of interest.

In the following, we provide full details on how to calculate shadow Hamiltonians, how to perform the PMMC and MDMC steps and the implementation of the reweighting procedure.

#### 2.3.2.1 Shadow Hamiltonians

The original formulation provides the expression of the $k$th order modified Hamiltonian for the leapfrog method (Akhmatskaya and Reich, 2008; Akhmatskaya et al., 2009c), which in the case of $k = 4$ has the explicit form

$$\tilde{H}^{[4]} = \frac{1}{2}\dot{\mathbf{X}}[M\dot{\mathbf{X}}] + U(\mathbf{X}) + \frac{h^2}{24}\Big(2\dot{\mathbf{X}}[M\mathbf{X}^{(3)}] - \ddot{\mathbf{X}}[M\ddot{\mathbf{X}}]\Big), \tag{2.28}$$

where $\mathbf{X}(t) \in \mathbb{R}^D$ is the unique interpolation polynomial of degree four, constructed for $t_n, n \in \{0, L\}$ from a given numerical trajectory $\{\mathbf{x}^i\}_{i=-2}^{L+2}$, passing through points

$$\mathbf{X}(t_i) = \mathbf{x}^i, \ i = n-2, \ldots, n, \ldots, n+2.$$

The derivatives of the position vector are approximated by the centered differences method.

The number of additional gradient evaluations per Monte Carlo step in GSHMC compared with HMC may vary between 3 and 10 for the modified Hamiltonian of order 4. The least number of additional evaluations can be achieved if previously calculated modified Hamiltonians and gradients are stored and used for further calculation when possible.

#### 2.3.2.2 PMMC

In this step, the partial momentum update is combined with the modified Metropolis test.

The *Partial Momentum Update (PMU)* is identical to the one in GHMC method, given by

$$\begin{aligned} \mathbf{p}^* &= \cos(\phi)\mathbf{p} + \sin(\phi)\mathbf{u} \\ \mathbf{u}^* &= -\sin(\phi)\mathbf{p} + \cos(\phi)\mathbf{u} \end{aligned} \tag{2.29}$$

with $\phi \in (0, \pi/2]$, the noise vector $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1}M)$ i.i.d. and $\beta = 1/k_BT$.

The *Modified Metropolis test*: the proposal $(\mathbf{p}^*, \mathbf{u}^*)$ is accepted according to

$$(\bar{\mathbf{p}}, \bar{\mathbf{u}}) = \begin{cases} (\mathbf{p}^*, \mathbf{u}^*) & \text{with probability } \mathcal{P} \\ (\mathbf{p}, \mathbf{u}) & \text{otherwise} \end{cases} \tag{2.30}$$

where

$$\mathcal{P} = \min\left\{1, \frac{\exp\left(-(\tilde{H}(\mathbf{x}, \mathbf{p}^*) + \frac{1}{2}(\mathbf{u}^*)^T M^{-1}\mathbf{u}^*)\right)}{\exp\left(-(\tilde{H}(\mathbf{x}, \mathbf{p}) + \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u})\right)}\right\}. \tag{2.31}$$

This step can be considered as a standard HMC method in which the vector $\mathbf{x}$ is fixed, the vector $\mathbf{p}$ plays a role of the "position" and the noise vector $\mathbf{u}$ becomes "conjugate momenta". The extended "Hamiltonian"

$$\hat{H}(\mathbf{x}, \mathbf{p}, \mathbf{u}) = \tilde{H}(\mathbf{x}, \mathbf{p}) + \frac{1}{2}\mathbf{u}^\intercal M^{-1}\mathbf{u} \tag{2.32}$$

defines the extended reference density $\hat{\pi}(\mathbf{x}, \mathbf{p}, \mathbf{u}) \propto \exp(-\beta\hat{H}(\mathbf{x}, \mathbf{p}, \mathbf{u}))$.

Note that additional steps forward and backward in time need to be performed to evaluate the interpolation polynomial $\mathbf{X}$ and thus shadow Hamiltonian $\tilde{H}(\mathbf{x}, \mathbf{p}^*)$. The noise vectors $\mathbf{u}$ and $\bar{\mathbf{u}}$ are discarded.

#### 2.3.2.3 MDMC

The MDMC step of GSHMC differs from GHMC only in the Metropolis test.

*Molecular dynamics* starts from the current state $(\mathbf{x}, \mathbf{p}) = (\mathbf{x}, \bar{\mathbf{p}})$ and integrates $L$ steps of Hamiltonian dynamics with a symplectic and time-reversible numerical integrator and a time step $h$ thus generating the proposal $(\mathbf{x}', \mathbf{p}') = \Psi_\tau(\mathbf{x}, \mathbf{p})$, where $\tau = Lh$. If the modified Hamiltonian is defined as in (2.28) then $\Psi_h(\mathbf{x}, \mathbf{p})$ is the Verlet integrator, i.e.

$$\begin{aligned} \mathbf{p} &= \mathbf{p} - \frac{h}{2}U_{\mathbf{x}}(\mathbf{x}) \\ \mathbf{x} &= \mathbf{x} + hM^{-1}\mathbf{p} \\ \mathbf{p} &= \mathbf{p} - \frac{h}{2}U_{\mathbf{x}}(\mathbf{x}). \end{aligned}$$

The *Monte Carlo* step consists of the Metropolis test in which the new state is assigned as

$$(\mathbf{x}^{new}, \mathbf{p}^{new}) = \begin{cases} (\mathbf{x}', \mathbf{p}') & \text{with probability} \quad \alpha = \min\left\{1, \exp(-\beta\Delta\tilde{H})\right\} \\ \mathcal{F}(\mathbf{x}, \mathbf{p}) & \text{otherwise} \end{cases} \tag{2.33}$$

where $\Delta\tilde{H} = \tilde{H}(\mathbf{x}', \mathbf{p}') - \tilde{H}(\mathbf{x}, \mathbf{p})$.

The average energy fluctuation $\mathbb{E}(\Delta\tilde{H})$ is

$$\mathbb{E}(\Delta\tilde{H}) = \mathcal{O}(Dh^{2k}),$$

where $k \geq 4$ is the order of shadow Hamiltonian. This means that the energy error depends on the order of shadow Hamiltonian rather than on order of an integrator (cf. Equation (2.19)). Therefore, an increase in the dimension of the problem can be counterbalanced by an increase in the order of the shadow Hamiltonian. This opens the possibility to maintain high acceptance rates even in simulations of high-dimensional systems while using an integrator of the same order of accuracy.

#### 2.3.2.4 Re-weighting

If $\Omega_n, n = 1, 2, \ldots, N$ are values of the observables along a sequence of states $(\mathbf{x}^n, \mathbf{p}^n)$, then the averages are calculated as

$$\langle \Omega \rangle = \frac{\sum_{n=1}^{N} w_n \Omega_n}{\sum_{n=1}^{N} w_n}, \tag{2.34}$$

where importance weights take into account the difference between the desired target distribution $\pi$ and the modified distribution $\tilde{\pi}$ from which samples are drawn. These importance weights are therefore given by

$$w_n = \exp\Big( -\beta(H(\mathbf{x}^n, \mathbf{p}^n) - \tilde{H}(\mathbf{x}^n, \mathbf{p}^n))\Big). \tag{2.35}$$

The GSHMC method is presented below in Algorithm 4.

**Algorithm 4** Generalized Shadow Hybrid Monte Carlo

**Input:** $N$: number of Monte Carlo samples
$h$: time step
$L$: number of integration steps
$M$: mass matrix
$T$: temperature ($\beta = 1/k_B T$)
$\phi \in (0, \pi/2]$: noise parameter
Initialize $(\mathbf{x}^0, \mathbf{p}^0)$
**for** $n = 1, \ldots, N$ **do**
  Calculate the shadow Hamiltonian at $(\mathbf{x}, \mathbf{p}) = (\mathbf{x}^{n-1}, \mathbf{p}^{n-1})$
PMMC step
  Draw $\mathbf{u} \sim \mathcal{N}(0, \beta^{-1} M)$
  Generate a proposal

$$\begin{aligned} \mathbf{p}^* &= \cos(\phi)\mathbf{p} + \sin(\phi)\mathbf{u} \\ \mathbf{u}^* &= -\sin(\phi)\mathbf{p} + \cos(\phi)\mathbf{u} \end{aligned}$$

  Calculate the shadow Hamiltonian $\tilde{H}(\mathbf{x}, \mathbf{p}^*)$
  Accept the proposed momentum $\bar{\mathbf{p}} = \mathbf{p}^*$ with probability

$$\mathcal{P} = \min\left\{1, \frac{\exp\big(-\beta(\tilde{H}(\mathbf{x}, \mathbf{p}^*) + \frac{1}{2}(\mathbf{u}^*)^T M^{-1} \mathbf{u}^*)\big)}{\exp\big(-\beta(\tilde{H}(\mathbf{x}, \mathbf{p}) + \frac{1}{2}\mathbf{u}^T M^{-1} \mathbf{u})\big)}\right\}$$

otherwise set $\bar{\mathbf{p}} = \mathbf{p}$
MDMC step

9: Generate a proposal by integrating Hamiltonian dynamics with time step $h$ over $L$ steps

$$(\mathbf{x}', \mathbf{p}') = \Psi_{h,L}(\mathbf{x}, \bar{\mathbf{p}})$$

10: Calculate the shadow Hamiltonian $\tilde{H}(\mathbf{x}', \mathbf{p}')$

11: Calculate the acceptance probability

$$\alpha = \min\{1, \exp(-\beta(\tilde{H}(\mathbf{x}', \mathbf{p}') - \tilde{H}(\mathbf{x}, \bar{\mathbf{p}}))\}$$

12: Metropolis test

$$(\mathbf{x}^n, \mathbf{p}^n) = \begin{cases} (\mathbf{x}', \mathbf{p}') & \text{with probability} \quad \alpha \\ \mathcal{F}(\mathbf{x}, \bar{\mathbf{p}}) & \text{otherwise} \end{cases}$$

13: Compute the weight

$$w_n = \exp(-\beta(H(\mathbf{x}^n, \mathbf{p}^n) - \tilde{H}(\mathbf{x}^n, \mathbf{p}^n)))$$

14: **end for**

15: Calculate the average of an observable $\Omega(\mathbf{x}, \mathbf{p})$ as

$$\langle \Omega \rangle = \frac{\sum_{n=1}^{N} w_n \Omega_n}{\sum_{n=1}^{N} w_n}$$

---

Note that the GSHMC method introduces computational overheads compared to HMC due to two evaluations of the shadow Hamiltonian per MC step. This means that for short trajectories the overheads might be significant, but for long trajectories, which require many gradient calculations, they become negligible.

### 2.3.3 Choice of parameters

**Time step** If $h$ is chosen to be too short, the computational cost of the simulation is increased, while choices of too long ones enlarge the integration inaccuracies and can potentially lead to the higher rejection rates.

**Number of integration steps** Similar to HMC, values of $L$ that are too small reduce sampling efficiency. In addition, they imply more frequent calculations of shadow Hamiltonians for a fixed simulation time, which may well introduce significant computational overheads.

**Angle** Smaller values of $\phi$ are advisable for retaining the dynamical information of the system, but values that are too small may reduce sampling efficiency. On the other hand, values that are too large increase momenta rejection rates and do not reproduce dynamical properties of a simulated system.

**Order of shadow Hamiltonian** Large orders might be computationally demanding but for some problems using orders that are too small may not provide a good approximation of the true Hamiltonian and consequently the simulation properties.

A general recommendation for GSHMC is to choose a combination of parameters such that simultaneous rejections of both momentum and position are kept sufficiently small (Akhmatskaya and Reich, 2012).

### 2.3.4 Applications

One of the successful applications of the GSHMC method was the study of a peptide toxin interacting with a phospholipid bilayer (Wee et al., 2008). At the beginning of the simulation, a toxin is placed at the center of the membrane. The focus is on measuring a distance from the center to the surface of the membrane and an orientation of the toxin with respect to the membrane's surface.

Both GSHMC and conventional MD simulation found the most probable position and orientation. Nevertheless, GSHMC offered approximately an eight times increase in sampling efficiency, measured in terms of autocorrelation functions for distances of toxin from a bilayer center (see Figure 2.1).

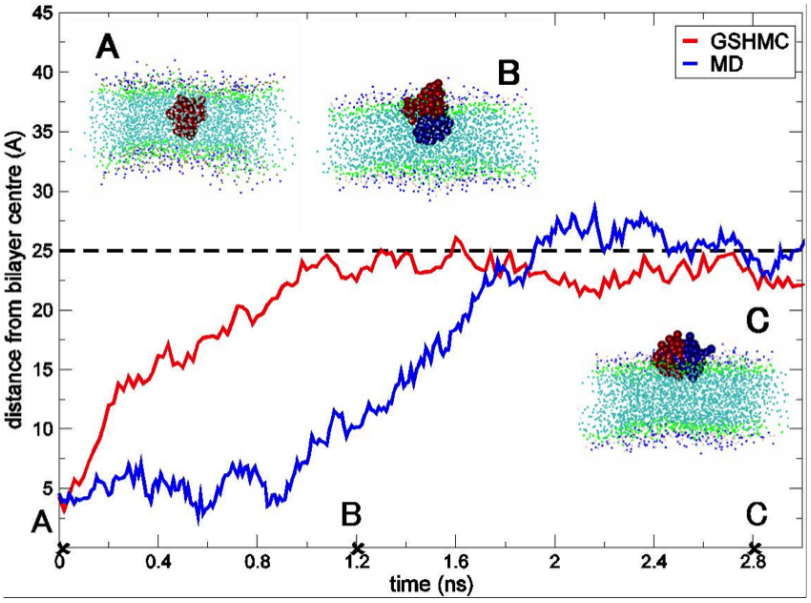


Figure 2.1: Comparison of GSHMC and Molecular Dynamics (MD) performance for peptide toxin / bilayer system. Image taken from (Akhmatskaya and Reich, 2011).

Another successful application is the study of the morphology development of multiphase polymers (Asua and Akhmatskaya, 2011). The simulation model accounts for the formation of graft copolymer and can be used to predict the particle morphology.

Analysis of autocorrelation functions of radii of gyration indicates that GSHMC finds the equilibrium morphology of graft polymer up to seven times faster than Langevin Dynamics – the methodology often used for simulation of polymer systems (Figure 2.2). Moreover,

using LD it was not possible to identify the optimal choice of the friction coefficient $\gamma$, the important parameter in modeling particles morphologies, while for GSHMC it is the one that corresponds to the optimal choice of the parameter $\phi$ in the terms of the momentum acceptance rate.

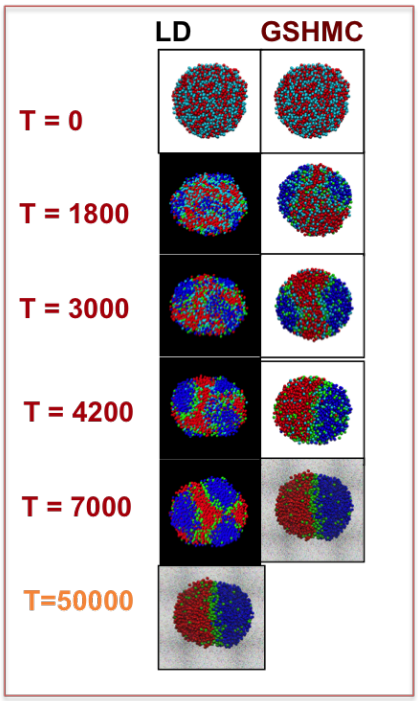


Figure 2.2: In Situ Formation of Graft Copolymer: Langevin Dynamics vs. GSHMC. Image from the presentation at The International Conference on Scientific Computation and Differential Equations, SciCADE 2015, Potsdam, Germany.

### 2.3.5 GSHMC in statistics

The GSHMC method has never been investigated for solving statistical inference problems although its applicability has been recognized. A formulation under the name Generalized shadow Hamiltonian Monte Carlo (GSHmMC) was given in (Akhmatskaya and Reich, 2008; Akhmatskaya and Reich, 2012).

For a statistical model with unknown parameter vector $\boldsymbol{\theta}$, the target density $\pi(\boldsymbol{\theta})$ is written as

$$\pi(\boldsymbol{\theta}) \propto \exp(-U(\boldsymbol{\theta})).$$

Introducing the momentum $\mathbf{p}$ conjugate to $\boldsymbol{\theta}$, with the 'mass' matrix $M$ (a preconditioner), the guided Hamiltonian is defined in the usual manner as

$$H(\boldsymbol{\theta}, \mathbf{p}) = U(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p}.$$

GSHmMC sampling is performed with respect to a modified canonical density

$$\tilde{\pi}(\boldsymbol{\theta}, \mathbf{p}) \propto \exp(-\tilde{H}(\boldsymbol{\theta}, \mathbf{p})),$$

where $\tilde{H}$ is an approximation of the modified Hamiltonian of a chosen order.

Therefore, the formulation of GSHmMC is equivalent to GSHMC, with the only difference that positions, momenta, mass matrix, temperature, and Hamiltonian do not have a physical interpretation.

## 2.4 Summary

The HMC method has proved to be a successful and valuable technique for a range of problems in computational statistics. The efficient use of gradient information of the posterior distribution allows to overcome the random walk behavior typical of the Metropolis-Hastings Monte Carlo method.

On the other hand, the performance of HMC deteriorates exponentially, in terms of acceptance rates, with respect to the system's size and the step size due to errors introduced by numerical approximations (Izaguirre and Hampton, 2004). Many rejections induce high correlations between samples and reduce the efficiency of the estimator. Thus, in systems with large numbers of parameters, or latent parameters, or when the data set of observations is very large, efficient sampling might require a substantial number of evaluations of the posterior distribution and its gradient. This may be computationally too demanding for HMC. In order to maintain the acceptance rate for larger systems at a high level, one should either decrease the step size or use a higher order numerical integrator, which is usually impractical for large systems.

Ideally, one would like to have a sampler that increases acceptance rates, converges fast, improves sampling efficiency and whose optimal simulation parameters are not difficult to determine.

In the following chapters, we provide a careful and detailed investigation of whether the GSHMC method, adapted to statistical applications, can compete with the state-of-the-art HMC method. Furthermore, we present some extensions and investigate in which settings they provide the most benefit.

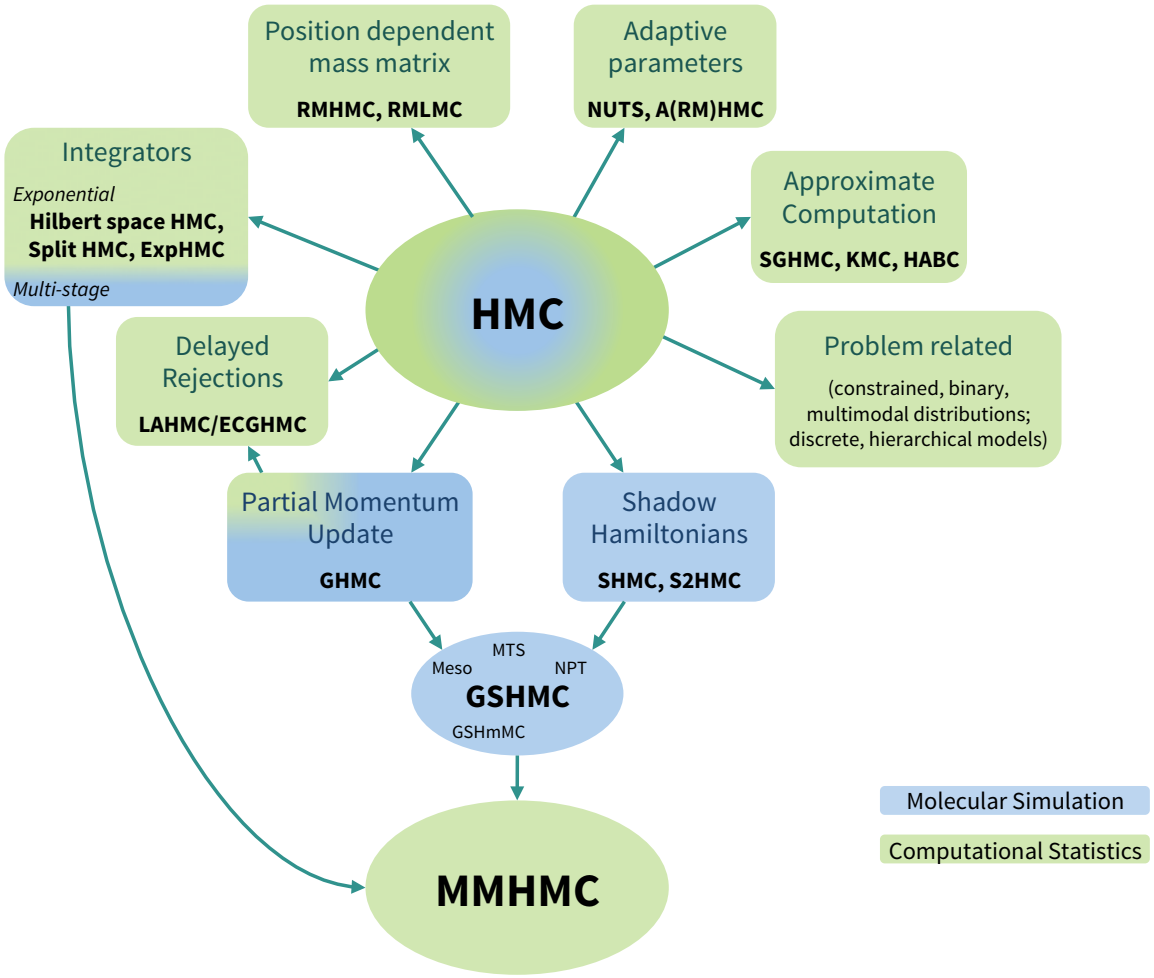


FIGURE 2.3: Evolution and relationships between HMC methods.

# 3

# Mix & Match Hamiltonian Monte Carlo

## 3.1 Preface

The GSHMC method was originally designed for molecular simulation. The differences and potential problems in adapting the method for computational statistics are the following.

1. All elements and parameters of the method, such as Hamiltonians, momenta, positions, do not have a physical interpretation and there are no natural hints regarding a reasonable choice of parameters. Contrary to molecular simulation, randomized simulation parameters for MD trajectories are preferable.

2. Due to the complex structure of target distributions, the resulting Hamiltonians are highly oscillatory. This kind of problem may require higher orders of modified Hamiltonians or better Hamiltonian conservation by integrators to avoid a loss of accuracy and sampling efficiency. Thus, an appropriate choice of the numerical integrator is not obvious and consequently neither is the form of the modified Hamiltonian, which depends directly on the choice of the integrator.

3. Hierarchical/latent-variable models, which are not typical in molecular simulation, require tuning of multiple non-independent sets of simulation parameters.

4. Simulations in transformed non-canonical space are often unavoidable due to constraints in parameter space.

The potential **advantage** of GSHMC compared to HMC is enhanced sampling as a consequence of: (i) higher acceptance rates, achieved due to better conservation of modified Hamiltonians by symplectic integrators; (ii) an access to second-order information about

the target distribution; (iii) an additional parameter $\phi$ for improving performance. Thus, the convergence to the target distribution might be faster.

On the other hand, potential **disadvantages** include one more parameter that should be tuned and some extra computational cost that is introduced through computation of modified Hamiltonians for each proposal and an additional Metropolis test for momentum update step.

In this chapter, we present the Mix & Match Hamiltonian Monte Carlo (MMHMC) method which is based on the GSHMC method but modified, enriched with new features and adapted specially to computational statistics. We also provide details of the implementation of MMHMC and present our software package HaiCS (details in Chapter 5), which offers implementation of several HMC based samplers including MMHMC as well as a range of popular statistical models. The modifications of GSHMC that led to the MMHMC method include:

- Derivation of novel multi-stage numerical integrators, as alternatives to the Verlet integrator, which can enhance accuracy in calculation of (modified) Hamiltonians.
- New formulations of modified Hamiltonians that allow for (i) employing the proposed and existing multi-stage integrators; (ii) efficient implementation using quantities available from a simulation; (iii) using non-canonical transformations of parameters that are being sampled.
- Incorporating momentum updates within the Metropolis test, resulting in less frequent calculation of derivatives in certain cases.
- An extension of the reduced momentum flipping technique to the methods sampling with modified Hamiltonians, which lessens the potentially negative impact of reverse trajectories.

In the following, we provide details on each modification to the original GSHMC. In Section 3.2.1 we cover new formulations of modified Hamiltonians. Novel numerical integrators for the Hamiltonian Dynamics step are introduced in Section 3.2.2. Section 3.2.3 includes a new Metropolis test for momentum, which incorporates the partial update, and a few alternative strategies for the momentum update. A new Metropolis test for reduced momentum flips upon rejections is presented in Section 3.2.4. In each of these sections, we include some numerical results demonstrating the impact of the proposed techniques on the performance of the method. We draw conclusions from the numerical tests taking into account the acceptance rates, effective sample size (ESS) and computational time. ESS is a commonly used metric for the sampling performance of MCMC methods (Geyer, 1992), which gives the number of effectively uncorrelated samples among all collected. More details can be found in Chapter 6, where we provide a new ESS metric, adjusted to weighted data. We also discuss a choice of simulation parameters and conclude this chapter with a summary of the current state of the MMHMC method.

## 3.2 Formulation

The MMHMC method aims at sampling a random variable of unknown parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ with the distribution

$$\pi(\boldsymbol{\theta}) \propto \exp(-U(\boldsymbol{\theta})).$$

This is achieved indirectly, as shown in Figure 3.1. MMHMC performs HMC importance sampling on the joint state space of parameters and momenta $(\boldsymbol{\theta}, \mathbf{p})$ with respect to a modified Hamiltonian $\tilde{H}$. The importance sampling distribution is defined as

$$\tilde{\pi}(\boldsymbol{\theta}, \mathbf{p}) \propto \exp(-\tilde{H}(\boldsymbol{\theta}, \mathbf{p})).$$

The target distribution on the joint state space $\pi(\boldsymbol{\theta}, \mathbf{p}) \propto \exp(-H(\boldsymbol{\theta}, \mathbf{p}))$, with respect to the true Hamiltonian $H$, is recovered through importance reweighting and finally, the desired distribution $\pi(\boldsymbol{\theta})$ can be computed by marginalizing momenta variables.

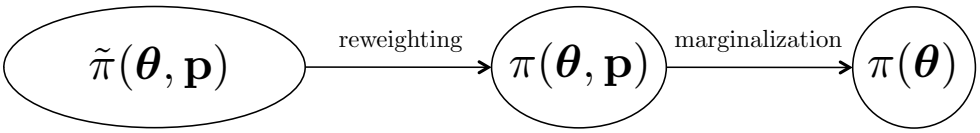


FIGURE 3.1: MMHMC sampling.

MMHMC consists of the three main steps:

1. **Partial Momentum Monte Carlo (PMMC)** – Momentum is partially updated using a noise vector $\mathbf{u} \sim \mathcal{N}(0, M)$ and accepted according to the extended modified distribution $\hat{\pi} \propto \exp(-\hat{H})$ with $\hat{H}$ defined as in (2.32).

2. **Hamiltonian Dynamics Monte Carlo (HDMC)** – A proposal $(\boldsymbol{\theta}', \mathbf{p}')$ is generated by simulating Hamiltonian dynamics using a symplectic and reversible numerical integrator and accepted with the Metropolis criterion corresponding to the modified distribution $\tilde{\pi} \propto \exp(-\tilde{H})$ as

$$(\boldsymbol{\theta}^{new}, \mathbf{p}^{new}) = \begin{cases} (\boldsymbol{\theta}', \mathbf{p}') & \text{with probability} \quad \alpha = \min\left\{1, \exp(-\Delta\tilde{H})\right\} \\ \mathcal{F}(\boldsymbol{\theta}, \mathbf{p}) & \text{otherwise} \end{cases} \tag{3.1}$$

   where $\mathcal{F}(\boldsymbol{\theta}, \mathbf{p})$ flips the momentum in the case of rejection and $\Delta\tilde{H} = \tilde{H}(\boldsymbol{\theta}', \mathbf{p}') - \tilde{H}(\boldsymbol{\theta}, \mathbf{p})$.

3. **Reweighting** – The estimation of the integral

$$I = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$$

makes use of the standard technique for importance samplers. The integral is rewritten as

$$\begin{aligned} I =& \mathbb{E}_\pi[f] = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta},\mathbf{p})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p} = \int f(\boldsymbol{\theta})\frac{\pi(\boldsymbol{\theta},\mathbf{p})}{\tilde{\pi}(\boldsymbol{\theta},\mathbf{p})}\tilde{\pi}(\boldsymbol{\theta},\mathbf{p})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p} \\ &= \int f(\boldsymbol{\theta})w(\boldsymbol{\theta},\mathbf{p})\tilde{\pi}(\boldsymbol{\theta},\mathbf{p})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p} = \mathbb{E}_{\tilde{\pi}}[fw], \end{aligned} \tag{3.2}$$

where $\tilde{\pi}(\boldsymbol{\theta},\mathbf{p})$ is the importance distribution and $w(\boldsymbol{\theta},\mathbf{p})$ the importance weight function. Since distributions $\pi$ and $\tilde{\pi}$ are known up to a normalizing constant, we may estimate this integral by following expressions (1.5)–(1.6) for importance samplers as

$$\hat{I} = \frac{\sum_{n=1}^{N} f(\boldsymbol{\theta}^n)w_n}{\sum_{n=1}^{N} w_n}, \quad w_n = \exp\left(\tilde{H}(\boldsymbol{\theta}^n,\mathbf{p}^n) - H(\boldsymbol{\theta}^n,\mathbf{p}^n)\right), \tag{3.3}$$

where $(\boldsymbol{\theta}^n,\mathbf{p}^n)$ are draws from $\tilde{\pi}$, and $w_n$ are the corresponding weights.

If a step size is chosen such that the modified Hamiltonian is a close approximation of the true Hamiltonian, backward error analysis is still valid. In particular, the difference between the true and modified Hamiltonian (2.12) implies that the reduction in efficiency of the estimator (3.3), introduced due to importance sampling, is minor in the case of the MMHMC method.

The main algorithmic differences between the Hamiltonian Monte Carlo (HMC) and MMHMC methods are listed in Table 3.1.

| | HMC | MMHMC |
|---|---|---|
| Momentum update | complete | partial |
| Momentum Metropolis test | ✗ | ✓ |
| Metropolis test | $H$ | $\tilde{H}$ |
| Re-weighting | ✗ | ✓ |

Table 3.1: Differences between HMC and MMHMC.

In the following sections, we proceed with the details on each modification introduced over the original GSHMC method. Due to the statistical framework in which MMHMC is formulated, we set the inverse temperature $\beta = 1$. We also refer to the Hamiltonian Dynamics step instead of calling it the Molecular Dynamics step and we define the parameter $\varphi \in (0,1]$ for the partial momentum update instead of $\phi \in (0,\pi/2]$, where $\varphi = \sin^2(\phi)$.

### 3.2.1 Modified Hamiltonians

The original GSHMC method has been formulated and implemented using the leapfrog integrator and the corresponding modified Hamiltonians. Our intention is to combine MMHMC with the numerical integrators which potentially can offer better conservation properties than Verlet. More specifically, we are interested in numerical integrators belonging to two-, three- and four-stage families of methods (2.21)–(2.23). For that, the formulation and

implementation of appropriate modified Hamiltonians are required. One procedure to calculate modified Hamiltonians of orders up to 24 is provided by Skeel and Hardy (2001) and Engle et al. (2005) for the Verlet integrator and it is further improved using Richardson extrapolation by Moan and Niesen (2014). This approach could be generalized to multi-stage integrators. Nevertheless, it requires a modification of the integrator by introducing an additional scalar variable into dynamics. We opt for a different strategy in deriving appropriate expressions for modified Hamiltonians depending on one, two and three parameters for two-, three- and four-stage methods, respectively.

We consider splitting methods and start with writing the expansion of the Hamiltonian function with a quadratic kinetic function, in terms of Poisson brackets of partial Hamiltonians (2.8)

$$\begin{aligned} \tilde{H} = H &+ h^2\alpha\{A,A,B\} + h^2\beta\{B,B,A\} \\ &+ h^4\gamma_1\{A,A,A,A,B\} + h^4\gamma_2\{B,A,A,A,B\} \\ &+ h^4\gamma_3\{B,B,A,A,B\} + h^4\gamma_4\{A,A,B,B,A\} + \mathcal{O}(h^6) \end{aligned} \tag{3.4}$$

where $\alpha, \beta, \gamma_{1-4}$ are polynomials written in terms of the integrators' coefficients $a_i, b_i$ (Blanes et al., 2014). Iterated Poisson brackets $\{F,\{G,H\}\}$ are denoted as $\{F,G,H\}$.

The expressions for a modified Hamiltonian of an arbitrary order can be obtained by directly applying the BCH formula to the exponentials of Lie derivatives $\mathcal{L}_A$ and $\mathcal{L}_B$ iteratively, but the computation is cumbersome except for a low order approximation (Sanz-Serna and Calvo, 1994). Alternatively, coefficients multiplying Poisson brackets for the 4th, 6th and 8th order modified Hamiltonians for symmetric composition methods can be derived from expressions given by Omelyan et al. (2002). In the case of general non-symmetric composition methods with an arbitrary number of stages, one can obtain the coefficients $\alpha$ and $\beta$ using results derived in (Hairer et al., 2006, see Lemma III.5.5).

Here we propose two alternative ways to derive the expression for the 4th and 6th order modified Hamiltonians. One uses analytical derivatives of the potential function whereas another one relies on numerical time derivatives of its gradient, obtained through the quantities available from a simulation.

#### 3.2.1.1 Analytical derivatives

For problems in which derivatives of the potential functions are available, we derive the 4th and 6th order modified Hamiltonians by first expanding terms from (3.4) using the definition (2.6) of Poisson brackets as

$$\begin{aligned} \{A,A,B\} &= \mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} \\ \{B,B,A\} &= U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \\ \{A,A,A,A,B\} &= U_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} M^{-1}\mathbf{p} M^{-1}\mathbf{p} M^{-1}\mathbf{p} \\ \{B,A,A,A,B\} &= -3U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} M^{-1}\mathbf{p} \\ \{B,B,A,A,B\} &= 2U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \end{aligned}$$

$$\begin{aligned}\{A,A,B,B,A\} &= 2U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} M^{-1}\mathbf{p} \\ &\quad +2\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}.\end{aligned}$$

This leads to the following 4th and 6th order modified Hamiltonians for splitting integrators

$$\tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) = H(\boldsymbol{\theta},\mathbf{p}) + h^2 c_{21}\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} + h^2 c_{22} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \tag{3.5}$$

$$\begin{aligned}\tilde{H}^{[6]}(\boldsymbol{\theta},\mathbf{p}) =& \tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) + h^4 c_{41} U_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} M^{-1}\mathbf{p} M^{-1}\mathbf{p} M^{-1}\mathbf{p} \\ &+ h^4 c_{42} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} M^{-1}\mathbf{p} \\ &+ h^4 c_{43} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \\ &+ h^4 c_{44}\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p},\end{aligned} \tag{3.6}$$

where

$$c_{21} = \alpha, \quad c_{22} = \beta, \quad c_{41} = \gamma_1, \quad c_{42} = 2\gamma_4 - 3\gamma_2, \quad c_{43} = 2\gamma_3, \quad c_{44} = 2\gamma_4. \tag{3.7}$$

Coefficients $\alpha, \beta, \gamma_{1-4}$ can be derived from expressions in terms of Poisson brackets, given by Omelyan et al. (2002) where the authors analyzed the so-called force-gradient integrators for molecular dynamics. In particular, they considered the splitting integrators that are extended by an additional higher-order operator into the single-exponential propagations.

If the potential function is quadratic, i.e. corresponding to problems of sampling from Gaussian distributions, the 6th order modified Hamiltonian (3.6) simplifies to

$$\begin{aligned}\tilde{H}^{[6]}(\boldsymbol{\theta},\mathbf{p}) = \tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) &+ h^4 c_{43} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \\ &+ h^4 c_{44}\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}.\end{aligned} \tag{3.8}$$

Combining (3.7) with expressions for $\alpha, \beta, \gamma_{1-4}$ we obtain the following coefficients for the two-stage integrator family (2.21)

$$\begin{aligned}c_{21} &= \frac{1}{24}\Big(6b - 1\Big) \\ c_{22} &= \frac{1}{12}\Big(6b^2 - 6b + 1\Big) \\ c_{41} &= \frac{1}{5760}\Big(7 - 30b\Big) \\ c_{42} &= \frac{1}{240}\Big(-10b^2 + 15b - 3\Big) \\ c_{43} &= \frac{1}{120}\Big(-30b^3 + 35b^2 - 15b + 2\Big) \\ c_{44} &= \frac{1}{240}(20b^2 - 1).\end{aligned} \tag{3.9}$$

For three-stage integrators (2.22) (a two-parameter family) we get

$$\begin{aligned}
c_{21} =& \frac{1}{12}\Big(1 - 6a(1-a)(1-2b)\Big)\\
c_{22} =& \frac{1}{24}\Big(6a(1-2b)^2 - 1\Big)\\
c_{41} =& \frac{1}{720}\Big(1 + 2(a-1)a(8 + 31(a-1)a)(1-2b) - 4b\Big)\\
c_{42} =& \frac{1}{240}\Big(6a^3(1-2b)^2 - a^2(19 - 116b + 36b^2 + 240b^3) + a(27 - 208b + 308b^2) - 48b^2 + 48b - 7\Big)\\
c_{43} =& \frac{1}{180}\Big(1 + 15a(1-2b)(-1 + 2a(2 - 3b + a(4b-2)))\Big)\\
c_{44} =& \frac{1}{240}\Big( -1 + 20a(1-2b)(b + a(1 + 6(b-1)b))\Big).
\end{aligned} \tag{3.10}$$

Finally, for four-stage integrators (2.23) (a three-parameter family) the coefficients read as

$$\begin{aligned}
c_{21} =& \frac{1}{12}\Big(6b_1^2 - 6b_1 + 1 + 6b_2(1-2a)(2b_1 + b_2 - 1)\Big)\\
c_{22} =& \frac{1}{24}\Big(6(b_1 + b_2(1-2a)^2) - 1\Big)\\
c_{41} =& \frac{1}{5760}\Big(7 + 60(8(a-1)^2a^2 - 1)b_1\Big)\\
c_{42} =& \frac{1}{96}\Big(1 - 12b_1 + 40b_1^2 - 24b_1^3 + 4(1-2a)(a - 3 + (20 - 6a)b_1 + 6(3 + 2a)b_1^2)b_2 + 8(1-2a)(5 +\\
& 9a^2 + 6a(b_1 - 2) - 9b_1)b_2^2 - 24(1-2a)^2b_2^3\Big)\\
c_{43} =& \frac{1}{360}\Big(2 - 15b_1 + 30b_1^2 + 15(1-2a)^2(4(1+a)b_1 - 1 - 2a)b_2 + 30(1-2a)^3b_2^2\Big)\\
c_{44} =& \frac{1}{120}\Big(2 - 30b_1^3 + 5b_1^2(7 - 6(4a(1+a) - 3)b_2) + 5(1-2a)b_2((7 - 6b_2)b_2 - 3 +\\
& 2a(6b_2^2 - 1 - 3b_2)) + 5b_1(2(1-2a)b_2(7 - 9b_2 + 6a(1 + b_2)) - 3)\Big).
\end{aligned} \tag{3.11}$$

Using (3.9) one can also obtain the modified Hamiltonian for the Verlet integrator since two steps of Verlet integration are equivalent to one step of the two-stage integrator with $b = 1/4$. The coefficients are therefore

$$\begin{aligned}
c_{21} &= \frac{1}{12}, \quad c_{22} = -\frac{1}{24}\\
c_{41} &= -\frac{1}{720}, \quad c_{42} = \frac{1}{120}, \quad c_{43} = -\frac{1}{240}, \quad c_{44} = \frac{1}{60}.
\end{aligned} \tag{3.12}$$

Figure 3.2 shows computational overheads of MMHMC, using the 4th order modified Hamiltonian (3.5), compared to the HMC method. The left-hand graph presents the overhead for a model with a tridiagonal Hessian matrix and indicates that for two different dimensions of the system the overhead becomes negligible as the number of integration steps increases. In contrast, for models with a dense Hessian matrix computation of modified Hamiltonians may introduce a significant additional cost, as shown in the right-hand graph.

For a 100-dimensional Gaussian problem, we also compare the resulting numerical integration error $\Delta$ observed in the true Hamiltonian $H$ and the 4th and 6th order modified

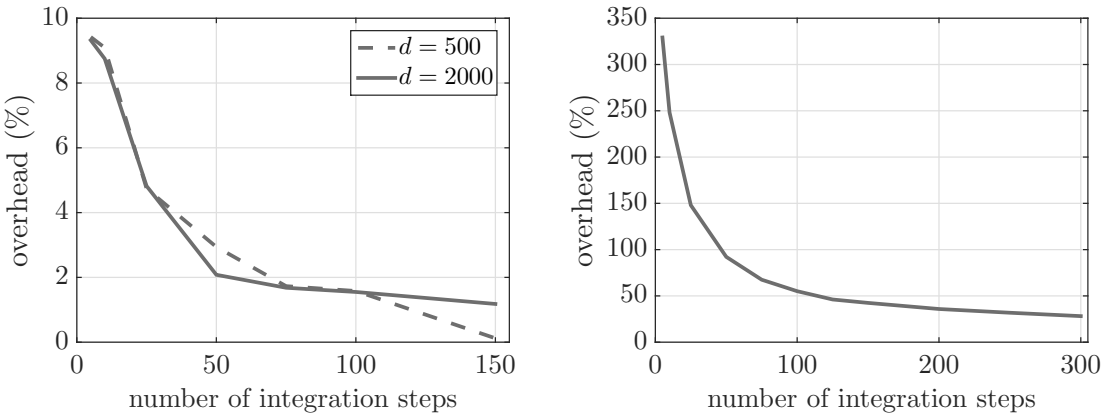


Figure 3.2: Computational overhead of MMHMC compared to HMC for models with a tridiagonal (left) and a dense Hessian matrix (right) using the 4th order modified Hamiltonian (3.5) where all derivatives are calculated analytically.

Hamiltonians given by (3.5) and (3.8), respectively (see Figure 3.3). $\tilde{H}^{[4]}$ is significantly better conserved than $H$. $\tilde{H}^{[6]}$ is even better conserved, as expected. Nevertheless, in practice, this must be weighted up against the computational cost of the calculation of the modified Hamiltonian (3.6) for non-Gaussian problems, which includes higher order derivatives.

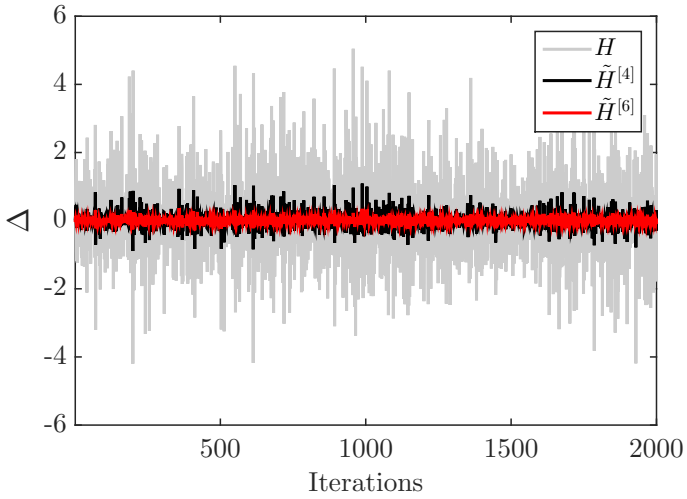


Figure 3.3: Error in Hamiltonians after numerical integration for a 100-dimensional Gaussian problem.

### 3.2.1.2 Numerical derivatives

For applications with a dense Hessian matrix (and higher derivatives), the computational overhead from calculations of modified Hamiltonians reduces the advantages of the MMHMC method. In order to implement such calculations in an efficient manner, we wish to express modified Hamiltonians in terms of quantities that are available during the simulation. Instead of making use of the time derivatives of the position vectors, as carried out in the

original GSHMC method, we employ identities for time derivatives of the gradient of the potential function, as follows,

$$\begin{aligned} U_{\boldsymbol{\theta}}^{(1)} &= U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta})M^{-1}\mathbf{p} \\ U_{\boldsymbol{\theta}}^{(2)} &= U_{\boldsymbol{\theta\theta\theta}}(\boldsymbol{\theta})M^{-1}\mathbf{p}M^{-1}\mathbf{p} - U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta})M^{-1}U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \\ U_{\boldsymbol{\theta}}^{(3)} &= U_{\boldsymbol{\theta\theta\theta\theta}}(\boldsymbol{\theta})M^{-1}\mathbf{p}M^{-1}\mathbf{p}M^{-1}\mathbf{p} - 3U_{\boldsymbol{\theta\theta\theta}}(\boldsymbol{\theta})M^{-1}U_{\boldsymbol{\theta}}(\boldsymbol{\theta})M^{-1}\mathbf{p} \\ &\quad -U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta})M^{-1}U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta})M^{-1}\mathbf{p}. \end{aligned} \tag{3.13}$$

Substituting these time derivatives (3.13) into the analytical expressions (3.5)–(3.6) for the 4th and 6th order modified Hamiltonians, we obtain

$$\tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) = H(\boldsymbol{\theta},\mathbf{p}) + h^2 k_{21}\mathbf{p}^T M^{-1}U_{\boldsymbol{\theta}}{}^{(1)} + h^2 k_{22}U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1}U_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \tag{3.14}$$

$$\begin{aligned} \tilde{H}^{[6]}(\boldsymbol{\theta},\mathbf{p}) &= \tilde{H}^{[4]}(\boldsymbol{\theta},\mathbf{p}) + h^4 k_{41}\mathbf{p}^T M^{-1}U_{\boldsymbol{\theta}}{}^{(3)} + h^4 k_{42}U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1}U_{\boldsymbol{\theta}}{}^{(2)} \\ &+ h^4 k_{43}U_{\boldsymbol{\theta}}{}^{(1)^T} M^{-1}U_{\boldsymbol{\theta}}{}^{(1)} + h^4 k_{44}U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1}U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta})M^{-1}U_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \end{aligned} \tag{3.15}$$

where the coefficients are

$$\begin{aligned} k_{21} &= c_{21}, \quad k_{22} = c_{22}, \\ k_{41} &= c_{41}, \quad k_{42} = 3c_{41} + c_{42}, \quad k_{43} = c_{41} + c_{44}, \quad k_{44} = 3c_{41} + c_{42} + c_{43}. \end{aligned} \tag{3.16}$$

We note that the newly derived expression (3.14) does not include the Hessian of the potential function and thus, allows for computation of $\tilde{H}^{[4]}$ using quantities available from a simulation. Nevertheless, this is not the case for the resulting 6th order Hamiltonians. The last term in (3.15), arising from an expansion of the Poisson bracket $\{B,B,A,A,B\}$, cannot be computed using time derivatives of available quantities and requires explicit calculation of the Hessian matrix of the potential function. Only for the Verlet integrator does this term vanish and the resulting coefficients are

$$\begin{aligned} k_{21} &= \frac{1}{12}, \quad k_{22} = -\frac{1}{24}, \\ k_{41} &= -\frac{1}{720}, \quad k_{42} = \frac{1}{240}, \quad k_{43} = \frac{11}{720}, \quad k_{44} = 0. \end{aligned} \tag{3.17}$$

We can now write explicit expressions for coefficients $k_{ij}$ by simply substituting the derived coefficients $c_{ij}$ (3.9), (3.10) or (3.11) into the relationship (3.16) for two-, three- or four-stage integrators, respectively. For two-stage integrator family (2.21) we obtain the

following coefficients

$$\begin{aligned}
k_{21} =& \frac{1}{24}\Big(6b - 1\Big)\\
k_{22} =& \frac{1}{12}\Big(6b^2 - 6b + 1\Big)\\
k_{41} =& \frac{1}{5760}\Big(7 - 30b\Big)\\
k_{42} =& \frac{1}{1920}\Big( - 80b^2 + 90b - 17\Big)\\
k_{43} =& \frac{1}{5760}\Big(480b^2 - 30b - 17\Big)\\
k_{44} =& \frac{1}{128}(1 - 4b)^2(1 - 2b).
\end{aligned} \tag{3.18}$$

For three-stage integrators (2.22) they are

$$\begin{aligned}
k_{21} =& \frac{1}{12}\Big(1 - 6a(1 - a)(1 - 2b)\Big)\\
k_{22} =& \frac{1}{24}\Big(6a(1 - 2b)^2 - 1\Big)\\
k_{41} =& \frac{1}{720}\Big(1 + 2(a - 1)a(8 + 31(a - 1)a)(1 - 2b) - 4b\Big)\\
k_{42} =& \frac{1}{240}\Big(2a(16 - a(34 - 31a^2)) - 4(a(45 + a(31a(2 + a) - 112)) - 7)b +\\
& 8(a(29 + a(62a - 78)) - 4)b^2 - 5\Big)\\
k_{43} =& \frac{1}{720}\Big(2a(19 - a^2(56 - 31a)) + 4(23 + a(a(99 + (50 - 31a)a) - 106))b +\\
& 24(a(29 + a(2a - 33)) - 4)b^2 - 11\Big)\\
k_{44} =& \frac{1}{120}(1 - 2b)(8b + a(16 + 31a^3 - 48b - 124a^2b + 12a(b(8 + 5b) - 2)) - 3).
\end{aligned} \tag{3.19}$$

Coefficients for four-stage integrators (2.23) are

$$\begin{aligned}
k_{21} =& \frac{1}{12}\Big(6b_1^2 - 6b_1 + 1 + 6b_2(1 - 2a)(2b_1 + b_2 - 1)\Big)\\
k_{22} =& \frac{1}{24}\Big(6(b_1 + b_2(1 - 2a)^2) - 1\Big)\\
k_{41} =& \frac{1}{5760}\Big(7 + 60(8(a - 1)^2a^2 - 1)b_1\Big)\\
k_{42} =& \frac{1}{1920}\Big( - 17 - 80b_1^2 + 20b_1(3 + 8a(3(a - 1)a(a - 4b_2 - 1) - b_2) - 8b_2) +\\
& 40(2a - 1)b_2(4a(3a - 1) + 2b_2 - 3)\Big)\\
k_{43} =& \frac{1}{5760}\Big( - 17 + 60(8b_1^2 - 8(2a - 1)b_2(a + b_2 + 6(a - 1)ab_2) + b_1(16b_2 + 8a((a - 1)^2a +\\
& 2(6a - 5)b_2) - 1))\Big)\\
k_{44} =& \frac{1}{128}\Big(1 - 32b_1^3 - 32b_1^2((4a(1 + a) - 3)b_2 - 1) + 8(2a - 1)b_2(4a^2 + (1 - 2b_2)^2 +\\
& 4a(1 - 2b_2)b_2) + 4b_1(8(a - 1)^2a^2 + 16b_2 - 8a(3 + 4a^2)b_2 - 8(3 + 4(a - 2)a)b_2^2 - 3)\Big).
\end{aligned} \tag{3.20}$$

In the original GSHMC method, an interpolating polynomial of positions $\boldsymbol{\Theta}(t_i) = \boldsymbol{\theta}^i,\ i = n-k, \dots, n, \dots, n+k,\ n \in \{0, L\}$ is constructed from a numerical trajectory $\{\boldsymbol{\theta}^i\}_{i=-k}^{L+k}$, where $k = 2$ and $k = 3$ for the 4th and 6th order modified Hamiltonian, respectively. This requires

four or six additional gradient calculations in order to compute $\tilde{H}^{[4]}$ or $\tilde{H}^{[6]}$, respectively. We choose a different strategy and calculate the polynomial in terms of the gradient of the potential function

$$\mathbf{U}(t_i) = U_{\boldsymbol{\theta}}(\boldsymbol{\theta}^i),\ i = n-k, \ldots, n, \ldots, n+k.$$

With this approach $k = 1$ for the 4th order and $k = 2$ for the 6th order modified Hamiltonian, meaning that an evaluation of $\tilde{H}^{[4]}$ or $\tilde{H}^{[6]}$ requires two or four additional gradient calculations, respectively. Note that $k$ corresponds to a multiple of the full integration step only in the case of the Verlet integrator; for others, it is the number of stages performed (e.g. $k = 2$ corresponds to a half integration step of a four-stage method). Also, note that an efficient implementation does not include the unnecessary integration sub-step of momentum update at the very beginning and very end of the numerical trajectory $\{U_{\boldsymbol{\theta}}(\boldsymbol{\theta}^i)\}_{i=-k}^{L+k}$.

Time derivatives of the gradient of the potential function are approximated using central finite difference of second order of accuracy for the 4th order modified Hamiltonian

$$U_{\boldsymbol{\theta}}^{(1)} \approx \frac{\mathbf{U}(t_{n+1}) - \mathbf{U}(t_{n-1})}{2\varepsilon} =: \mathbf{U}^{(1)}, \tag{3.21}$$

where $\varepsilon = h$ for the Verlet, $\varepsilon = h/2$ for two-stage and $\varepsilon = ah$ for three- and four-stage integrators, $h$ being the integration step size and $a$ being the integrator's coefficient advancing position variables. The 6th order modified Hamiltonian, here considered only for the Verlet and two-stage integrators, is calculated using centered differences of fourth order accuracy for the first derivative and second order accuracy for the second and third derivatives

$$\begin{aligned}
U_{\boldsymbol{\theta}}^{(1)} &\approx \frac{\mathbf{U}(t_{n-2}) - 8\mathbf{U}(t_{n-1}) + 8\mathbf{U}(t_{n+1}) - \mathbf{U}(t_{n+2})}{12\varepsilon} =: \mathbf{U}^{(1)} \\
U_{\boldsymbol{\theta}}^{(2)} &\approx \frac{\mathbf{U}(t_{n-1}) - 2\mathbf{U}(t_n) + \mathbf{U}(t_{n+1})}{\varepsilon^2} =: \mathbf{U}^{(2)} \\
U_{\boldsymbol{\theta}}^{(3)} &\approx \frac{-\mathbf{U}(t_{n-2}) + 2\mathbf{U}(t_{n-1}) - 2\mathbf{U}(t_{n+1}) + \mathbf{U}(t_{n+2})}{2\varepsilon^3} =: \mathbf{U}^{(3)},
\end{aligned}$$

where $\varepsilon$ depends on the integrator as before. Different orders of accuracy are necessary in order to achieve the overall required accuracy of the modified Hamiltonian.

The final expressions for our newly derived modified Hamiltonians are

$$\tilde{H}^{[4]}(\boldsymbol{\theta}, \mathbf{p}) = H(\boldsymbol{\theta}, \mathbf{p}) + hk_{21}\mathbf{p}^T M^{-1} P_1 + h^2 k_{22} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \tag{3.22}$$

$$\begin{aligned}
\tilde{H}^{[6]}(\boldsymbol{\theta}, \mathbf{p}) &= \tilde{H}^{[4]}(\boldsymbol{\theta}, \mathbf{p}) + hk_{41}\mathbf{p}^T M^{-1} P_3 + h^2 k_{42} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} P_2 \\
&+ h^2 k_{43} P_1^T M^{-1} P_1 + h^4 k_{44} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}),
\end{aligned} \tag{3.23}$$

where $P_i = \mathbf{U}^{(i)} \cdot h^i$. We note that the term with the coefficient $k_{22}$ is calculated exactly, i.e. avoiding finite difference approximation, which therefore improves the approximation of the modified Hamiltonian compared to the original strategy used in GSHMC. We also note that compared to the expressions with analytical derivatives (3.5) and (3.6) with coefficients $c_{ij}$ multiplying exact derivatives, in the formulations (3.22) and (3.23) for

the 4th and 6th order Hamiltonians, respectively, the terms arising from those multiplying $c_{21}, c_{41}, c_{42}$, and $c_{44}$ are approximated with $P_i$. The level of accuracy provided by the modified Hamiltonians (3.22) and (3.23), however, are not affected by these approximations.

The computational overhead of MMHMC compared to the HMC method is shown in Figure 3.4 for models with a tridiagonal (left-hand graph) and a dense Hessian matrix (right-hand graph) using the modified Hamiltonians (3.22) and (3.23) of 4th and 6th order, respectively, with numerical approximations of derivatives. Compared to Figure 3.2, where all derivatives are calculated analytically, we note that for models with a sparse Hessian (left-hand graphs), the 4th order modified Hamiltonian (3.5) with analytical derivatives introduces less computational overhead than (3.22) with a numerical approximation. This is due to additional forward and backward integration steps, which do not counterbalance the inexpensive Hessian calculation. For models with a dense Hessian matrix (right-hand graphs), we recommend always using (3.22), which significantly reduces the overhead. The 6th order modified Hamiltonian (3.23) clearly requires additional computational effort, due to two extra gradient calculations per MC iteration. In the following sections, we show that using the order of modified Hamiltonian higher than four can be avoided by introducing accurate multi-stage integrators specially tuned for MMHMC.

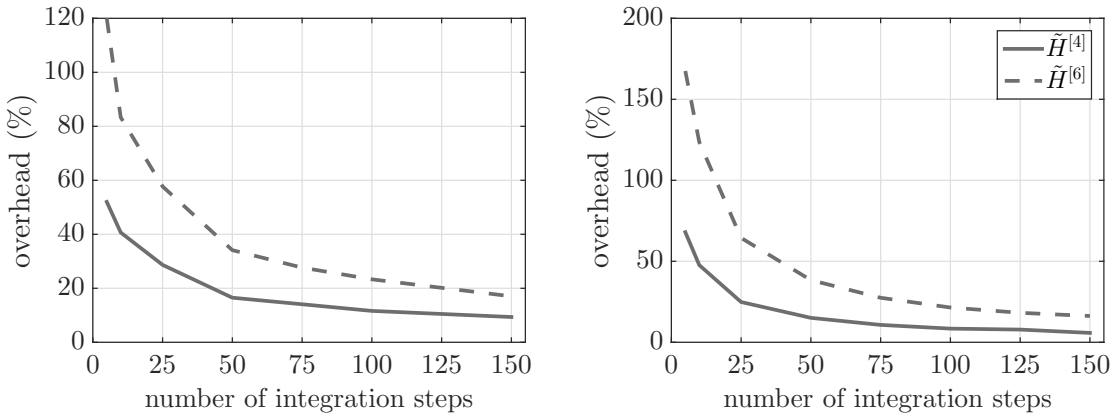


Figure 3.4: Computational overhead of MMHMC compared to HMC for models with a tridiagonal (left) and a dense (right) Hessian matrix, using 4th and 6th order modified Hamiltonians with numerical approximation of the time derivatives.

In summary, we provided two alternative formulations of the 4th and 6th order modified Hamiltonians corresponding to multi-stage integrators (2.21)–(2.23) with arbitrary coefficients. For the cases when analytical derivatives of the potential function are available and inexpensive to compute, the modified Hamiltonians can be calculated using (3.5)–(3.12). For problems in which this is not the case, we provided formulations of modified Hamiltonians which mainly rely on quantities available from the simulation. Both approaches can be used with any multi-stage integrator (2.21)–(2.23) including the Verlet integrator.

In the following section, we devise the novel numerical integrators specifically for sampling with modified Hamiltonians and examine their performance in comparison with already proposed integrators for HMC methods.

### 3.2.2 Integrators

Until now, the Verlet/leapfrog integrator has been the integrator of choice for the GSHMC method. The modified Hamiltonian of order four is explicitly formulated, and a general formula for modified Hamiltonians of an arbitrary order of accuracy has been obtained using Lagrangian formalism by Akhmatskaya and Reich (2008). In this section, we consider alternative integrators and investigate their competitiveness with the Verlet integrator. Explicit expressions for the corresponding modified Hamiltonians of order four and six were derived in Section 3.2.1.

#### 3.2.2.1 Multi-stage integrators

Our focus now shifts to multi-stage integrators. There are two reasons for our interest in these integrators. One is their potentially-higher-than-in-Verlet accuracy at the same computational cost. This implies higher acceptance rate and longer step sizes, thus an improved sampling performance. Another possible benefit from the integrators of this class is avoiding the need for computationally expensive higher order modified Hamiltonians due to the accurate integration.

In Section 2.2.3 we reviewed multi-stage integrators designed for molecular dynamics and HMC simulations. In this section, we derive the new multi-stage integrators for sampling with modified Hamiltonians and investigate whether the integrators previously proposed for HMC (Blanes et al., 2014) and the newly derived integrators can improve the performance of MMHMC compared to the Verlet integrator. We now proceed with the derivation of multi-stage integrators specific to sampling with modified Hamiltonians.

In the MMHMC method, the underlying system is driven by Hamiltonian dynamics (2.3). The equations of motion are therefore the same as in the HMC method; however, MMHMC includes the different Metropolis test whose success depends on the accuracy of an integrator. Indeed, the sampling performance of MMHMC is controlled not by an energy error as in HMC but by a modified energy error. Thus, inspired by the ideas of McLachlan (1995) and Blanes et al. (2014) for improving HMC performance by minimizing energy error / expected energy error through the appropriate choice of parameters of an integrator, we design the new integrators by considering the (expected) error in the modified Hamiltonian $\tilde{H}^{[l]}$ of order $l$, in order to enhance performance of MMHMC. The expected values of such errors are taken with respect to the modified density $\tilde{\pi}$, instead of the true density $\pi$.

We choose integrating parameters through minimization of either Hamiltonian error introduced after integration

$$\Delta = \tilde{H}^{[l]}(\Psi_{h,L}(\boldsymbol{\theta}, \mathbf{p})) - \tilde{H}^{[l]}(\boldsymbol{\theta}, \mathbf{p}), \tag{3.24}$$

or its expected value $\mathbb{E}_{\tilde{\pi}}(\Delta)$. Recall that $\Psi_{h,L}(\boldsymbol{\theta}, \mathbf{p})$ is the exact $hL$-time map of the modified Hamiltonian $\tilde{H}$. With this approach, we design the minimum error and minimum expected error integrators for sampling with modified (M) Hamiltonians. In order to distinguish

these integrators from the corresponding ones designed for the HMC method, we denote them as M-ME and M-BCSS, respectively.

**Minimum error (M-ME) integrators**

We wish to construct the minimum error integrators for the 4th order modified Hamiltonian.

The Taylor expansion of the 4th order modified Hamiltonian after one integration step with the method $\Psi_h$ can be written as (Sanz-Serna and Calvo, 1994)

$$\begin{aligned}\tilde{H}^{[4]}(\boldsymbol{\theta}', \mathbf{p}') &= \tilde{H}^{[4]}(\Psi_h(\boldsymbol{\theta}, \mathbf{p})) = \exp\left(h\mathcal{L}_{\tilde{H}}\right)\tilde{H}^{[4]}(\boldsymbol{\theta}, \mathbf{p}) \\ &= \tilde{H}^{[4]}(\boldsymbol{\theta}, \mathbf{p}) + h\mathcal{L}_{\tilde{H}}\tilde{H}^{[4]}(\boldsymbol{\theta}, \mathbf{p}) + \frac{1}{2}h^2\mathcal{L}^2_{\tilde{H}}\tilde{H}^{[4]}(\boldsymbol{\theta}, \mathbf{p}) + \dots,\end{aligned}$$

where $\tilde{H}$ is the modified Hamiltonian (3.4) expressed in terms of Poisson brackets. Recalling the definition of the Lie derivative, $\mathcal{L}_F(\cdot) = \{\cdot, F\}$, the error $\Delta$ in $\tilde{H}^{[4]}$ after one integration step reads

$$\begin{aligned}\Delta(\boldsymbol{\theta}, \mathbf{p}) = h^5\big(&\gamma_1\{A,A,A,A,A,B\}(\boldsymbol{\theta}, \mathbf{p}) + \gamma_1\{B,A,A,A,A,B\}(\boldsymbol{\theta}, \mathbf{p}) \\ &+\gamma_2\{A,B,A,A,A,B\}(\boldsymbol{\theta}, \mathbf{p}) + \gamma_2\{B,B,A,A,A,B\}(\boldsymbol{\theta}, \mathbf{p}) \\ &+\gamma_3\{A,B,B,A,A,B\}(\boldsymbol{\theta}, \mathbf{p}) + \gamma_3\{B,B,B,A,A,B\}(\boldsymbol{\theta}, \mathbf{p}) \\ &+\gamma_4\{A,A,A,B,B,A\}(\boldsymbol{\theta}, \mathbf{p}) + \gamma_4\{B,A,A,B,B,A\}(\boldsymbol{\theta}, \mathbf{p})\big).\end{aligned} \tag{3.25}$$

An error metric for the 4th order modified Hamiltonian can then be defined as a function of the integrating coefficients

$$E = \sqrt{\gamma_1^2 + \gamma_2^2 + \gamma_3^2 + \gamma_4^2}, \tag{3.26}$$

where the explicit expressions for $\gamma_{1-4}$ follow from relationship (3.7) as

$$\gamma_1 = c_{41}, \quad \gamma_2 = \frac{1}{3}(c_{44} - c_{42}), \quad \gamma_3 = \frac{1}{2}c_{43}, \quad \gamma_4 = \frac{1}{2}c_{44}$$

and the coefficients $c_{ij}$ are calculated from (3.9), (3.10) or (3.11) for two-, three- or four-stage integrators, respectively. For quadratic potential and kinetic functions, corresponding to the problem of sampling from a Gaussian distribution, error (3.25) simplifies and we can define the error metric as

$$E^G = |\gamma_2 + \gamma_4|. \tag{3.27}$$

In contrast to this approach, the error metric for the minimum error integrator derived for sampling with the true Hamiltonian, i.e. the HMC method, is defined through the Hamiltonian truncation error $H - \tilde{H}$ at the state $(\boldsymbol{\theta}, \mathbf{p})$ (McLachlan, 1995), rather than the error in Hamiltonian after numerical integration. Minimization of the error metric

$$E_{HMC} = \alpha^2 + \beta^2$$

results in the coefficient $b = 0.193183$ for the two-stage integrator.

In order to obtain numerical values for integrating coefficients for the MMHMC method, we minimized the metrics $E$ and $E^G$ on the interval $(0, 0.5)$ using *Mathematica*. In Table 3.2 we summarize the coefficients obtained for each integrator with the corresponding error metrics for multi-stage minimum error integrators. The smallest error metric is achieved using three-stage integrators.

| Integrator | Coefficients | $E$ | Coefficients | $E^G$ |
|---|---|---|---|---|
| 2-stage | $b = 0.23061$ | $2.720 \cdot 10^{-4}$ | $b = 0.230907$ | $1.444 \cdot 10^{-11}$ |
| 3-stage | $a = 0.355423$<br>$b = 0.184569$ | $7.391 \cdot 10^{-5}$ | $a = 0.39263$<br>$b = 0.199778$ | $2.304 \cdot 10^{-19}$ |
| 4-stage | $a = 0.0840641$<br>$b_1 = 0.0602952$<br>$b_2 = 0.216673$ | $7.782 \cdot 10^{-4}$ | $a = 0.441252$<br>$b_1 = 0.266011$<br>$b_2 = 0.181055$ | $8.289 \cdot 10^{-12}$ |

TABLE 3.2: Coefficients for the novel multi-stage minimum error integrators derived for sampling with the 4th order modified Hamiltonian, with the corresponding error metric $E$ for general problems and $E^G$ for Gaussian problems.

Error metric definitions (3.26) and (3.27) are based on the assumption that the iterated brackets from the error (3.25) in $\tilde{H}^{[4]}$ contribute equally to the Hamiltonian error. This assumption does not hold in general, although it is a reasonable assumption to start with. Moreover, the weights of the brackets depend on the problem at hand, and their estimation could lead to problem specific integrators. Nevertheless, in this thesis, our aim is to obtain the integrators for use in a broad range of problems.

**Minimum expected error (M-BCSS) integrators**

The modified Hamiltonians we consider here are of order 4 and 6. We adopt a strategy similar to the one proposed by Blanes et al. (2014), namely to find the parameters of integrators that minimize the expected value of the error. In our case, the error (3.24), resulting from numerical integration is in terms of the modified Hamiltonian and the expected value is taken with respect to the modified density $\tilde{\pi}$.

As in the case when considering the error in the true Hamiltonian, we may prove that the expected error in the modified Hamiltonian $\mathbb{E}_{\tilde{\pi}}(\Delta)$ is also positive. Our objective is, therefore, to find a function $\rho(h, \boldsymbol{\xi})$ that bounds $\mathbb{E}_{\tilde{\pi}}(\Delta)$, i.e.

$$0 \leq \mathbb{E}_{\tilde{\pi}}(\Delta) \leq \rho(h, \boldsymbol{\xi}).$$

Here $\boldsymbol{\xi}$ is a parameter vector, e.g. $\boldsymbol{\xi} = \{\xi_1\}, \xi_1 = b$, for two-stage integrators. From now on we consider only univariate model problem, as suggested by Blanes et al. (2014), namely a univariate harmonic oscillator with equations of motion given in (2.25) and the corresponding Hamiltonian (2.26). This implies that the error $\Delta$ defined in (3.24) becomes

$$\Delta = \tilde{H}^{[l]}(\Psi_{h,L}(\theta, p)) - \tilde{H}^{[l]}(\theta, p). \tag{3.28}$$

We first find the numerical solution to the dynamics (2.25) for a single time step $(\theta_{n+1}, p_{n+1}) = \Psi_h(\theta_n, p_n)$. In matrix form this is given by

$$\begin{bmatrix}\theta_{n+1}\\ p_{n+1}\end{bmatrix} = \tilde{M}_h \begin{bmatrix}\theta_n\\ p_n\end{bmatrix}, \quad \tilde{M}_h = \begin{bmatrix}A_h & B_h\\ C_h & A_h\end{bmatrix},$$

where the coefficients $A_h, B_h, C_h$ depend on the integrator. After $L$ integration steps the state of the system $(\theta_L, p_L) = \Psi_{hL}(\theta, p)$ is given by

$$\begin{bmatrix}\theta_L\\ p_L\end{bmatrix} = \underbrace{\tilde{M}_h \dots \tilde{M}_h}_{L \text{ times}} \begin{bmatrix}\theta\\ p\end{bmatrix} = \tilde{M}_h^L \begin{bmatrix}\theta\\ p\end{bmatrix}. \tag{3.29}$$

For the Verlet integrator the matrix $\tilde{M}_h$ can be calculated as

$$\tilde{M}_h = B\left(\frac{1}{2}\right) \cdot A\left(1\right) \cdot B\left(\frac{1}{2}\right),$$

where

$$A(a) = \begin{bmatrix}1 & ah\\ 0 & 1\end{bmatrix}, \; B(b) = \begin{bmatrix}1 & 0\\ -bh & 1\end{bmatrix}$$

correspond to mappings $\varphi_h^A$ and $\varphi_h^B$, respectively. The resulting elements of $\tilde{M}_h$ are

$$\begin{aligned} A_h &= 1 - \frac{h^2}{2} \\ B_h &= h \\ C_h &= \frac{h^3}{4} - h. \end{aligned} \tag{3.30}$$

We derive the matrix $\tilde{M}_h$ for two-stage integrators, which follows as

$$\begin{aligned} \tilde{M}_h &= B\left(b\right) \cdot A\left(\frac{1}{2}\right) \cdot B\left(1-2b\right) \cdot A\left(\frac{1}{2}\right) \cdot B\left(b\right) \\ &= \begin{bmatrix}1 & 0\\ -bh & 1\end{bmatrix} \cdot \begin{bmatrix}1 & h/2\\ 0 & 1\end{bmatrix} \cdot \begin{bmatrix}1 & 0\\ -(1-2b)h & 1\end{bmatrix} \cdot \begin{bmatrix}1 & h/2\\ 0 & 1\end{bmatrix} \cdot \begin{bmatrix}1 & 0\\ -bh & 1\end{bmatrix} \\ &= \begin{bmatrix}A_h & B_h\\ C_h & A_h\end{bmatrix}, \end{aligned}$$

such that

$$\begin{aligned} A_h &= \frac{h^4}{4} b(1-2b) - \frac{h^2}{2} + 1 \\ B_h &= -\frac{h^3}{4}(1-2b) + h \\ C_h &= -\frac{h^5}{4} b^2(1-2b) + h^3 b(1-b) - h. \end{aligned} \tag{3.31}$$

Similarly, for three-stage integrators we compute

$$\tilde{M}_h = B(b) \cdot A(a) \cdot B(\frac{1}{2} - b) \cdot A(1-2a) \cdot B(\frac{1}{2} - b) \cdot A(a) \cdot B(b)$$

and obtain

$$\begin{aligned} A_h &= \frac{h^6}{4} a^2(2a-1)(1-2b)^2 b + \frac{h^4}{4} a\left(1 - 4b^2 - a(1-4b)\right) - \frac{h^2}{2} + 1 \\ B_h &= \frac{h^5}{4} a^2(1-2a)(1-2b)^2 - h^3 a(1-a)(1-2b) + h \\ C_h &= \frac{h^7}{4} a^2(1-2a)(1-2b)^2 b^2 + \frac{h^5}{2} a(2a(1-b)-1)b(1-2b) + \\ &\quad \frac{h^3}{4}\left(1 - 2a(1-2b)^2\right) - h. \end{aligned} \tag{3.32}$$

For four-stage integrators the computation becomes a bit more involved and results in coefficients

$$\begin{aligned} A_h &= \frac{h^8}{4}(1-2a)^2 a^2 b_1 b_2^2 (1 - 2b_1 - 2b_2) + \\ &\quad \frac{h^6}{4} a b_2 \left(2b_1(1-2a)(2b_1 - 1 + 3b_2 - 4ab_2) + b_2(2b_2 + a(4 - 5a - 8(1-a)b_2))\right) + \\ &\quad \frac{h^4}{4}\left(b_1(1 - 4b_2(1-2a)) + b_2(1 - 2b_2 + 4a(b_2 - a)) - 2b_1^2\right) - \frac{h^2}{2} + 1 \\ B_h &= -\frac{h^7}{4}(1-2a)^2 a^2 b_2^2 (1 - 2b_1 - 2b_2) + \frac{h^5 a}{2}(1-2a) b_2 (1 - 2b_1 + 2(a-1)b_2) + \\ &\quad \frac{h^3}{4}\left(2(1-2a)^2 b_2 - (1 - 2b_1)\right) + h \\ C_h &= \frac{h^9}{4} a b_1^2 b_2^2 (8b_1 + a(2b_1 - 1 + 4a(1 - a - 2b_1) + 2b_2 + 8(a-1)ab_2)) - \\ &\quad \frac{h^7}{2} a b_1 b_2 (1-2a)\left(2b_1^2 - (1-2a)b_2(1-2b_2) - b_1(1 + 2(3a-2)b_2)\right) + \\ &\quad \frac{h^5}{4}(2b_1^3 - (1-2a)^2 b_2^2 (1-2b_2) - b_1^2(1 + (8a(1+a) - 6)b_2) - \\ &\quad 2(1-2a) b_1 b_2 (1 - 3b_2 + 2a(1+b_2))) + \\ &\quad h^3\left((1-2a)(1-b_2)b_2 + b_1(1 - 2(1-2a)b_2) - b_1^2\right) - h. \end{aligned} \tag{3.33}$$

It is well known that if step size $h$ is such that $|A_h| < 1$ the integration is stable. In that case we may define

$$\begin{aligned} \zeta_h &:= \arccos A_h \\ \chi_h &:= B_h / \sin \zeta_h, \end{aligned}$$

for which the one-step and $L$-steps integration matrices $\tilde{M}_h$ and $\tilde{M}_h^L$, respectively, are

$$\tilde{M}_h = \begin{bmatrix} \cos(\zeta_h) & \chi_h \sin(\zeta_h) \\ -\chi_h^{-1} \sin(\zeta_h) & \cos(\zeta_h) \end{bmatrix}$$

and

$$\tilde{M}_h^L = \begin{bmatrix} \cos(L\zeta_h) & \chi_h \sin(L\zeta_h) \\ -\chi_h^{-1}\sin(L\zeta_h) & \cos(L\zeta_h) \end{bmatrix}. \tag{3.34}$$

We now proceed to the calculation of the univariate error (3.28) for the 6th order modified Hamiltonian, which for the univariate harmonic oscillator model problem has the form

$$\begin{aligned} \tilde{H}^{[6]}(\theta,p) &= \frac{1}{2}\theta^2 + \frac{1}{2}p^2 + h^2 c_{21}p^2 + h^2 c_{22}\theta^2 + h^4 c_{44}p^2 + h^4 c_{43}\theta^2 \\ &= \left(\frac{1}{2} + h^2 c_{22} + h^4 c_{43}\right)\theta^2 + \left(\frac{1}{2} + h^2 c_{21} + h^4 c_{44}\right)p^2, \end{aligned} \tag{3.35}$$

where coefficients $c_{ij}$ depend on the integrator's formulation and its coefficients, which we derived before in Section 3.2.1. The derivation for the 4th order modified Hamiltonian follows directly from setting $c_{43} = c_{44} = 0$.

In order to calculate the expected value of the error (3.28) we follow the calculations from the proof of Proposition 3 in (Blanes et al., 2014) and denote

$$\begin{aligned} c &= \cos(L\zeta_h) \\ s &= \sin(L\zeta_h) \\ S_1 &= 1 + 2h^2 c_{22} + 2h^4 c_{43} \\ S_2 &= 1 + 2h^2 c_{21} + 2h^4 c_{44} \end{aligned}$$

for a simplified notation. Substituting (3.35), (3.34) and (3.29) into (3.28) we obtain

$$\begin{aligned} 2\Delta &= S_1\left(c\theta + \chi_h sp\right)^2 + S_2\left(-\frac{1}{\chi_h}s\theta + cp\right)^2 - S_1\theta^2 - S_2p^2 \\ &= s^2\left(\frac{1}{\chi_h^2}S_2 - S_1\right)\theta^2 + s^2\left(\chi_h^2 S_1 - S_2\right)p^2 + 2sc\left(S_1\chi_h - S_2\frac{1}{\chi_h}\right)\theta p. \end{aligned}$$

Since the expectations are taken with respect to the modified density $\tilde{\pi}$,

$$\mathbb{E}_{\tilde{\pi}}(\theta^2) = \frac{1}{S_1},\ \mathbb{E}_{\tilde{\pi}}(p^2) = \frac{1}{S_2},\ \mathbb{E}_{\tilde{\pi}}(\theta p) = 0,$$

it follows that

$$2\mathbb{E}_{\tilde{\pi}}(\Delta) = s^2\Big(\frac{1}{\chi_h^2}\frac{S_2}{S_1} + \chi_h^2\frac{S_1}{S_2} - 2\Big).$$

We can simplify the equation by defining

$$\widetilde{\chi}_h^2 := \chi_h^2\frac{S_1}{S_2} = \chi_h^2 S$$

so that we obtain

$$\mathbb{E}_{\tilde{\pi}}(\Delta) = s^2\rho(h,\boldsymbol{\xi})$$

where

$$\rho(h, \boldsymbol{\xi}) = \frac{1}{2}\Big(\widetilde{\chi}_h - \frac{1}{\widetilde{\chi}_h}\Big)^2 = \frac{\Big(SB_h + C_h\Big)^2}{2S(1 - A_h^2)}. \tag{3.36}$$

For the 4th order modified Hamiltonian

$$S = \frac{1 + 2h^2 c_{22}}{1 + 2h^2 c_{21}}$$

and for the 6th order modified Hamiltonian

$$S = \frac{1 + 2h^2 c_{22} + 2h^4 c_{43}}{1 + 2h^2 c_{21} + 2h^4 c_{44}}.$$

The conditions for stable integration and positivity of $\rho(h, \boldsymbol{\xi})$ are that $|A_h| < 1$ and $S > 0$. For the two-stage integrators and the 4th order modified Hamiltonian this is equivalent to the following conditions

$$\begin{aligned} h &< \sqrt{12/(1-6b)} \quad \text{for } b < \frac{1}{6}, \\ h &> \sqrt{12/(1-6b)} \quad \text{for } b > \frac{1}{6}, \\ 0 < \quad h &< \min\big\{\sqrt{2/b}, \sqrt{1/(1-2b)}\big\}, \end{aligned}$$

which are always satisfied for $b \in (0, \frac{1}{2})$.

We note that we can recover the true Hamiltonian by setting coefficients $c_{ij}$ to zero. Doing so, we obtain exactly the same function as derived by Blanes et al. (2014)

$$\rho_{\text{HMC}}(h, \boldsymbol{\xi}) = \frac{(B_h + C_h)^2}{2(1 - A_h^2)}. \tag{3.37}$$

Finally, we choose coefficients $\boldsymbol{\xi}$ that minimize the function

$$\|\rho\|_{(\bar{h})} = \max_{0<h<\bar{h}} \rho(h, \boldsymbol{\xi}), \tag{3.38}$$

where $\bar{h}$ is equal to the number of stages in the integrator (see Section 2.2.3). For the family of two-stage integrators and 4th order modified Hamiltonian we have

$$\rho(h, b) = \frac{h^8\Big(b\big(12 + 4b(6b-5) + b(1 + 4b(3b-2))h^2\big) - 2\Big)^2}{4\big(2 - bh^2\big)\big(4 + (2b-1)h^2\big)\big(2 + b(2b-1)h^2\big)\big(12 + (6b-1)h^2\big)\big(6 + (1 + 6(b-1)b)h^2\big)}. \tag{3.39}$$

Minimizing the function $\|\rho\|_{(2)}$ we obtain the coefficient $b = 0.238016$ for the two-stage M-BCSS integrator derived for sampling with the MMHMC method. We note the difference in value for the coefficient of the original two-stage BCSS integrator, introduced for HMC, being $b = 0.21178$.

Minimization of $\|\rho\|_{(3)}$ and $\|\rho\|_{(4)}$ for the three- and four-stage integrators, respectively,

is more laborious than for two-stage integrators and we leave this derivation and performance comparison for future research.

The above analysis can be extended to the multivariate case. Thus, for a $D$-variate Gaussian distribution

$$\pi(\boldsymbol{\theta}) \propto \exp\left(-\frac{1}{2}\boldsymbol{\theta}^T\Sigma^{-1}\boldsymbol{\theta}\right),$$

we bound the expected error as

$$\mathbb{E}_{\tilde{\pi}}(\Delta) \leq \sum_{d=1}^{D} \rho(f_d h, \boldsymbol{\xi}),$$

where $f_d = 1/\sigma_d$ and $\sigma_d^2$ are the eigenvalues of the covariance matrix $\Sigma$ (Blanes et al., 2014). This model problem can be seen as $D$ coupled harmonic oscillators with angular frequencies $f_d$.

In Figure 3.5 we plot $\|\rho_{\text{HMC}}\|_{(\bar{h})}$ (3.37)–(3.38) as a function of the maximal step size $\bar{h}$ for the two-stage BCSS, ME, and Verlet integrators for the HMC method (dashed lines), and the corresponding function $\|\rho\|_{(\bar{h})}$ (3.38)–(3.39) for the two-stage M-BCSS, M-ME, and Verlet integrators, derived in this section for sampling with MMHMC (solid lines). The upper bound of the expected error in Hamiltonian, and thus the error of the method, is lower for integrators developed for MMHMC than in the case of the HMC specific integrators, which confirms a better conservation of modified Hamiltonians by symplectic integrators than true Hamiltonian. This is becoming more obvious when comparing $\|\rho_{\text{HMC}}\|_{(\bar{h})}$ and $\|\rho\|_{(\bar{h})}$ for the Verlet integrator. As follows from Figure 3.5 the two-stage integrators derived for HMC and MMHMC provide better accuracy than Verlet for step sizes less or equal to a half stability limit of Verlet, i.e. $\bar{h} = 2$. The integrators derived for MMHMC guarantee a better accuracy than other integrators for $\bar{h}$ even bigger than $2$ (Figure 3.5), which implies their efficiency for longer step sizes compared with Verlet and two-stage integrators for HMC. Please notice that $\bar{h}$ in Figure 3.5 refers to a step size for a two-stage integrator. If Verlet is viewed as a single stage integrator, this corresponds to $\bar{h} = 1$. It is important to note that the Verlet integrator has the highest stability limit among other two-stage integrators. Nevertheless, as Figure 3.5 suggests, the accuracy is degrading with $\bar{h}$ approaching the stability limit. It is the characteristics of the sampling problem (such as the number of parameters, the number of observations, the nature of the underlying model) that determine the optimal step size and therefore the integrator which would provide the best performance. A zoom-in of the left-hand graph, shown in the right-hand graph, gives a bit better insight into the functions' behavior for the MMHMC method.

We compare the performance of the standard Verlet integrator, the previously proposed two-stage BCSS and ME integrators, and the newly derived two-stage M-BCSS and M-ME and three-stage minimum error (M-ME$_3$) integrators, for sampling from a multivariate Gaussian distribution of dimension $D = 100, 1000, 2000$ with the MMHMC method. We also tested the four-stage M-ME$_4$ integrator, but since the results are worse than for M-ME$_3$, we do not include them in the plots for the sake of clarity. We adjust the step size $h$ and the

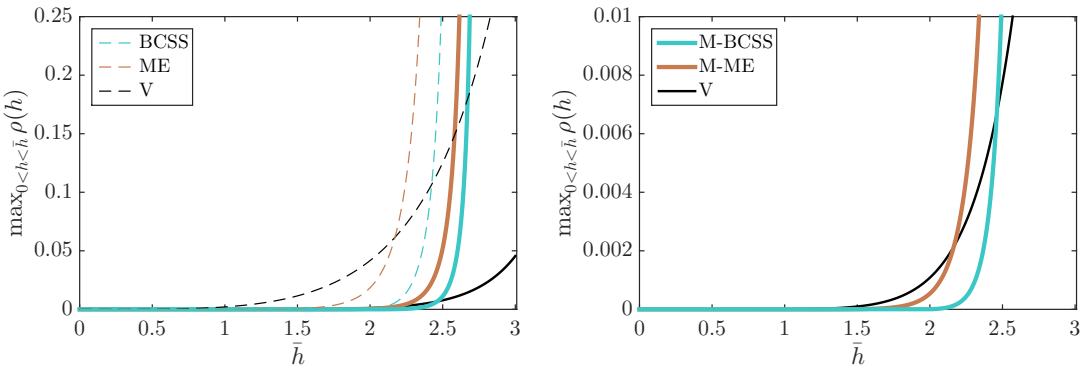


FIGURE 3.5: Upper bound for the expected energy error for the (M-)BCSS, (M-)ME and Verlet integrators for sampling with the true Hamiltonian (dashed) and 4th order modified Hamiltonian (solid). Right-hand graph is a zoom-in of the left-hand graph.

number of integration steps $L$ to the number of stages in the integrator such that the computational cost is equal for all tested integrators, e.g. for the Verlet we set $h_V = h/2$ and $L_V = 2L$. We discard the first 2000 samples from the collected 10000 and show results averaged over ten runs. Figure 3.6 presents the obtained acceptance rates as functions of the step size $h$. MMHMC specific integrators always result in higher AR than the corresponding ones derived for the HMC method. We note that for the small dimension ($D = 100$) the Verlet integrator remains the best choice, due to its larger stability limit. For bigger dimensions, which require smaller step sizes, better Hamiltonian conservation of two-stage integrators (see Figure 3.5) implies higher acceptance rates. In this case both the newly derived two-stage integrators show improvement over Verlet, with M-BCSS performing better than M-ME. Although the smallest error metric was obtained with M-ME$_3$ in the design of minimum error integrators (see Table 3.2), this integrator shows the worst performance, which might mean that the considered range of step sizes is close to the stability limit for the M-ME$_3$ (please, note that the stability limit of multi-stage integrators is dropping with number of stages).

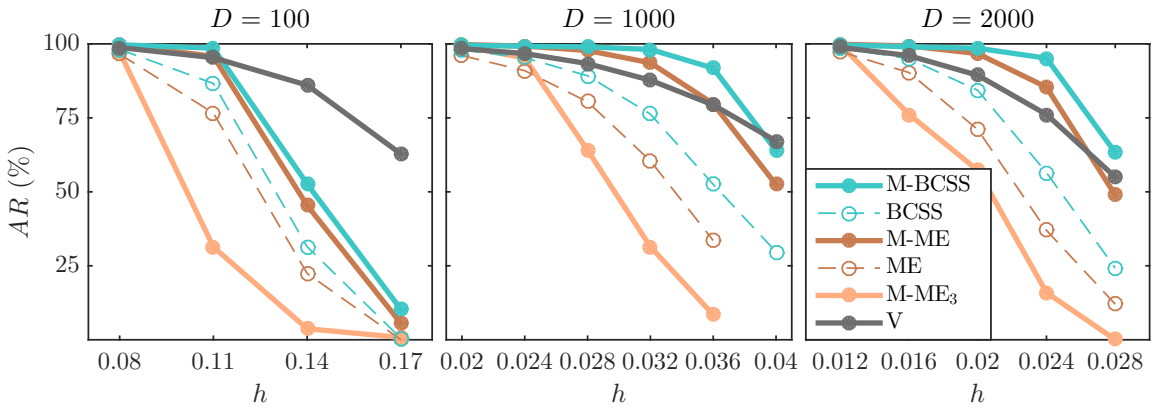


FIGURE 3.6: Acceptance rates as functions of the step size $h$ for sampling from a $D$-dimensional Gaussian distribution. Comparison of the two-stage (M-)BCSS, (M-)ME, three-stage M-ME$_3$ and Verlet integrators.

The relative sampling performance with respect to the Verlet integrator, in terms of minimum, median, and maximum ESS, obtained for the tested integrators is presented in Figure 3.7. Values below 1 correspond to cases of lower than Verlet's sampling efficiency and analogously, values above 1 correspond to an outperformance of an integrator over Verlet. The stars on the step size scale mark the choices of step size providing the best sampling performance for the considered problem. As in the case of resulting acceptance rates, for the smallest dimension, the Verlet integrator demonstrates the best performance. We note that for the smallest step sizes there is no difference among integrators. For bigger step sizes and dimensions, the M-BCSS integrator improves sampling efficiency over the Verlet up to 2.5 times for minimum ESS and up to 4 times for median and maximum ESS. The improvement clearly increases with dimension; therefore we believe that for high dimensional problems the new two-stage integrators are crucial component of an efficient sampler.

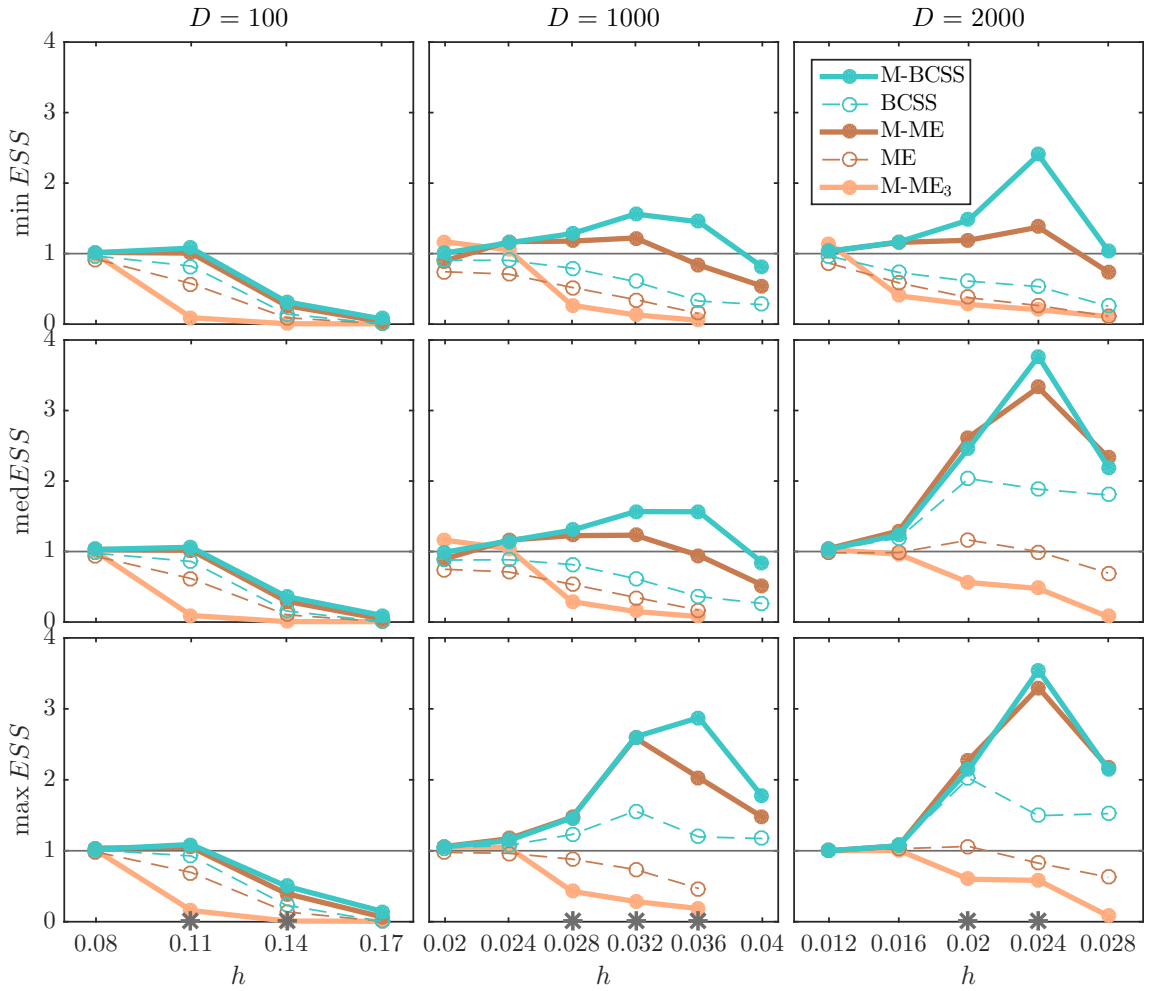


Figure 3.7: The relative sampling performance with respect to the Verlet integrator, as functions of the step size $h$ for sampling from a $D$-dimensional Gaussian distribution. Comparison of the two-stage (M-)BCSS, (M-)ME and three-stage M-ME$_3$ integrators.

In this section we have derived the first two- and three-stage integrators specially tuned

for modified Hamiltonians that guarantee minimal (expected) modified energy, leading to a better acceptance rate and sampling performance.

In the next two sections, we investigate alternative strategies for some components of the MMHMC method which may improve sampling or computational efficiency of MMHMC. We start with the analysis of a momentum update step.

### 3.2.3 Momentum update

Contrary to the HMC method, in which momentum is completely reset before numerical integration, the MMHMC method employs the Partial Momentum Monte Carlo (PMMC) step in the following manner.

For the current momentum $\mathbf{p}$ and a noise vector $\mathbf{u} \sim \mathcal{N}(0, M)$ we make a proposal

$$\begin{aligned} \mathbf{p}^* &= \sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u} \\ \mathbf{u}^* &= -\sqrt{\varphi}\mathbf{p} + \sqrt{1-\varphi}\mathbf{u} \end{aligned} \tag{3.40}$$

that is accepted according to the extended p.d.f.

$$\hat{\pi}(\boldsymbol{\theta}, \mathbf{p}, \mathbf{u}) \propto \exp\left(-\big(\tilde{H}(\boldsymbol{\theta}, \mathbf{p}) + \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u}\big)\right) \tag{3.41}$$

with probability

$$\mathcal{P} = \min\left\{1, \frac{\exp\big(-(\tilde{H}(\boldsymbol{\theta}, \mathbf{p}^*) + \frac{1}{2}(\mathbf{u}^*)^T M^{-1}\mathbf{u}^*)\big)}{\exp\big(-(\tilde{H}(\boldsymbol{\theta}, \mathbf{p}) + \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u})\big)}\right\}. \tag{3.42}$$

The parameter $\varphi \in (0, 1]$ controls the amount of noise introduced in every iteration and is related to the parameter $\phi$ from the original GSHMC formulation as $\varphi = \sin^2(\phi)$.

In the continuation of this Section, we derive a modified PMMC step that reduces the number of calculations of derivatives, and we also investigate a few alternative strategies for the momentum update, which were previously proposed in the literature.

#### 3.2.3.1 Modified PMMC step

The computational overhead of MMHMC compared to the HMC method includes two evaluations of the modified Hamiltonian within the Metropolis probability (3.42). With the aim of reducing the overhead, we modify the PMMC step such that the partial momentum update step is integrated into the modified Metropolis test.

Let us first consider the 4th order modified Hamiltonian (3.5) with analytical derivatives of the potential function, for which coefficients $c_{21}, c_{22}$ can be calculated either from (3.9), (3.10) or (3.11) for two-, three- or four-stage integrators or from (3.12) for the Verlet integrator. We find the difference in the extended "Hamiltonian", introduced in equation (2.32), between the current state and a state with partially updated momentum as

$$\Delta\hat{H} = \tilde{H}^{[4]}(\boldsymbol{\theta}, \mathbf{p}^*) + \frac{1}{2}(\mathbf{u}^*)^T M^{-1}\mathbf{u}^* - \tilde{H}^{[4]}(\boldsymbol{\theta}, \mathbf{p}) - \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u}$$

$$
\begin{aligned}
=&\cancel{U(\boldsymbol{\theta})} + \frac{1}{2}(\mathbf{p}^*)^T M^{-1}\mathbf{p}^* + h^2 c_{21}(\mathbf{p}^*)^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}^* + \cancel{h^2 c_{22} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})}) + \\
&\frac{1}{2}(\mathbf{u}^*)^T M^{-1}\mathbf{u}^* - \cancel{U(\boldsymbol{\theta})} - \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} - h^2 c_{21}\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} - \\
&\cancel{h^2 c_{22} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1} U_{\boldsymbol{\theta}}(\boldsymbol{\theta})}) - \frac{1}{2}\mathbf{u}^T M^{-1}\mathbf{u} \\
=&\frac{1}{2}\Big((\sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u})^T M^{-1}(\sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u}) + \\
&(-\sqrt{\varphi}\mathbf{p} + \sqrt{1-\varphi}\mathbf{u})^T M^{-1}(-\sqrt{\varphi}\mathbf{p} + \sqrt{1-\varphi}\mathbf{u}) - \mathbf{p}^T M^{-1}\mathbf{p} - \mathbf{u}^T M^{-1}\mathbf{u}\Big) + \\
&h^2 c_{21}\Big((\sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}(\sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u}) - \\
&\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}\Big) \\
=&\frac{1}{2}\Big(\cancel{(1-\varphi)\mathbf{p}^T M^{-1}\mathbf{p}} + \cancel{\varphi\mathbf{u}^T M^{-1}\mathbf{u}} + \cancel{2\sqrt{\varphi(1-\varphi)}\mathbf{u}^T M^{-1}\mathbf{p}} + \\
&\cancel{\varphi\mathbf{p}^T M^{-1}\mathbf{p}} + \cancel{(1-\varphi)\mathbf{u}^T M^{-1}\mathbf{u}} - \cancel{2\sqrt{\varphi(1-\varphi)}\mathbf{u}^T M^{-1}\mathbf{p}} - \cancel{\mathbf{p}^T M^{-1}\mathbf{p}} - \cancel{\mathbf{u}^T M^{-1}\mathbf{u}}\Big) + \\
&h^2 c_{21}\Big((\cancel{1-\varphi})\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} + \varphi\mathbf{u}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{u} - \\
&\cancel{\mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}} + 2\sqrt{\varphi(1-\varphi)}\mathbf{u}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}\Big) \\
=&h^2 c_{21}\Big(\varphi\big(\mathbf{u}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{u} - \mathbf{p}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}\big) + \\
&2\sqrt{\varphi(1-\varphi)}\mathbf{u}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}\Big).
\end{aligned}
\tag{3.43}
$$

Therefore, if the 4th order modified Hamiltonian (3.5) with analytical derivatives is used, a new momentum can be defined as

$$
\bar{\mathbf{p}} = \begin{cases} \sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u} & \text{with probability } \mathcal{P} = \min\{1, \exp(-\Delta\hat{H})\} \\ \mathbf{p} & \text{otherwise} \end{cases}
\tag{3.44}
$$

where $\mathbf{u} \sim \mathcal{N}(0, M)$ is the noise vector, $\varphi \in (0, 1]$ and

$$
\Delta\hat{H} = h^2 c_{21}\Big(\varphi A + 2\sqrt{\varphi(1-\varphi)}B\Big)
\tag{3.45}
$$

with

$$
\begin{aligned}
A &= (\mathbf{u} - \mathbf{p})^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}(\mathbf{u} + \mathbf{p}) \\
B &= \mathbf{u}^T M^{-1} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}.
\end{aligned}
\tag{3.46}
$$

Consequently, for models with no hierarchical structure, there is no need to calculate gradients within the PMMC step, second derivatives can be taken from the previous Hamiltonian Dynamics Metropolis test, and there is no need to generate $\mathbf{u}^*$.

We note here that in our implementation of the MMHMC method, a gradient calculation is not necessary at this stage even when using the original PMMC step because we keep a track of the current gradient in addition to the current position and momenta variables.

In Figure 3.8 we show the saving in computational time observed when using the new PMMC step instead of the original PMMC step, as a function of the number of integration steps, for a model with dense Hessian matrix, using the modified Hamiltonian (3.5)

with analytical derivatives. Clearly, for shorter HD trajectories the new momentum update significantly improves the performance of MMHMC (up to 60% faster).

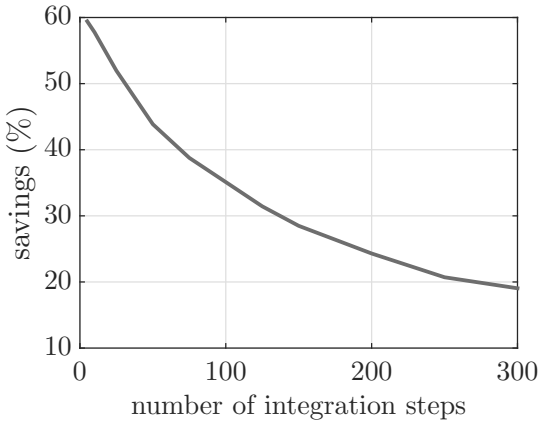


FIGURE 3.8: Saving in computational time with the new PMMC step over the original PMMC step, using the 4th order modified Hamiltonian (3.5) with analytical derivatives, for a model with no hierarchical structure and dense Hessian of the potential function.

In the case of the 6th order modified Hamiltonian (3.8) for Gaussian problems, the error in the extended Hamiltonian (2.32) that enters the Metropolis test (3.44) can be calculated in a similar manner

$$\Delta\hat{H} = h^2 c_{21}\Big(\varphi(A-B) + 2\sqrt{\varphi(1-\varphi)}C\Big) + h^4 c_{44}\Big(\varphi(D-E) + 2\sqrt{\varphi(1-\varphi)}F\Big), \quad (3.47)$$

with

$$\begin{aligned} A &= \mathbf{u}^T M^{-1} U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{u} \\ B &= \mathbf{p}^T M^{-1} U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} \\ C &= \mathbf{u}^T M^{-1} U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} \\ D &= (U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{u})^T M^{-1} U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{u} \\ E &= (U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p})^T M^{-1} U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} \\ F &= (U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{u})^T M^{-1} U_{\boldsymbol{\theta\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p}. \end{aligned} \quad (3.48)$$

For the 4th order modified Hamiltonian (3.22) calculated using numerical time derivatives of the gradient of the potential function, and for the Verlet, two-, three- and four-stage integrators, we calculate the difference in the extended Hamiltonian as

$$\Delta\hat{H} = h k_{21}\Big((\mathbf{p}^*)^T P_1^* - \mathbf{p}^T P_1\Big), \quad (3.49)$$

where $P_1^*$ is the first order scaled time derivative of the gradient (see Section 3.2.1.2) calculated from the trajectory with updated momentum $\mathbf{p}^*$. The computational gain of the new PMMC step, in this case, results from not having to calculate the term multiplying $k_{22}$ in the modified Hamiltonian (3.22). In our implementation, however, this term is of negligible cost, therefore, the gain from the new expression for the error (3.49) in the extended

Hamiltonian that enters the Metropolis test (3.44) is not as significant as for the error (3.45) derived for the 4th order modified Hamiltonian (3.5) with analytical derivatives.

For the 6th order modified Hamiltonian (3.23) with numerical time derivatives the difference in the extended Hamiltonian may be calculated as

$$\begin{aligned}\Delta\hat{H} =& hk_{21}\Big((\mathbf{p}^*)^T P_1^* - \mathbf{p}^T P_1\Big) + hk_{41}\Big((\mathbf{p}^*)^T P_3^* - \mathbf{p}^T P_3\Big) \\ &+ h^2 k_{42}\Big(U_{\mathbf{x}}^T P_2^* - U_{\mathbf{x}}^T P_2\Big) + h^2 k_{43}\Big((P_1^*)^T P_1^* - P_1^T P_1\Big),\end{aligned} \tag{3.50}$$

where $P_2^*, P_3^*$ are second and third order scaled time derivatives, respectively. These may be computed as in Section 3.2.1.2 from trajectories with updated momenta $\mathbf{p}^*$. The saving in computation arises from the absence of terms multiplying $k_{22}$ and $k_{44}$ in the modified Hamiltonian (3.23), which in this case is not negligible, contrary to the case of the 4th order modified Hamiltonian (3.22).

In this section, we provided new formulations for the momentum Metropolis test for the 4th and 6th order modified Hamiltonians, with analytical and numerical derivatives. In the case of the 6th order modified Hamiltonian, with derivatives calculated either analytically or numerically, the new expressions for momentum refreshment lead to computational saving compared to the original GSHMC method, as is the case with the 4th order modified Hamiltonian with analytical derivatives. In the latter case, however, if the Hessian matrix of the potential function is dense, instead of using the modified Hamiltonian with analytical derivatives, we recommend using numerical derivatives, for which the saving is negligible. On the other hand, if the computation of the Hessian matrix is not very costly (e.g. being block-diagonal, sparse, close to constant), it might be more efficient to use analytical derivatives, for which the new formulation of the Metropolis test leads to computational saving.

#### 3.2.3.2 Change of momentum variables

It might be useful to have a control on the acceptance rate of a PMMC step, $\text{AR}_{\text{PMMC}}$, as extremely low or high $\text{AR}_{\text{PMMC}}$ may lead to loss of sampling performance.

One possible strategy for keeping the acceptance rate of a PMMC step from being too low was mentioned in (Akhmatskaya and Reich, 2008). Inspired by the approach in the S2HMC method (Sweet et al., 2009), the authors suggest performing a change of momenta variables as

$$\hat{\mathbf{p}} = \mathcal{T}(\boldsymbol{\theta}, \mathbf{p}, h),$$

where the transformation $\mathcal{T}$ is invertible in $\mathbf{p}$. The PMU step

$$\begin{aligned}\hat{\mathbf{p}}^* &= \sqrt{1-\varphi}\hat{\mathbf{p}} + \sqrt{\varphi}\mathbf{u} \\ \mathbf{u}^* &= -\sqrt{\varphi}\hat{\mathbf{p}} + \sqrt{1-\varphi}\mathbf{u}\end{aligned}$$

then takes the place of the original PMU step (3.40). The new momentum $\mathbf{p}^* = \mathcal{T}^{-1}(\boldsymbol{\theta}, \hat{\mathbf{p}}^*, h)$ is then still accepted with probability (3.42). It was suggested in (Akhmatskaya and Reich,

2008) that the transformation $\mathcal{T}$ for the Verlet integrator could be defined as

$$\hat{\mathbf{p}} = \mathbf{p} - \frac{h}{24}\left(U_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{+}) - U_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{-})\right),$$

where $\boldsymbol{\theta}^{+}$ and $\boldsymbol{\theta}^{-}$ are position vectors for the additional forward and backward integration steps, respectively. We note that this definition actually corresponds to

$$\hat{\mathbf{p}} = \mathbf{p} - hk_{21}P_1 \tag{3.51}$$

for the 4th order modified Hamiltonian (3.22) with numerical time derivatives, or to

$$\hat{\mathbf{p}} = \mathbf{p} - h^2 c_{21} U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) M^{-1}\mathbf{p} \tag{3.52}$$

for the 4th order modified Hamiltonian (3.5) with analytical derivatives. Both types of transformation correspond to multi-stage integrators (2.21)–(2.23). Making use of definition (3.51), the new momentum $\mathbf{p}^*$ is implicitly defined by

$$\hat{\mathbf{p}}^* = \mathbf{p}^* - hk_{21}P_1^*$$

and can be obtained as an iterative solution.

In this thesis, however, we choose a different strategy and recover the "untransformed" momentum $\mathbf{p}^*$ using $P_1$ calculated from the old momentum $\mathbf{p}$ as

$$\mathbf{p}^* = \hat{\mathbf{p}}^* + hk_{21}P_1 \approx \mathcal{T}^{-1}(\boldsymbol{\theta}, \hat{\mathbf{p}}^*, h). \tag{3.53}$$

Thus, we avoid the iterative solution by assuming the $h$ term in the transformation $\mathcal{T}$ defined in (3.51) is constant within an MC step. The new momentum $\mathbf{p}^*$ is accepted according to the target distribution, and so the invariant distribution is preserved. The Metropolis probability for the new PMMC step in our case becomes

$$\mathcal{P} = \min\{1, \exp(-\Delta\hat{H})\},$$

where

$$\Delta\hat{H} = hk_{21}\Big((\mathbf{p}^*)^T P_1^* - 2\mathbf{p}^T P_1 + (\hat{\mathbf{p}}^*)^T P_1 + hk_{21}P_1^T P_1\Big).$$

Making use of the transformation (3.52) and following the same approach, we obtain

$$\Delta\hat{H} = h^2 c_{21}\Big((\mathbf{p}^*)^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathbf{p}^* - 2\mathbf{p}^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathbf{p} + (\hat{\mathbf{p}}^*)^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathbf{p} + h^2 c_{21}(U_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathbf{p})^T U_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathbf{p}\Big)$$

for the modified Hamiltonian (3.5) with analytical derivatives.

We implemented a change of momenta variables and tested this technique sampling from a 100-dimensional Gaussian distribution with the 4th order modified Hamiltonian (3.22) with numerical time derivatives. We note that the computational overhead consists only in a few simple multiplications of already precomputed values and is, therefore,

negligible.

Figure 3.9 demonstrates the steady improvement in momentum acceptance rates with the use of a change of momenta variables for a range of step sizes $h$ and parameter $\varphi$. Nevertheless, it clearly reveals that a high momentum acceptance rate does not necessarily mean better performance. Indeed, as follows from Figure 3.9 for the studied system, the best performance (minimum ESS across variates) is observed at small values of $\varphi$. Such a choice of $\varphi$ always guarantees a high momentum acceptance rate and using a change of variables does not provide extra benefits.

Nevertheless, too high momentum acceptance rates achieved with a change of variables for bigger $\varphi$ can lead to lowering a position acceptance rate and thus to a performance degradation. This becomes more obvious when $\varphi$ is increasing. In these cases, too high acceptance of unfavorable momenta with a change of variable leads to a noticeable decrease in performance.

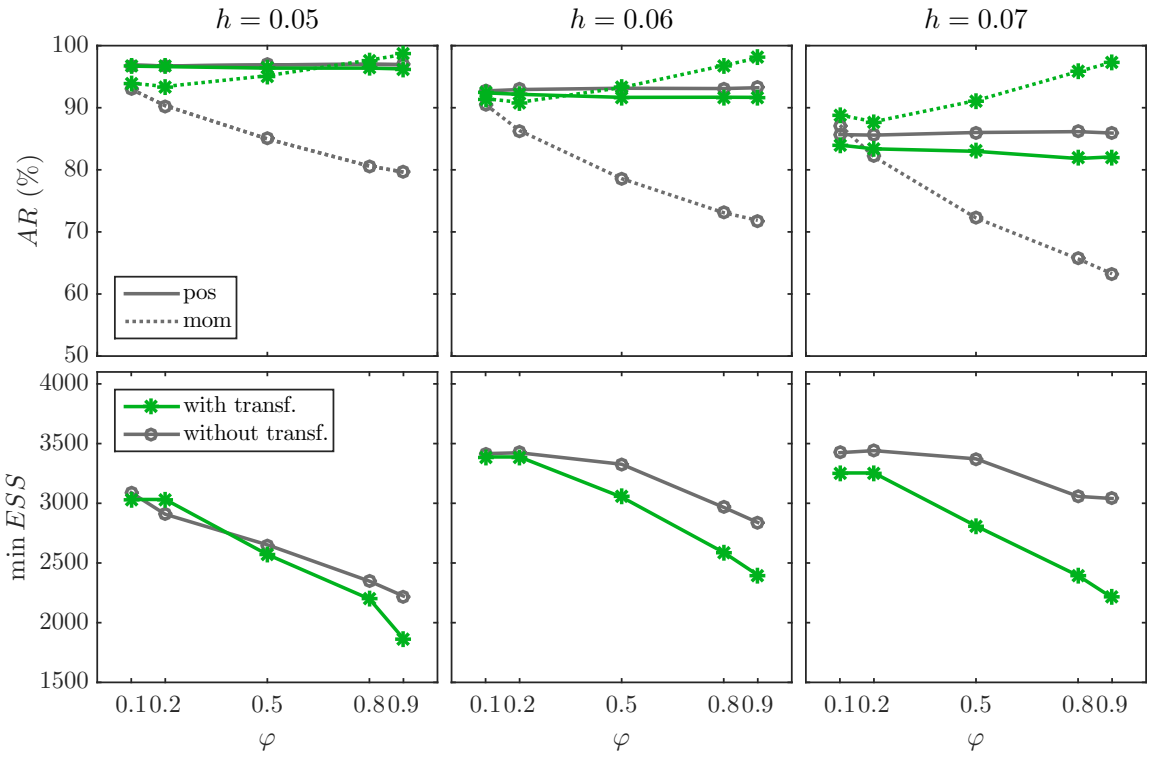


Figure 3.9: Acceptance rate and minimum ESS across variates for sampling from a 100-dimensional Gaussian distribution with the 4th order modified Hamiltonian (3.22) with numerical time derivatives of the gradient, depending on different step size $h$ and noise parameter $\varphi$. Although transformation of momenta variables (green) improves momentum acceptance rate for all parameters, it does not improve position acceptance rate and ESS compared to the original method without momenta transformation (grey).

#### 3.2.3.3 Repeat momenta update

Increasing an acceptance rate in a PMMC step can also be achieved by a repetition of a momentum update step. We mention four alternatives to the single PMMC step in the

MMHMC method. These techniques were proposed earlier, but it was not clear whether they provide improved results. A strategy suggested by Akhmatskaya and Reich (2008) is to update momentum by repeating the PMMC step $n$ times iteratively. Each of the several PMMC steps, an HDMC step, and a momentum flip step in the case of rejection, leaves the target distribution invariant; therefore their concatenation preserves the target distribution. This technique may plausibly improve the efficiency but at an increased computational cost. The second alternative is to repeat the momentum update until acceptance, as carried out in the SHMC method (Izaguirre and Hampton, 2004). Thirdly, the momentum update could be performed $n$ times but each time starting from the current momentum and taking the first accepted value as the next momentum. If none of the $n$ proposed momenta was accepted, we continue to use the current one. This procedure can easily be performed in parallel, resulting in no additional computational cost.

Finally, the orientational bias Monte Carlo (OBMC) (or multiple-try Metropolis method) as proposed by Liu et al. (2000) can be applied to enhance the acceptance rate in the momentum update within Generalized Hamiltonian Monte Carlo methods. The OBMC method provides a rigorous tool to exploit multiple (parallel) proposals within a Monte Carlo context.

The basic idea is to generate in parallel $k$ trial momentum vectors $\mathbf{p}_i^*, i = 1, \ldots, k$ given a momentum vector $\mathbf{p}$ as

$$\begin{aligned}\mathbf{p}_i^* &= \sqrt{1-\varphi}\mathbf{p} + \sqrt{\varphi}\mathbf{u}_i \\ \mathbf{u}_i^* &= -\sqrt{\varphi}\mathbf{p} + \sqrt{1-\varphi}\mathbf{u}_i.\end{aligned}$$

Select $\bar{\mathbf{p}} = \mathbf{p}_l^*$ among the momentum vectors $\{\mathbf{p}_i^*\}$ with probability proportional to the extended probability $\hat{\pi}$ given in Equation (3.41). Next, generate another $k-1$ reference points $\{\hat{\mathbf{p}}_i\}$ using the momentum proposal step with $\bar{\mathbf{p}}$ as the initial value and set $\hat{\mathbf{p}}_k = \mathbf{p}$. Finally accept $\bar{\mathbf{p}}$ with probability

$$\min\left\{1, \frac{\sum_{i=1}^k \hat{\pi}(\mathbf{p}_i^*)}{\sum_{i=1}^k \hat{\pi}(\hat{\mathbf{p}}_i)}\right\}$$

and reject with the remaining probability.

This method may be reasonable to consider only if all multiple trials are implemented in parallel in the MMHMC code. Even in this case, the computational overhead is expected due to an additional momentum update on each processing element (PE) and unavoidable communications between PEs.

We implemented the first and third technique, namely iterative repetition of the PMMC step $n$ times, which we denote itPMMC, and $n$ parallel momenta updates with the first accepted as the next momentum, which we call parPMMC. We performed tests on a 100-dimensional Gaussian problem and show results in Tables 3.3 and 3.4, respectively. We used a step size $h = 0.07$, a number of integration steps $L = 300$ and three different choices of parameter $\varphi$ and the number of repetitions $n = 1, 5, 10$. The itPMMC technique (Table 3.3) improves only slightly both position and momentum acceptance rates. Only maximum ESS across variates increases for larger numbers of repetitions and median ESS in the case of the

small value of $\varphi$. Nevertheless, time normalized ESS indicates that there is no advantage in using this technique.

| | | AR (%) | | Time | ESS | | | ESS/sec | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\varphi$ | $n$ | $\boldsymbol{\theta}$ | $\mathbf{p}$ | (sec.) | min | med | max | min | med | max |
| | 1 | 85.32 | 87.04 | 8.53 | 3388 | 4025 | 5079 | 397 | 472 | 595 |
| 0.1 | 5 | 85.74 | 87.11 | 9.18 | 3355 | 4160 | 5737 | 366 | 453 | 625 |
| | 10 | 86.19 | 87.12 | 10 | 3109 | 4200 | 5988 | 310 | 419 | 597 |
| | 1 | 86.05 | 71.8 | 8.53 | 3354 | 4202 | 5806 | 393 | 493 | 681 |
| 0.5 | 5 | 86.24 | 71.52 | 9.18 | 2575 | 3844 | 6278 | 281 | 419 | 684 |
| | 10 | 86.48 | 71.75 | 10 | 2218 | 3372 | 6322 | 221 | 336 | 630 |
| | 1 | 86.06 | 63.31 | 8.53 | 3038 | 4149 | 5778 | 356 | 486 | 677 |
| 0.9 | 5 | 86.34 | 63.25 | 9.18 | 1955 | 3260 | 6156 | 213 | 355 | 671 |
| | 10 | 86.32 | 62.81 | 10 | 2013 | 3191 | 6305 | 201 | 318 | 629 |

Table 3.3: Iterative repetition of the PMMC step $n$ times for sampling from a $100$-dimensional Gaussian distribution.

The parPMMC (Table 3.4) clearly improves the momentum acceptance rate; however, the position acceptance rate remains on the same level as well as ESS values.

| | | AR (%) | | ESS | | |
|---|---|---|---|---|---|---|
| $\varphi$ | $n$ | $\boldsymbol{\theta}$ | $\mathbf{p}$ | min | med | max |
| | 1 | 85.32 | 87.28 | 3278 | 3988 | 4904 |
| 0.1 | 5 | 85.42 | 100 | 3398 | 4040 | 4967 |
| | 10 | 85.58 | 100 | 3325 | 3948 | 5233 |
| | 1 | 85.8 | 72.27 | 3288 | 4076 | 5547 |
| 0.5 | 5 | 85.37 | 99.56 | 3178 | 4141 | 5531 |
| | 10 | 85.5 | 99.99 | 3198 | 4151 | 5543 |
| | 1 | 85.82 | 63.14 | 3062 | 4146 | 5829 |
| 0.9 | 5 | 85.59 | 97.57 | 2545 | 3737 | 5826 |
| | 10 | 85.14 | 99.72 | 2515 | 3644 | 5701 |

Table 3.4: Repetition of the current momentum update $n$ times, taking the first accepted as the next momentum or continuing with the current one if all $n$ proposed momenta were rejected, for sampling from a $100$-dimensional Gaussian distribution. No data for ESS/sec is shown as parPMMC does not introduce the overhead if run in parallel.

Since our results show little or no improvement between these different approaches for the momentum update, we subsequently employ only a single momentum update for the rest of this work.

In the next section, we outline previously proposed alternative strategies to the automatic momenta flip upon rejections and investigate whether they improve sampling efficiency within the MMHMC method.

### 3.2.4 Reduced flipping

In order to satisfy the detailed balance condition and ensure a stationary distribution, a momentum flip upon rejection of a Hamiltonian Dynamics proposal step is required for methods employing the partial momentum update. These momentum reversals combined with small values of parameter $\varphi$ may lead to potential problems. It was noted that momentum reversals might cause slow exploration of the state space and therefore slow decorrelation of the chain or can have a significant impact on molecular kinetics (Akhmatskaya et al., 2009a; Akhmatskaya et al., 2009b; Wagoner and Pande, 2012). This effect was investigated for molecular simulation problems in (Akhmatskaya et al., 2009a; Akhmatskaya et al., 2009b; Wagoner and Pande, 2012) and only tackled for a simple statistical problem in (Sohl-Dickstein, 2012). For a computational statistics problem, there is no physical dynamics of the simulated system that we wish to maintain and it is not clear, however, whether momenta reversals cause problems or actually help sampling.

A possible way to reduce an impact of flipping would be to decrease the rejection rate so that double-backing of trajectories occur only occasionally. This could be achieved by (a) reducing the step size, which actually increases the computational cost; (b) using multistage integrators for high dimensional problems (Blanes et al. (2014), Section 3.2.2 of this thesis) or (c) delaying rejections, as done in (Sohl-Dickstein et al., 2014; Campos and Sanz-Serna, 2015). Another strategy would be to try to decrease the number of momentum flips.

In this section, we outline the techniques for the latter strategy and investigate the effect of reducing momentum flips on sampling efficiency in computational statistics.

**No flip (Akhmatskaya et al., 2009a)** The authors proposed the GHMC/GSHMC method without momentum flipping upon rejection, for which the Metropolis test (3.1) becomes

$$(\boldsymbol{\theta}^{\text{new}}, \mathbf{p}^{\text{new}}) = \begin{cases} (\boldsymbol{\theta}', \mathbf{p}') & \text{with probability } \alpha\big((\boldsymbol{\theta}, \mathbf{p}), (\boldsymbol{\theta}', \mathbf{p}')\big) \\ (\boldsymbol{\theta}, \mathbf{p}) & \text{otherwise.} \end{cases}$$

It is demonstrated that the method without momentum flipping is capable of accurately reproducing the desired distribution, provided the rejection rate is kept sufficiently small, though the algorithm cannot be proven to satisfy the detailed balance condition. Numerical evidence indicates, however, that the standard GHMC/GSHMC method with momentum flip leads to higher acceptance rates and more efficient sampling. On the other hand, the results demonstrate the large impact of the momentum flip on dynamic properties of the simulated system.

**Reduced flip (Sohl-Dickstein, 2012)** The author proposed a technique within the GHMC method, which can reduce the number of momentum flips by making the distribution of interest a fixed point. This technique introduces calculation of the probability of the momentum flip, which in the case of rejection of the proposal $\Psi_{hL}(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}', \mathbf{p}')$ is given

by

$$P_F(\boldsymbol{\theta}, \mathbf{p}) = \max\left\{0, \min\left\{1, \frac{p\left(\Psi_{hL}(\boldsymbol{\theta}, -\mathbf{p})\right)}{p(\boldsymbol{\theta}, \mathbf{p})}\right\} - \min\left\{1, \frac{p\left(\Psi_{hL}(\boldsymbol{\theta}, \mathbf{p})\right)}{p(\boldsymbol{\theta}, \mathbf{p})}\right\}\right\}.$$

The Metropolis test is then defined as

$$(\boldsymbol{\theta}^{\text{new}}, \mathbf{p}^{\text{new}}) = \begin{cases} \Psi_{hL}(\boldsymbol{\theta}, \mathbf{p}) & \text{with probability } \alpha\big((\boldsymbol{\theta}, \mathbf{p}), \Psi_{hL}(\boldsymbol{\theta}, \mathbf{p})\big) \\ \mathcal{F}(\boldsymbol{\theta}, \mathbf{p}) & \text{with probability } P_F \\ (\boldsymbol{\theta}, \mathbf{p}) & \text{otherwise.} \end{cases} \tag{3.54}$$

The computational overhead introduced by this probability includes an additional evaluation of the reverse trajectory and the probability of the resulting state $\Psi_{hL}(\boldsymbol{\theta}, -\mathbf{p})$. This overhead depends on the rejection rate, due to additional calculations only in the case of rejections. The author demonstrates a slight improvement in the autocovariance function compared to automatic flipping, though the test was performed on a very simple 2-dimensional model.

**Reduced flip (Wagoner and Pande, 2012)** Another modification of the traditional automatic-flipping GHMC method, called Reduced-Flipping GHMC, was suggested by Wagoner and Pande (2012). The authors proposed a simple technique that uses the information of the previous, current, and candidate states to reduce the probability of momentum flipping following the candidate rejection, while rigorously satisfying the detailed balance condition. In the case of rejection of the HD proposal $(\boldsymbol{\theta}', \mathbf{p}')$, the probability of flipping momentum within the Metropolis test (3.54) is given by

$$P_F((\boldsymbol{\theta}, \mathbf{p})|(\boldsymbol{\theta}^{\text{prev}}, \mathbf{p}^{\text{prev}}), (\boldsymbol{\theta}', \mathbf{p}')) = 1 - \alpha\big((\boldsymbol{\theta}, \mathbf{p}), (\boldsymbol{\theta}', \mathbf{p}')\big) - P_S((\boldsymbol{\theta}, \mathbf{p})|(\boldsymbol{\theta}^{\text{prev}}, \mathbf{p}^{\text{prev}}), (\boldsymbol{\theta}', \mathbf{p}')),$$

where the probability of the state $(\boldsymbol{\theta}, \mathbf{p})$ is

$$P_S((\boldsymbol{\theta}, \mathbf{p})|(\boldsymbol{\theta}^{\text{prev}}, \mathbf{p}^{\text{prev}}), (\boldsymbol{\theta}', \mathbf{p}')) =$$

$$\begin{cases} \min\left\{1 - \alpha\big((\boldsymbol{\theta}, \mathbf{p}), (\boldsymbol{\theta}', \mathbf{p}')\big), \frac{\alpha\left((\boldsymbol{\theta}, \mathbf{p}), (\boldsymbol{\theta}', \mathbf{p}')\right)}{\alpha\left(\mathcal{F}(\boldsymbol{\theta}, \mathbf{p}), \mathcal{F}(\boldsymbol{\theta}^{\text{prev}}, \mathbf{p}^{\text{prev}})\right)}\left(1 - \alpha\big(\mathcal{F}(\boldsymbol{\theta}, \mathbf{p}), \mathcal{F}(\boldsymbol{\theta}^{\text{prev}}, \mathbf{p}^{\text{prev}})\big)\right)\right\} \\ \qquad \text{if } (\boldsymbol{\theta}^{\text{prev}}, \mathbf{p}^{\text{prev}}) \to (\boldsymbol{\theta}, \mathbf{p}) \text{ was an accepted move} \\ 0 \qquad \text{otherwise.} \end{cases}$$

and $\alpha(\boldsymbol{\xi}, \boldsymbol{\xi}')$ is the acceptance probability of a transition $\boldsymbol{\xi} \to \boldsymbol{\xi}'$. Using this method, the authors observed an improvement in terms of autocorrelations over automatic flipping for high acceptance rates. Nevertheless, no advantage of this technique was noted for bigger step sizes and low acceptance rates neither bigger values of $\varphi$ (Wagoner and Pande, 2012).

We adapted the reduced flipping technique by Wagoner and Pande (2012) within the MMHMC method. The Metropolis test (3.1) now becomes

$$(\boldsymbol{\theta}^{\text{new}}, \mathbf{p}^{\text{new}}) = \begin{cases} (\boldsymbol{\theta}', \mathbf{p}') & \text{with probability } \alpha\big((\boldsymbol{\theta}, \mathbf{p}), (\boldsymbol{\theta}', \mathbf{p}')\big) \\ \mathcal{F}(\boldsymbol{\theta}, \mathbf{p}) & \text{with probability } P_F \\ (\boldsymbol{\theta}, \mathbf{p}) & \text{otherwise,} \end{cases}$$

where we simplified the flipping probability as

$$P_F((\boldsymbol{\theta},\mathbf{p})|(\boldsymbol{\theta}^{\text{prev}},\mathbf{p}^{\text{prev}}),(\boldsymbol{\theta}',\mathbf{p}')) = \begin{cases} \max\left\{0, 1-\frac{\alpha\left((\boldsymbol{\theta},\mathbf{p}),(\boldsymbol{\theta}',\mathbf{p}')\right)}{\alpha\left(\mathcal{F}(\boldsymbol{\theta},\mathbf{p}),\mathcal{F}(\boldsymbol{\theta}^{\text{prev}},\mathbf{p}^{\text{prev}}))\right)}\right\} \\ \qquad \text{if } (\boldsymbol{\theta}^{\text{prev}},\mathbf{p}^{\text{prev}}) \to (\boldsymbol{\theta},\mathbf{p}) \text{ was an accepted move} \\ 1-\alpha((\boldsymbol{\theta},\mathbf{p}),(\boldsymbol{\theta}',\mathbf{p}')) \quad \text{otherwise.} \end{cases}$$

and probability $\alpha(\cdot,\cdot)$ is defined through a modified Hamiltonian.

In Table 3.5 we report all position and momentum acceptance rates and the reduced flipping rate (RFR) obtained with the reduced flipping technique on a 100-dimensional Gaussian problem. RFR is calculated as a portion of rejected samples for which the momentum flip was not applied. We also compare MMHMC with automatic flipping, reduced

| | | AR (%) | | RFR |
|---|---|---|---|---|
| $\varphi$ | $h$ | $\boldsymbol{\theta}$ | $\mathbf{p}$ | (%) |
| 0.1 | 0.055 | 95.26 | 91.71 | 33.03 |
| | 0.06 | 93.00 | 90.17 | 29.51 |
| | 0.07 | 85.32 | 87.26 | 22.28 |
| | 0.08 | 72.12 | 83.86 | 12.99 |
| | 0.085 | 62.41 | 82.18 | 8.73 |
| 0.5 | 0.055 | 95.22 | 82.00 | 30.32 |
| | 0.06 | 93.10 | 78.68 | 28.07 |
| | 0.07 | 85.93 | 72.20 | 22.02 |
| | 0.08 | 73.00 | 65.52 | 12.93 |
| | 0.085 | 63.22 | 61.58 | 8.97 |
| 0.9 | 0.055 | 95.23 | 75.79 | 28.35 |
| | 0.06 | 93.02 | 71.90 | 27.46 |
| | 0.07 | 86.09 | 63.40 | 22.08 |
| | 0.08 | 72.85 | 54.76 | 13.52 |
| | 0.085 | 62.86 | 50.84 | 8.53 |

Table 3.5: Position and momenta acceptance rates and reduced flipping rates (RFR) obtained in sampling from a $100$-dimensional Gaussian distribution using MMHMC with reduced flipping, for a range of values of the noise parameter $\varphi$ and step size $h$.

flipping and no flipping techniques on a 100-dimensional Gaussian problem. Figure 3.10 shows acceptance rates and minimum ESS across variates obtained for different values of the noise parameter $\varphi$ and step size $h$. We observe that acceptance rates are not affected by any of these techniques and sampling efficiency is comparable for all of them.

While in molecular simulations a momentum flip can indeed have a negative impact on dynamics, in computational statistics there is no clear evidence regarding a harmful influence on the sampling performance. Nevertheless, the implementation of a statistically rigorous yet optional tool for reduced flipping can help in collecting the information on the role of momentum flip in MMHMC.

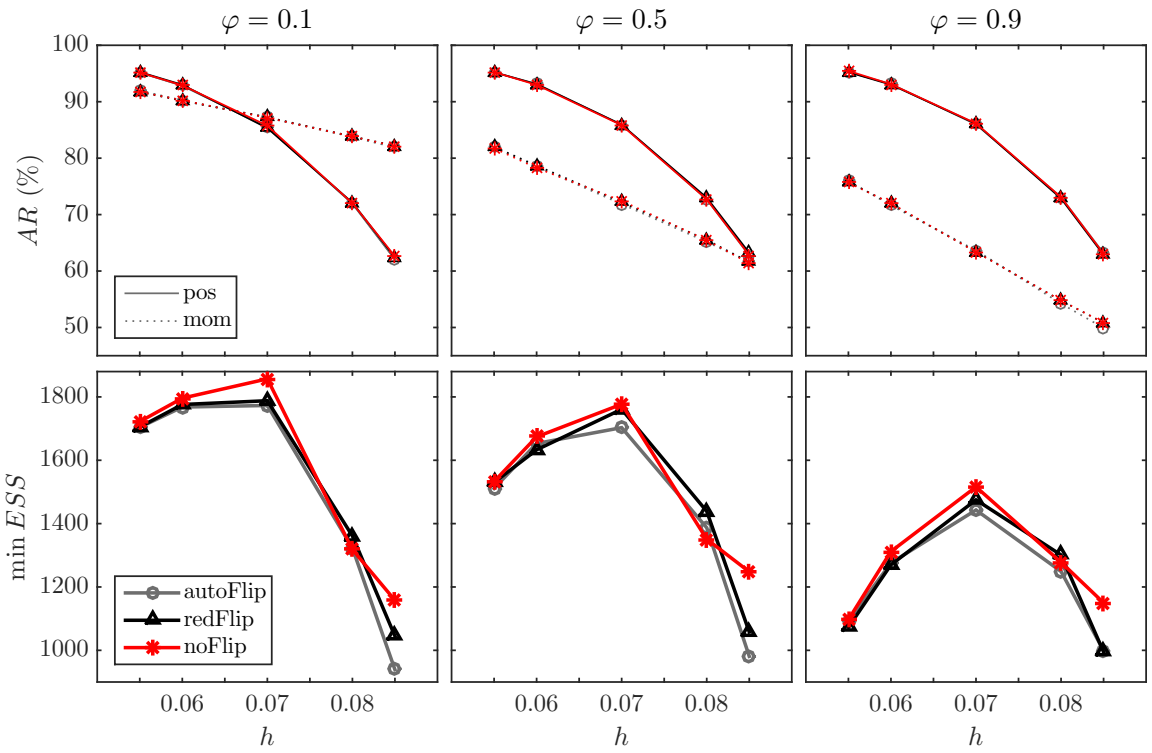


Figure 3.10: Acceptance rates and minimum ESS across variates for sampling from a 100-dimensional Gaussian distribution using MMHMC with automatic (grey), reduced (black) and no flipping (red) techniques. All methods demonstrate comparable sampling efficiency for the range of values of the noise parameter $\varphi$ and step size $h$.

### 3.2.5 Choice of parameters

The performance of the MMHMC method is affected by the choice of simulation parameters, namely the step size $h$, the number of integration steps $L$, the mass matrix $M$, the noise parameter $\varphi$ and the order of the modified Hamiltonian. A typical procedure for tuning parameters is heuristic and time-consuming, which is also true for the HMC method but with only three parameters. The whole discussion on the choice of parameters in HMC and GSHMC (see Sections 2.2.4 and 2.3.3) applies to the MMHMC method, with a few additional insights. In this section, we present some examples illustrating an effect of different parameters on the MMHMC performance in sampling from a 100-dimensional Gaussian distribution.

As we stated before, MMHMC allows for larger values of step sizes compared to HMC, while maintaining a high level of acceptance rate. In many cases, those larger values can result in better overall sampling efficiency than do values found to be optimal for HMC. Figure 3.11 illustrates this fact by displaying the dependence of acceptance rates (left-hand graph) and minimum ESS (right-hand graph) on the choice of step size $h$ for MMHMC and HMC. The plot shows that the best performance of MMHMC is achieved when acceptance rates are around $90\%$ whereas for HMC the best ESS results are achieved at smaller step sizes and lower acceptance rates.

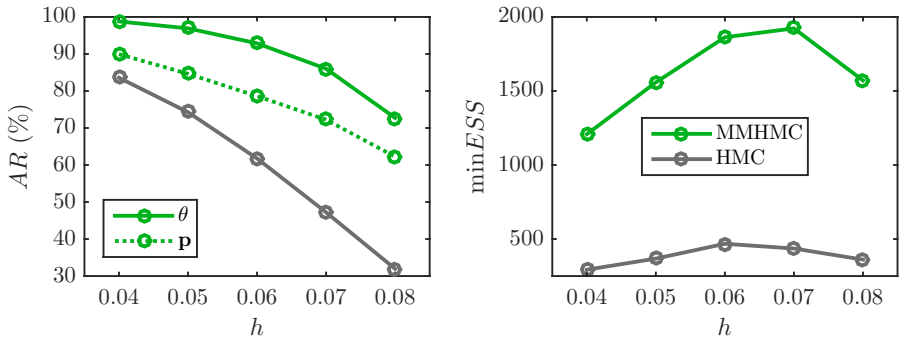


Figure 3.11: Position and momenta acceptance rates (left) and minimum ESS (right) obtained in sampling from a 100-dimensional Gaussian distribution using MMHMC and HMC with different step size $h$.

MMHMC has an additional advantage over HMC in terms of its sensitivity to the choice of the number of integration steps $L$. As will be shown in Chapter 6, the numerical experiments demonstrate that MMHMC is not as sensitive on the choice of $L$ as the HMC method, which may reduce the necessity for fine tuning of this parameter.

Nevertheless, as in HMC, in MMHMC the meaning of an "optimal" $L$ often remains unclear. Figure 3.12 demonstrates that there is no single optimal choice of $L$ for all variates.

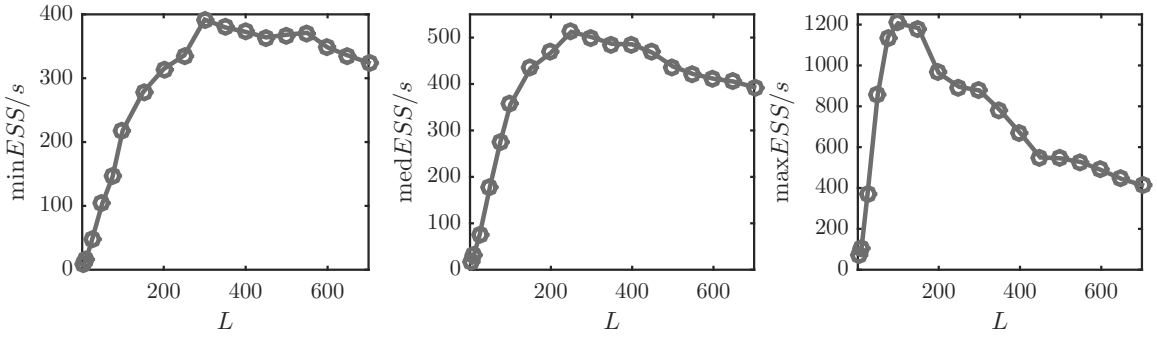


Figure 3.12: Time-normalized minimum, median and maximum ESS obtained in sampling from a 100-dimensional Gaussian distribution using MMHMC with different number of integration steps $L$.

As a general recommendation for HMC, it is advisable to randomize both step size and number of integration steps within the MMHMC method.

We now show the effect of the noise parameter $\varphi$ on the performance of MMHMC. Figure 3.13 presents position and momenta acceptance rates (top) and sampling efficiency, in terms of time-normalized minimum ESS (bottom) in the problem of sampling from a 100-dimensional Gaussian distribution for different choices of trajectory length $hL$. We report results for three different choices of the noise parameter $r\varphi$, namely using a fixed value $\varphi$ at every MC iteration, i.e. $r = 1$; choosing a random value uniformly from the interval $(0.8\varphi, 1.2\varphi)$, i.e. $r \sim \mathcal{U}(0.8, 1.2)$; and choosing a random value uniformly from the interval $(0, \varphi)$, i.e. $r \sim \mathcal{U}(0, 1)$. Position acceptance rate is not affected by $\varphi$, unless $\varphi = 1$ at which it slightly drops, whereas the acceptance rate of the PMMC step is higher for smaller

values of $\varphi$. Bigger values of $\varphi$, corresponding to more random noise introduced in momenta, might mean better space exploration; however, those values lead to more momenta rejections. For smaller trajectory length $hL$, smaller values of $\varphi$ result in better sampling efficiency, while for longer $hL$ very small values of $\varphi$ might not be the best choice. A noticeable drop in efficiency appears for a fixed value $\varphi = 1$, however, randomization around 1 mitigates the effect of complete momentum update.

We believe that a random value around 0.5 drawn for every MC iteration is a safe initial guess for a good choice of the parameter $\varphi$. Finally, we note that different values of $\varphi$ can be assigned to different variates – those that require longer trajectories to decorrelate could have assigned smaller values of $\varphi$ and those that do not, can use bigger values.

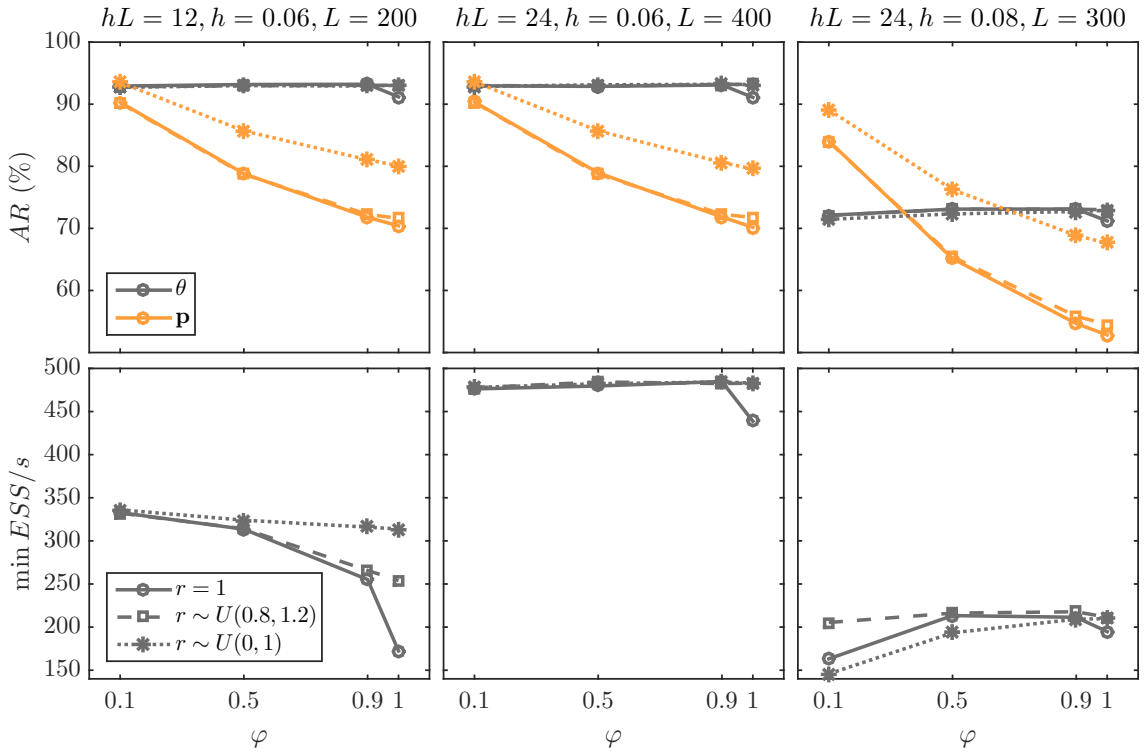


Figure 3.13: Position and momenta acceptance rates and time-normalized minimum ESS obtained in sampling from a 100-dimensional Gaussian distribution using MMHMC with the noise parameter set as $r\varphi$, resulting in fixed values of $\varphi$ for every MC iteration and two randomizing schemes.

The decision on the order of modified Hamiltonian is not a problematic one. Our experiments indicate that the 4th order modified Hamiltonian combined with the new integrators performs just well. For more complex models, if the acceptance rate is low with the 4th order and one wish to maintain the trajectory length $hL$, the 6th order modified Hamiltonian might be needed. This comes at a higher computational cost; however, such complex models might require large values of $L$ for which the computational overhead due to the calculation of modified Hamiltonian becomes negligible.

## 3.3 Summary

In this chapter, we introduced the Mix & Match Hamiltonian Monte Carlo method, an alternative to HMC for efficient sampling in computational statistics. It is based on the GSHMC method by Akhmatskaya and Reich (2008) designed for molecular simulation but has been modified, enriched with new features and adapted specifically to computational statistics. The MMHMC method can be defined as a generalized HMC importance sampler. It offers an update of momentum in a general form and samples from a modified distribution that is determined through modified Hamiltonians.

In Section 3.2.1 we have provided new formulations of modified Hamiltonians of 4th and 6th order for the splitting integrating schemes, which include families of two-, three- and four-stage integrators, recently proposed in the literature for improving the accuracy of numerical integration. The newly derived modified Hamiltonians are defined either through analytical derivatives of the potential function or numerical time derivatives of its gradient, which are computed from the quantities accessible during the simulation. We consider the former formulation being appropriate for sparse Hessian matrices of the potential and the latter, although including additional integration steps, being beneficial for cases where higher order derivatives are computationally demanding.

The novel numerical integrators from the two- and three-stage families of splitting integrators and specific to sampling with modified Hamiltonians have been derived in Section 3.2.2. We have designed new integrators by minimizing either error in modified Hamiltonian introduced due to numerical integration or its expected value, taken with respect to the modified density. With a high dimensional Gaussian model problem, two-stage integrators demonstrate a remarkable improvement over Verlet, both in terms of acceptance rates and sampling efficiency. Moreover, the improvement increases with dimension and comes at no additional computational cost. Our recommendation is to use the new two-stage integrators instead of Verlet for high dimensional problems.

In Section 3.2.3 we have proposed a computationally effective Metropolis test for momentum update and show that its use can potentially reduce computational time by 60%. In addition, different alternative strategies for momentum update, including a transformation of momenta variables and several repetitive momentum update schemes have been investigated. We have implemented, tested and analyzed these strategies but have not found any benefit from these formulations whatsoever.

In Section 3.2.4 we have adapted the reduced momenta flipping technique (Wagoner and Pande, 2012) to MMHMC, which potentially can improve sampling. Nevertheless, the tested models did not reveal a significant improvement in sampling efficiency of MMHMC with the use of this methodology.

We provide the summary for the MMHMC method using Hessian of the potential function and numerical time derivatives of its gradient in Algorithms 5 and 6, respectively. Both algorithms are formulated for the case of the 4th order modified Hamiltonian.

**Algorithm 5** MMHMC using Hessian of the potential function

**Input:** $N$: number of Monte Carlo samples
$p(h)$: step size randomization policy
$p(L)$: number of integration steps randomization policy
$p(\varphi)$: noise parameter randomization policy
$M$: mass matrix
$r$: number of stages in the numerical integrator ($r = 1, 2, 3, 4$)
$\Psi_{h,L}$: symplectic $r$-stage numerical integrator

Initialize $(\boldsymbol{\theta}^0, \mathbf{p}^0)$

Calculate Hessian $U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}^0)$

**for** $n = 1, \ldots, N$ **do**

Draw $h_n \sim p(h), L_n \sim p(L), \varphi_n \sim p(\varphi)$

$(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}^{n-1}, \mathbf{p}^{n-1})$

PMMC step

Draw noise $\mathbf{u} \sim \mathcal{N}(0, M)$

Update momenta

$$\bar{\mathbf{p}} = \begin{cases} \sqrt{1-\varphi_n}\mathbf{p} + \sqrt{\varphi_n}\mathbf{u} & \text{with probability } \mathcal{P} = \min\{1, \exp(-\Delta\hat{H})\} \\ \mathbf{p} & \text{otherwise} \end{cases}$$

$\Delta\hat{H}$ defined in (3.45)–(3.46)

Calculate modified Hamiltonian $\tilde{H}^{[4]}(\boldsymbol{\theta}, \bar{\mathbf{p}})$ defined in (3.5)

HDMC step

Generate a proposal by integrating Hamiltonian dynamics with step size $h_n$ over $L_n$ steps

$$(\boldsymbol{\theta}', \mathbf{p}') = \Psi_{h_n, L_n}(\boldsymbol{\theta}, \bar{\mathbf{p}})$$

Calculate Hessian $U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}')$ and modified Hamiltonian $\tilde{H}^{[4]}(\boldsymbol{\theta}', \mathbf{p}')$

Calculate acceptance probability

$$\alpha = \min\left\{1, \exp\left(-(\tilde{H}^{[4]}(\boldsymbol{\theta}', \mathbf{p}') - \tilde{H}^{[4]}(\boldsymbol{\theta}, \bar{\mathbf{p}})\right)\right\}$$

Metropolis test

$$(\boldsymbol{\theta}^n, \mathbf{p}^n) = \begin{cases} (\boldsymbol{\theta}', \mathbf{p}') & \text{accept with probability } \alpha \\ \mathcal{F}(\boldsymbol{\theta}, \bar{\mathbf{p}}) & \text{reject otherwise} \end{cases}$$

$$\mathcal{F}(\boldsymbol{\theta}, \mathbf{p}) = \begin{cases} (\boldsymbol{\theta}, -\mathbf{p}) \\ \text{reduced flip (optionally)} \end{cases}$$

Compute weight

$$w_n = \exp\left(\tilde{H}^{[4]}(\boldsymbol{\theta}^n, \mathbf{p}^n) - H(\boldsymbol{\theta}^n, \mathbf{p}^n)\right)$$

**end for**

Estimate integral (1.3) as

$$\hat{I} = \frac{\sum_{n=1}^{N} f(\boldsymbol{\theta}^n) w_n}{\sum_{n=1}^{N} w_n}$$

**Algorithm 6** MMHMC using numerical derivatives of the gradient of the potential

**Input:** $N$: number of Monte Carlo samples
$p(h)$: step size randomization policy
$p(L)$: number of integration steps randomization policy
$p(\varphi)$: noise parameter randomization policy
$M$: mass matrix
$r$: number of stages in the numerical integrator ($r = 1, 2, 3, 4$)
$\Psi_{h,L}$: symplectic $r$-stage numerical integrator

Initialize $(\boldsymbol{\theta}^0, \mathbf{p}^0)$

Integrate one stage (i.e. one gradient calculation) backward ($\Psi_{h,-1}(\boldsymbol{\theta}^0, \mathbf{p}^0)$) and forward ($\Psi_{h,1}(\boldsymbol{\theta}^0, \mathbf{p}^0)$)

Calculate scaled time derivative of the gradient $P_1$ using (3.21)

**for** $n = 1, \ldots, N$ **do**

  Draw $h_n \sim p(h), L_n \sim p(L), \varphi_n \sim p(\varphi)$

  $(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}^{n-1}, \mathbf{p}^{n-1})$

  PMMC step

  Draw noise $\mathbf{u} \sim \mathcal{N}(0, M)$

  Propose momenta

$$\mathbf{p}^* = \sqrt{1 - \varphi_n}\mathbf{p} + \sqrt{\varphi_n}\mathbf{u}$$

  Integrate one stage backward ($\Psi_{h_n,-1}(\boldsymbol{\theta}, \mathbf{p}^*)$) and forward ($\Psi_{h_n,1}(\boldsymbol{\theta}, \mathbf{p}^*)$)

  Calculate the resulting scaled time derivative of the gradient $P_1^*$

  Update momenta

$$\bar{\mathbf{p}} = \begin{cases} \mathbf{p}^* & \text{with probability } \mathcal{P} = \min\{1, \exp(-\Delta\hat{H})\} \\ \mathbf{p} & \text{otherwise} \end{cases}$$

  $\Delta\hat{H}$ defined in (3.49)

  Calculate modified Hamiltonian $\tilde{H}^{[4]}(\boldsymbol{\theta}, \bar{\mathbf{p}})$ defined in (3.22)

  HDMC step

  Integrate Hamiltonian dynamics with step size $h_n$ over $L_n^+$ steps {$^+$ stands for an additional forward integration}

  Assign a proposal

$$(\boldsymbol{\theta}', \mathbf{p}') = \Psi_{h_n,L_n}(\boldsymbol{\theta}, \bar{\mathbf{p}})$$

  Calculate the resulting scaled time derivative of the gradient $P_1'$

  Calculate modified Hamiltonian $\tilde{H}^{[4]}(\boldsymbol{\theta}', \mathbf{p}')$

  Calculate acceptance probability

$$\alpha = \min\left\{1, \exp\left(-(\tilde{H}^{[4]}(\boldsymbol{\theta}', \mathbf{p}') - \tilde{H}^{[4]}(\boldsymbol{\theta}, \bar{\mathbf{p}})\right)\right\}$$

  Metropolis test

$$(\boldsymbol{\theta}^n, \mathbf{p}^n) = \begin{cases} (\boldsymbol{\theta}', \mathbf{p}') & \text{accept with probability } \alpha \\ \mathcal{F}(\boldsymbol{\theta}, \bar{\mathbf{p}}) & \text{reject otherwise} \end{cases}$$

$$\mathcal{F}(\boldsymbol{\theta}, \mathbf{p}) = \begin{cases} (\boldsymbol{\theta}, -\mathbf{p}) \\ \text{reduced flip (optionally)} \end{cases}$$

20: Compute weight

$$w_n = \exp\left(\tilde{H}^{[4]}(\boldsymbol{\theta}^n, \mathbf{p}^n) - H(\boldsymbol{\theta}^n, \mathbf{p}^n)\right)$$

21: **end for**
22: Estimate integral (1.3) as

$$\hat{I} = \frac{\sum_{n=1}^{N} f(\boldsymbol{\theta}^n) w_n}{\sum_{n=1}^{N} w_n}$$

---

Considering ideas used for designing the MMHMC method, one could expect its advantages over HMC originating from: (i) higher acceptance rates (due to better conservation of modified Hamiltonians by symplectic integrators than true Hamiltonian); (ii) access to second-order information about the target distribution and (iii) an extra parameter for improving the performance. These advantages come with an expense in terms of (i) a reduced efficiency of an estimator of the integral (1.3) due to importance sampling and (ii) a higher computational cost, consisting of the computation of modified Hamiltonian for each proposal (higher orders being even more expensive) and extra Metropolis test for momentum update. In Chapter 6 we examine the performance of MMHMC on various benchmark models and answer the question of whether MMHMC emerges as a competitor to HMC, a method which is rather successful in computational statistics.

In the next chapter, we introduce some extensions to the MMHMC method. In particular, we formulate a parallel tempering algorithm for efficient multimodal sampling that utilizes MMHMC as an underlying sampler. An algorithm for Bayesian adaptation of MMHMC parameters is also proposed. In addition, we discuss the estimation of the marginal likelihood using MMHMC and formulate sampling of constrained parameters in the context of the MMHMC method.

# 4

# Extensions of Mix & Match Hamiltonian Monte Carlo

This chapter introduces the important extensions of the MMHMC method which make possible a use of MMHMC in a wide range of applications. We formulate an approach for sampling of a certain class of constrained parameters using MMHMC in Section 4.1. Two algorithms for Bayesian adaptation of MMHMC simulation parameters are formulated in Section 3.2.5, and the Parallel Tempering MMHMC method is devised in Section 4.3. Estimation of the marginal likelihood using MMHMC as the underlying method is discussed in Section 4.4.

## 4.1 Sampling constrained parameters using MMHMC

Similar to HMC, the MMHMC method has been designed to sample unconstrained parameters with respect to which the posterior distribution is differentiable almost everywhere. Some simple constraints, like nonnegativity, lower or upper bounds, can be dealt with an appropriate transformation of variables. Examples of such constraints and transformations suitable for HMC sampling are listed in Stan manual (Stan Development Team, 2016).

Here we formulate MMHMC for sampling constrained parameters using a transformation of variables.

We consider transformation $\mathcal{T}$ that is a bijection, monotonic and such that the inverse transformation $\mathcal{T}^{-1}$ is differentiable. If $\pi(\cdot)$ is the p.d.f. of the random variable $\boldsymbol{\theta}$, then the p.d.f. of the random variable $\boldsymbol{\psi} = \mathcal{T}(\boldsymbol{\theta})$ is

$$\bar{\pi}(\boldsymbol{\psi}) = \pi(\mathcal{T}^{-1}(\boldsymbol{\psi}))|\det(\mathcal{J}_{\mathcal{T}^{-1}})|, \tag{4.1}$$

where $\mathcal{J}_{\mathcal{T}^{-1}}$ is the Jacobian of the transformation $\mathcal{T}^{-1}$ at $\boldsymbol{\psi}$ defined as

$$\mathcal{J}_{\mathcal{T}^{-1}} = \begin{bmatrix} \frac{\partial \theta_1}{\partial \psi_1} & \cdots & \frac{\partial \theta_1}{\partial \psi_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial \theta_D}{\partial \psi_1} & \cdots & \frac{\partial \theta_D}{\partial \psi_D} \end{bmatrix}.$$

The absolute determinant of the Jacobian accounts for the differential change in volume in the state space due to introduced transformation.

Since the p.d.f. can be written as $\pi \propto \exp(-U)$ it follows that the potential function with respect to the unconstrained variable $\boldsymbol{\psi}$ is

$$\bar{U}(\boldsymbol{\psi}) = U(\mathcal{T}^{-1}(\boldsymbol{\psi})) - \log|\det(\mathcal{J}_{\mathcal{T}^{-1}})|.$$

Proposal states in the Markov chain are generated using Hamiltonian dynamics driven with respect to the transformed variables $\boldsymbol{\psi}$; therefore the required gradient of the potential energy is with respect to $\boldsymbol{\psi}$, i.e. $\bar{U}_{\boldsymbol{\psi}}$. If for the purpose of implementation, one wants to store original, constrained variables, the potential function computed in terms of $\boldsymbol{\theta}$ is

$$\bar{U}(\mathcal{T}(\boldsymbol{\theta})) = U(\boldsymbol{\theta}) - \log(|\det(\mathcal{J}_{\mathcal{T}^{-1}})|)$$

and the gradient is

$$\bar{U}_{\mathcal{T}(\boldsymbol{\theta})} = \bar{U}_{\boldsymbol{\theta}} \mathcal{J}_{\mathcal{T}^{-1}}.$$

The Hamiltonian function for the unconstrained parameters is then defined as

$$\bar{H} = H \circ \mathcal{T}^{-1} - \log|\det(\mathcal{J}_{\mathcal{T}^{-1}})|,$$

as also noted by Fang et al. (2014), and the target joint density of unconstrained parameters and momenta variables is

$$\bar{\pi}(\boldsymbol{\psi}, \mathbf{p}) = \bar{\pi}(\boldsymbol{\psi})\pi(\mathbf{p}) \propto \exp(-\bar{H}(\boldsymbol{\psi}, \mathbf{p})).$$

The MMHMC method draws samples with respect to the modified density

$$\tilde{\bar{\pi}}(\boldsymbol{\psi}, \mathbf{p}) \propto \exp(-\tilde{\bar{H}}(\boldsymbol{\psi}, \mathbf{p}))$$

with modified Hamiltonian defined as

$$\tilde{\bar{H}}(\boldsymbol{\psi}, \mathbf{p}) = \bar{U}(\boldsymbol{\psi}) + K(\mathbf{p}) + h^2\Big(c_{21}\mathbf{p}^\intercal M^{-1}\bar{U}_{\boldsymbol{\psi}\boldsymbol{\psi}}(\boldsymbol{\psi})M^{-1}\mathbf{p} + c_{22}\bar{U}_{\boldsymbol{\psi}}(\boldsymbol{\psi})^\intercal M^{-1}\bar{U}_{\boldsymbol{\psi}}(\boldsymbol{\psi})\Big).$$

**Estimating expected values.** The expected value with respect to probability distribution $\pi$ of the function $f$ of constrained parameters $\boldsymbol{\theta}$ is equivalent to the expected value with respect to distribution $\bar{\pi}$ of $f$ of unconstrained parameters $\boldsymbol{\phi}$. This follows from simple

change of variables and Equation (4.1) as

$$\begin{aligned}\mathbb{E}_\pi[f] &= \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta},\mathbf{p})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p} = \int f(\mathcal{T}^{-1}(\boldsymbol{\psi}))\pi(\mathcal{T}^{-1}(\boldsymbol{\psi}),\mathbf{p})|\det(\mathcal{J}_{\mathcal{T}^{-1}})|\mathrm{d}\boldsymbol{\psi}\mathrm{d}\mathbf{p} \\ &= \int f(\mathcal{T}^{-1}(\boldsymbol{\psi}))\bar{\pi}(\boldsymbol{\psi},\mathbf{p})\mathrm{d}\boldsymbol{\psi}\mathrm{d}\mathbf{p} = \mathbb{E}_{\bar{\pi}}\left[f\circ\mathcal{T}^{-1}\right]\end{aligned}$$

For MMHMC however, we need to take into account the importance weights and find the expected value with respect to the modified density as

$$\begin{aligned}\mathbb{E}_\pi[f] &= \int f(\mathcal{T}^{-1}(\boldsymbol{\psi}))\bar{\pi}(\boldsymbol{\psi},\mathbf{p})\mathrm{d}\boldsymbol{\psi}\mathrm{d}\mathbf{p} = \int f(\mathcal{T}^{-1}(\boldsymbol{\psi}))\frac{\bar{\pi}(\boldsymbol{\psi},\mathbf{p})}{\tilde{\bar{\pi}}(\boldsymbol{\psi},\mathbf{p})}\tilde{\bar{\pi}}(\boldsymbol{\psi},\mathbf{p})\mathrm{d}\boldsymbol{\psi}\mathrm{d}\mathbf{p} \\ &= \int f(\mathcal{T}^{-1}(\boldsymbol{\psi}))w(\boldsymbol{\psi},\mathbf{p})\tilde{\bar{\pi}}(\boldsymbol{\psi},\mathbf{p})\mathrm{d}\boldsymbol{\psi}\mathrm{d}\mathbf{p} \\ &= \mathbb{E}_{\tilde{\bar{\pi}}}\left[w\cdot f\circ\mathcal{T}^{-1}\right].\end{aligned}$$

Finally, this can be estimated by

$$\frac{\sum_{n=1}^{N} w_n f(\mathcal{T}^{-1}(\boldsymbol{\psi}^n))}{\sum_{n=1}^{N} w_n}$$

with importance weights

$$\begin{aligned}w_n &= \exp\left(\tilde{\bar{H}}(\boldsymbol{\psi}^n,\mathbf{p}^n) - \bar{H}(\boldsymbol{\psi}^n,\mathbf{p}^n)\right) \\ &= \exp\left(h^2(c_{21}(\mathbf{p}^n)^T\bar{U}_{\boldsymbol{\psi}\boldsymbol{\psi}}(\boldsymbol{\psi}^n)\mathbf{p}^n + c_{22}\bar{U}_{\boldsymbol{\psi}}(\boldsymbol{\psi}^n)^T\bar{U}_{\boldsymbol{\psi}}(\boldsymbol{\psi}^n)\right).\end{aligned}$$

## 4.2 Bayesian adaptation of MMHMC simulation parameters

We now formulate two algorithms for adaptation of MMHMC simulation parameters based on ideas from (Mahendran et al., 2012; Wang et al., 2013). In one of the approaches, we perform adaptation in a finite number of steps, prior to sampling, while in the other, adaptation is carried out on the fly, during sampling with a diminishing condition. Our aim is to replace a manual tuning of the parameters with the rational automatic adaptation in order to (i) improve sampling efficiency by locating parameters that lead to more uncorrelated samples; (ii) reduce the computational cost that some choices of parameters imply and (iii) reduce the effort of manual tuning.

Instead of finding particular fixed values of simulation parameters, we choose a Bayesian approach that provides a distribution over simulation parameters with probabilities estimated during adaptation procedure. The advantage of randomization of simulation parameters has been discussed in Section 2.2.4.

**Adaptation prior to sampling**

We employ a two-stage mechanism based on (Mahendran et al., 2012) that consists of (1) finite *adaptation*, which guarantees convergence of the Markov chain and (2) *sampling*. In the first stage, the chain is being adapted for a finite number of steps using Bayesian optimization and a randomized policy over parameter space is constructed. In the second

stage, we run the chain again with parameters randomly drawn from this policy at each MC step.

More specifically, if we denote the vector of MMHMC simulation parameters as $\boldsymbol{\lambda} = (h, L, \varphi)$, in the first stage of the algorithm each adaptation step $i = 1, \ldots, I$ consists of the following:

- Specify the number of Monte Carlo steps $N_{\text{update}}$ between a Bayesian update and a variance $\sigma_\eta^2$ of the noise in the objective function (4.2).
- Run MMHMC chain for $N_{\text{update}}$ steps with parameters $\boldsymbol{\lambda}_i \in \boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is a box constraint with lower (l) and upper (u) bounds for all parameters, i.e.

$$\boldsymbol{\Lambda} = \{(h, L, \varphi) : h \in [h_l, h_u], L \in [L_l, L_u], \varphi \in [\varphi_l, \varphi_u]\}\,.$$

- Use $N_{\text{update}}$ samples to obtain a noisy evaluation of the *objective function* $f(\cdot)$

$$z_i = f(\boldsymbol{\lambda}_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\eta^2), \tag{4.2}$$

  where a Gaussian process is a surrogate model for the true objective. In particular, we take a zero-mean Gaussian prior

$$f(\cdot) \sim GP(0, k(\cdot)),$$

  with covariance function

$$k(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_j) = \exp(-\frac{1}{2}\boldsymbol{\lambda}_i^T \Sigma^{-1} \boldsymbol{\lambda}_j)$$

  and $\Sigma$ being a diagonal matrix

$$\Sigma = \text{diag}\left([0.2(h_u - h_l)]^2; [0.2(L_u - L_l)]^2; [0.2(\varphi_u - \varphi_l)]^2\right),$$

  as suggested by Wang et al. (2013).
- Augment the data $\mathcal{D}_{1:i} = \{\mathcal{D}_{1:i-1}, (\boldsymbol{\lambda}_i, z_i)\}$.
- Update the Gaussian process mean $\mu_i(\boldsymbol{\lambda})$ and covariance function $\sigma_i^2(\boldsymbol{\lambda})$ of the posterior predictive distribution of the objective function

$$f_{i+1}|\mathcal{D}_{1:i}, \boldsymbol{\lambda} \sim \mathcal{N}(\mu_i(\boldsymbol{\lambda}), \sigma_i^2(\boldsymbol{\lambda}))$$

  such that

$$\begin{aligned} \mu_i(\boldsymbol{\lambda}) &= \mathbf{k}^T(\mathbf{K} + \sigma_\eta^2\mathbf{I})^{-1}\mathbf{z}_i \\ \sigma_i^2(\boldsymbol{\lambda}) &= k(\boldsymbol{\lambda}, \boldsymbol{\lambda}) - \mathbf{k}^T(\mathbf{K} + \sigma_\eta^2\mathbf{I})^{-1}\mathbf{k}, \end{aligned}$$

where

$$\mathbf{K} = \begin{bmatrix} k(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_1) & \dots & k(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_i) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_1) & \dots & k(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_i) \end{bmatrix}$$

and $\mathbf{k} = [k(\boldsymbol{\lambda}, \boldsymbol{\lambda}_i) \dots k(\boldsymbol{\lambda}, \boldsymbol{\lambda}_i)]^T$ and $\mathbf{z}_i = [z_1 \dots z_i]^T$.

- Find $\boldsymbol{\lambda}_{i+1}$ by maximizing an *acquisition function* $u(\boldsymbol{\lambda}|\mathcal{D}_{1:i})$ derived from the predictive distribution.

The adaptive procedure results in a Gaussian process of $I$ objective function observations $z_{1:I}$ obtained with $\boldsymbol{\lambda}_{1:I}$. We then construct a randomization policy $p(\boldsymbol{\lambda}|z_{1:I})$ over parameter space.

In the sampling stage, we run the chain again with $\boldsymbol{\lambda}$ randomly drawn from $p(\boldsymbol{\lambda}|z_{1:I})$ at each step and sample the target distribution. In this way, the final sampler consists of a mixture of $N$ transition kernels each parametrized by $\boldsymbol{\lambda}_n, n = 1, ..., N$.

Algorithm 7 summarizes the procedure of the adaptation of MMHMC parameters using the Bayesian approach performed prior to sampling. The choice of the objective and acquisition function will be discussed at the end of this section.

**Algorithm 7** Bayesian adaptation of MMHMC parameters prior to sampling

**Input:** $N_0$: number of iterations for adaptation
$N_{\text{update}}$: number of Monte Carlo steps between a Bayesian update
$N$: number of posterior samples
$\boldsymbol{\lambda}_l, \boldsymbol{\lambda}_u$: lower and upper bounds for parameters
$\boldsymbol{\lambda}_1$: initial set of parameters
$I = N_0/N_{\text{update}}$
**for** $i = 1, \dots, I$ **do**
  Run MMHMC chain for $N_{\text{update}}$ steps with parameters $\boldsymbol{\lambda}_i$
  Use $N_{\text{update}}$ samples to obtain a noisy evaluation of the objective function

$$z_i = f(\boldsymbol{\lambda}_i) + \epsilon$$

  Augment the data $\mathcal{D}_{1:i} = \{\mathcal{D}_{1:i-1}, (\boldsymbol{\lambda}_i, z_i)\}$
  Update the Gaussian process mean and covariance function
  $\boldsymbol{\lambda}_{i+1} \leftarrow \arg\max_{\boldsymbol{\lambda}} u(\boldsymbol{\lambda}|\mathcal{D}_{1:i})$
**end for**
Construct a randomized policy $p(\boldsymbol{\lambda}|z_{1:I})$ over parameter space

**Adaptation built-in sampling**

The second approach follows ideas from (Wang et al., 2013) and allows for adaptation on the fly, hence avoiding parameter traps that might occur in the finite adaptation approach. This approach introduces a parameter $p$ that ensures that the diminishing adaptation condition is satisfied and therefore ergodicity of the chain can be proved (Roberts

and Rosenthal, 2007). In this case, there is no need for running the Markov chain again, and all generated samples can be used for estimation of quantities of interest.

We set initial mean values of parameters $\bar{\boldsymbol{\lambda}}_1$ and draw $\boldsymbol{\lambda}$ for every Monte Carlo step from a chosen distribution $p(\boldsymbol{\lambda})$. The mean $\bar{\boldsymbol{\lambda}}$ of this randomization policy is being adapted using Bayesian optimization. Algorithm 8 presents our version of the on the fly Bayesian adaptation of MMHMC parameters with diminishing condition. This algorithm reduces the computational cost of the algorithm by Wang et al. (2013) for adaptation of HMC parameters, by means of calculating the objective function only when it is required for optimization. Indeed, optimization is performed with probability $p$ that is vanishing throughout the simulation, so is a number of objective function evaluations. This is not the case with the original algorithm (Wang et al., 2013) where the objective function is calculated after every $N_{\text{update}}$ Monte Carlo steps.

**Algorithm 8** Bayesian adaptation of MMHMC parameters built-in sampling

**Input:** $N_{\text{update}}$: number of Monte Carlo steps between a Bayesian update
$N$: number of Monte Carlo samples
$\bar{\boldsymbol{\lambda}}_l, \bar{\boldsymbol{\lambda}}_u$: lower and upper bounds for parameters' mean
$\bar{\boldsymbol{\lambda}}_1$: initial mean of parameters
$p(\boldsymbol{\lambda})$: probability distribution with mean $\bar{\boldsymbol{\lambda}}$ for randomization
$l \in \mathbb{N}^+$: parameter for a diminishing condition
$I = N/N_{\text{update}}$
$t = 0$ {counter of repeated $\boldsymbol{\lambda}$}
**for** $i = 1, \ldots, I$ **do**
  Run MMHMC chain for $N_{\text{update}}$ steps with parameters $\boldsymbol{\lambda}$ randomly drawn from $p(\boldsymbol{\lambda})$ with mean $\bar{\boldsymbol{\lambda}}_i$
  $p_i = (\max\{i - l + 1, 1\})^{-0.5}$
  Draw $u \sim \mathcal{U}(0, 1)$
  **if** $u < p_i$ **then**
    Use $N_{\text{update}}$ samples to obtain a noisy evaluation of the objective function

$$z_i = f(\bar{\boldsymbol{\lambda}}_i) + \epsilon$$

    Augment the data
      **if** $t = 0$
        $\mathcal{D}_{1:i} = \{\mathcal{D}_{1:i-1}, (\bar{\boldsymbol{\lambda}}_i, z_i)\}$
      **else**
        $\mathcal{D}_{1:i} = \{\mathcal{D}_{1:i-t}, \underbrace{(\bar{\boldsymbol{\lambda}}_i, z_i), \ldots, (\bar{\boldsymbol{\lambda}}_i, z_i)}_{t \text{ times}}\}$
      **end if**
    Update the Gaussian process mean and covariance function
    $\bar{\boldsymbol{\lambda}}_{i+1} \leftarrow \arg\max_{\boldsymbol{\lambda}} u(\boldsymbol{\lambda}|\mathcal{D}_{1:i})$
    $t = 0$
  **else**
    $\bar{\boldsymbol{\lambda}}_{i+1} \leftarrow \bar{\boldsymbol{\lambda}}_i$
    $t = t + 1$
  **end if**

18: **end for**

A critical aspect of the Bayesian adaptation procedure for MMHMC parameters is the choice of the objective and acquisition function. The objective function should represent some measure of performance of the sampler. Performance metrics are usually expensive and cannot be evaluated analytically, however running the sampler for some number of steps with a particular set of parameters $\boldsymbol{\lambda}$, or $\boldsymbol{\lambda}$ drawn from the same distribution $p(\boldsymbol{\lambda})$, one can obtain noisy observations of the objective function to be employed within the Bayesian adaptation.

The acquisition function uses the Gaussian process posterior mean and variance to identify areas with potentially higher objective function and areas of high variability, respectively. The acquisition function should be optimized by some method, but this is significantly easier than optimizing the original objective function.

Several different objective and acquisition functions have been proposed in the literature but we adopt the option from (Wang et al., 2013).

The **objective function** is assigned to be the expected squared jumping distance (ESJD) normalized by the number of integration steps $L$, i.e.

$$f(\boldsymbol{\lambda}) = \frac{ESJD}{\sqrt{L}},$$

where ESJD, an efficiency measure proposed by Pasarica and Gelman (2010), takes into account first-order autocorrelations and is defined as

$$ESJD(\boldsymbol{\lambda}) = \mathbb{E}_{\boldsymbol{\lambda}} \|\boldsymbol{\theta}^{n+1} - \boldsymbol{\theta}^{n}\|^2.$$

The intractable expectation is approximated by an empirical estimator from $N_{\text{update}}$ samples. Therefore, the objective function accounts for both correlation among samples and computational cost.

The **acquisition function** is defined as the Upper Confidence Bound (UCB) (Srinivas et al., 2010)

$$u(\boldsymbol{\lambda}|\mathcal{D}_{1:i}) = \mu_i(\boldsymbol{\lambda}) + p_i \beta_{i+1}^{\frac{1}{2}} \sigma_i(\boldsymbol{\lambda}),$$

where $\beta_{i+1} = 2\log(\frac{(i+1)^{7/2}\pi^2}{3\delta})$, $p_i = (\max\{i - l + 1, 1\})^{-0.5}$ ensures that the diminishing adaptation is satisfied, $l$ is an integer value for the diminishing condition and $\delta$ is set to 0.1.

## 4.3 Parallel Tempering with MMHMC

If the distribution of interest has more than one mode, as is often the case in practice, there is a risk that a Markov chain is being trapped in one of them. In order to make the chain explore all areas of high probability one can (i) use "tunnels" through barriers, as e.g. suggested by Lan et al. (2014a) or (ii) "flatten/melt down" the roughness of the

distribution by annealing or tempering. In order to improve convergence to multimodal target distributions using MMHMC sampler, we focus on the second approach, namely the parallel tempering (PT) method (Earl and Deem, 2005), in which multiple chains at different temperatures are used for exploration of the target distribution.

The PT method, also known as the exchange Monte Carlo (Hukushima and Nemoto, 1996), Metropolis-Coupled Markov Chain Monte Carlo ((MC)$^3$) (Geyer, 1991), replica-exchange Monte Carlo (Sugita and Okamoto, 1999), is originating from physics (Swendsen and Wang, 1986). The method is simulating $K$ parallel chains (replicas) whose stationary distributions of parameters $\boldsymbol{\theta}$ given data $\mathbf{y}$ are different though related and defined as

$$\pi_k(\boldsymbol{\theta}|\mathbf{y}) \propto L(\boldsymbol{\theta}|\mathbf{y})^{\beta_k} p(\boldsymbol{\theta}), \quad k = 1, ..., K, \tag{4.3}$$

where $\{\beta_k\}$ is a sequence of inverse temperatures such that $0 < \beta_1 < ... < \beta_K = 1$. The replicas are assumed to be independent. Smaller $\beta_k$ are "flattening" the rough surface of the posterior and allowing broader exploration of the space by escaping modes. On the other hand, higher values of $\beta_k$ enable the chain to accurately sample peaks of the posterior. The prior distribution of parameters $\boldsymbol{\theta}$ is recovered for $\beta = 0$ ("hot" chain), whereas $\beta = 1$ ("cool" chain) corresponds to the posterior distribution of interest. The Markov chain including all replicas is now defined with a joint density

$$\pi(\boldsymbol{\Theta}|\mathbf{y}) = \prod_{k=1}^{K} \pi_k(\boldsymbol{\theta}_{(k)}|\mathbf{y}),$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_{(1)}, \ldots, \boldsymbol{\theta}_{(K)})$ is a state of the chain. The stationary distribution (4.3) of each individual chain $k = 1, ..., K$ can be written as

$$\pi_k(\boldsymbol{\theta}_{(k)}|\mathbf{y}) \propto \mathrm{e}^{-\beta_k U_k(\boldsymbol{\theta}_{(k)})},$$

where

$$U_k(\boldsymbol{\theta}_{(k)}) = -\log L(\boldsymbol{\theta}_{(k)}|\mathbf{y}) - \frac{1}{\beta_k} \log p(\boldsymbol{\theta}_{(k)})$$

is the potential function. Following MMHMC methodology, we introduce momenta $\mathbf{P} = (\mathbf{p}_{(1)}, \ldots, \mathbf{p}_{(K)})$, $\mathbf{p}_{(k)} \sim \mathcal{N}(0, \beta_k M)$ and for each individual chain define joint density

$$\pi_k(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)}) \propto \exp\left(-\beta_k H_k(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)})\right),$$

where

$$H_k(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)}) = U_k(\boldsymbol{\theta}_{(k)}) + \frac{1}{2}\mathbf{p}_{(k)}^T M^{-1} \mathbf{p}_{(k)}$$

is the Hamiltonian function. Each MMHMC individual chain samples with respect to the modified density

$$\tilde{\pi}_k(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)}) \propto \exp\left(-\beta_k \tilde{H}_k(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)})\right), \tag{4.4}$$

with modified Hamiltonian $\tilde{H}_k$ defined in Section 3.2.1. Finally, the joint density of all MMHMC chains is defined as

$$\tilde{\pi}(\boldsymbol{\Theta}, \mathbf{P}) = \prod_{k=1}^{K} \tilde{\pi}_k(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)}). \tag{4.5}$$

**Exchange step.** Occasionally, states of individual chains with adjacent temperature levels are *exchanged*, i.e.

$$\begin{cases} (\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)}, \beta_k) & \to (\boldsymbol{\theta}_{(k+1)}, \mathbf{p}_{(k+1')}, \beta_k) \\ (\boldsymbol{\theta}_{(k+1)}, \mathbf{p}_{(k+1)}, \beta_{k+1}) & \to (\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k')}, \beta_{k+1}). \end{cases} \tag{4.6}$$

Following Sugita and Okamoto (1999), who proposed the replica-exchange method for molecular dynamics, the new momenta are defined as

$$\begin{aligned} \mathbf{p}_{(k')} &= \sqrt{\frac{\beta_k}{\beta_{k+1}}} \mathbf{p}_{(k)} \\ \mathbf{p}_{(k+1')} &= \sqrt{\frac{\beta_{k+1}}{\beta_k}} \mathbf{p}_{(k+1)}. \end{aligned}$$

Transitions (4.6) are then accepted with exchange Metropolis probability

$$\alpha_E = \min \left\{ 1, \frac{\tilde{\pi}_k(\boldsymbol{\theta}_{(k+1)}, \mathbf{p}_{(k+1')}) \tilde{\pi}_{k+1}(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k')})}{\tilde{\pi}_k(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)}) \tilde{\pi}_{k+1}(\boldsymbol{\theta}_{(k+1)}, \mathbf{p}_{(k+1)})} \right\}. \tag{4.7}$$

Exchanges between the individual chains introduce dependence among them, and they are no longer Markov. However, as the probability (4.7) ensures the detailed balance condition, the chain $\boldsymbol{\Theta}^1, \boldsymbol{\Theta}^2, \ldots$ is Markov with (4.5) as a stationary distribution. The probability (4.7) reduces to expression

$$\begin{aligned} \alpha_E = \min \Big\{ 1, \exp \Big[ \beta_k \left( \tilde{H}(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)}) - \tilde{H}(\boldsymbol{\theta}_{(k+1)}, \mathbf{p}_{(k+1')}) \right) \\ - \beta_{k+1} \left( \tilde{H}(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k')}) - \tilde{H}(\boldsymbol{\theta}_{(k+1)}, \mathbf{p}_{(k+1)}) \right) \Big] \Big\}, \end{aligned}$$

in which most of the terms either cancel out or are already computed within the MMHMC update. This implies just a minor computational overhead introduced by the exchange step.

**MMHMC update.** Between exchange steps, all chains are updated $N_{\text{update}}$ times according to a $\tilde{\pi}_k$-reversible transition at each Monte Carlo iteration. In particular, a partial momenta update is performed and accepted with probability

$$\mathcal{P} = \min \left\{ 1, \exp \left[ -\beta_k \Delta \hat{H} \right] \right\}, \tag{4.8}$$

where $\Delta \hat{H}$ is defined as (3.45) or (3.49) for the 4th order modified Hamiltonian with analytical or numerical derivatives, respectively. Then, Hamiltonian dynamics is simulated

with a symplectic splitting integrator (2.21)–(2.23) and the generated proposal is accepted according to the update probability

$$\alpha_U = \min\left\{1, \exp\left[-\beta_k\left(\tilde{H}(\boldsymbol{\theta}'_{(k)}, \mathbf{p}'_{(k)}) - \tilde{H}(\boldsymbol{\theta}_{(k)}, \mathbf{p}_{(k)})\right)\right]\right\}. \tag{4.9}$$

The algorithm of parallel tempering with MMHMC is summarized below.

**Algorithm 9** Parallel Tempering with MMHMC

**Input:** $K$: number of replicas
  $\{\beta_k\}$: set of inverse temperatures
  $N_{\text{update}}$: number of Monte Carlo iterations between each exchange
  $N$: total number of Monte Carlo iterations
  $\boldsymbol{\lambda} = (h, L, \varphi)$: MMHMC simulation parameters
**for** $k = 1$ to $K$ **do** {in parallel}
  Initialize $\boldsymbol{\theta}^0_{(k)}, \mathbf{p}^0_{(k)}$
**end for**
$t = 0$
**while** $t < N - N_{\text{update}}$ **do**
  **for** $k = 1$ to $K$ **do** {in parallel}
    **for** $s = 1$ to $N_{\text{update}}$ **do**
      Perform a partial momenta update and accept with probability (4.8)
      Propose a state $(\boldsymbol{\theta}'_{(k)}, \mathbf{p}'_{(k)})$ by integrating Hamiltonian equations
      Assign

$$(\boldsymbol{\theta}^{t+s}_{(k)}, \mathbf{p}^{t+s}_{(k)}) = \begin{cases} (\boldsymbol{\theta}'_{(k)}, \mathbf{p}'_{(k)}) & \text{with probability (4.9)} \\ (\boldsymbol{\theta}^{t+s-1}_{(k)}, \mathcal{F}(\mathbf{p}^{t+s-1}_{(k)})) & \text{otherwise.} \end{cases}$$

    **end for**
  **end for**
  $t = t + N_{\text{update}}$
  draw $k$ from $\{1, ..., K-1\}$
  exchange $(\boldsymbol{\theta}^t_{(k)}, \mathbf{p}^t_{(k)})$ and $(\boldsymbol{\theta}^t_{(k+1)}, \mathbf{p}^t_{(k+1)})$ with probability (4.7)
**end while**

**Estimation of expected values.** If the ultimate goal is the estimation of integral (1.3), i.e. sampling from the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$, then the auxiliary chains serve only for bridging towards the target distribution. Integral (1.3) is then estimated from the marginal chain $\{\boldsymbol{\theta}^n_{(K)}, \mathbf{p}^n_{(K)}\}_{n=1}^N$ with inverse temperature $\beta_K = 1$ and importance weights

$$w_n = \exp\left(\tilde{H}(\boldsymbol{\theta}_{(K)}, \mathbf{p}_{(K)}) - H(\boldsymbol{\theta}_{(K)}, \mathbf{p}_{(K)})\right).$$

On the other hand, samples from intermediary distributions $\pi_k(\boldsymbol{\theta}|\mathbf{y})$ can be employed for an efficient estimation of the marginal likelihood, which we discuss in Section 4.4.

### 4.3.1 Choice of parameters

The coupling of MMHMC chains may make all of the chains mix faster than any of them individually by making the entire method more than $1/K$ times more efficient than a single-chain simulation. The overall performance of the method is highly affected by efficiency of (i) parallel implementation and (ii) a choice of simulation parameters, which besides the MMHMC parameters include the following:

**Number of replicas ($K$)** Each replica requires either a processing unit or additional computational time if the simulation is not parallelized. Nevertheless, the total number of replicas should be sufficiently large to ensure that exchanges occur between all neighboring replicas.

**Sequence of temperatures $\{\beta_k\}$** The choice on temperature schedule is vital to any parallel tempering method. The spacing of temperatures affects the acceptance rate of swaps. Temperature values that are too distant result in low acceptance but values that are too close cause significant overlapping of distributions, and so weak improvement in sample variability. The distributions on adjacent temperature levels, however, should have fair overlap so as to ensure a reasonable acceptance rate.

The lowest $\beta_k$ should be low enough to enable chain escaping from local minima and exploring the space. This way some unlikely reachable states in "cool" chains can be obtained from warmer chains by crossing regions of low probability.

In order to take advantage of all replicas for sampling the target distribution, states should be exchanged between the "warmest" and "coolest" chain after traveling through the intermediate ones. One of the criteria to achieve this is to ensure that each chain spends the same amount of time at each temperature level (Earl and Deem, 2005). Accordingly, several techniques were proposed by achieving uniform acceptance rates across all chains. A common choice, for which asymptotical optimality results can be found in (Predescu et al., 2004), is to space temperatures geometrically, i.e. so that $\beta_k/\beta_{k+1}$ is a constant.

Instead of considering average acceptance rates, in the method called feedback-optimized parallel tempering (FOPT) (Katzgraber et al., 2006), the authors propose to optimize the sequence of temperatures by analyzing the round-trip times between the lowest and highest temperature chain.

Hamze et al. (2010) further improved FOPT by proposing a technique for optimizing the initial choice of both the number of replicas and values of temperatures. In this case, the number of processing units required for parallelization is not known in advance.

**Frequency of swap steps** The swap steps may be performed at every Monte Carlo iteration or after a given number of iterations ($N_{\text{update}}$) chosen randomly or as fixed values. Due to longer correlation times, "cool" chains might require more computational effort either as a larger number of Monte Carlo iterations $N_{\text{update}}$ or larger number of integration steps

$L$. Nevertheless, different values across chains would cause unsynchronized parallelization tasks.

## 4.4 Marginal likelihood estimation with MMHMC

In practice, instead of considering a single statistical model, very often the practitioner is assuming several models which could have plausibly generated the data to be analyzed. Bayesian framework offers a principled way for comparison of different models through the calculation of Bayes factors (see Chapter 1). The crucial component of a Bayes factor is the marginal likelihood of the data $\mathbf{y}$ for a given model $m$ with associated parameters $\boldsymbol{\theta}_m$ defined as

$$p(\mathbf{y}|m) = \int_{\boldsymbol{\theta}_m} L(\boldsymbol{\theta}_m|\mathbf{y})p(\boldsymbol{\theta}_m)\mathrm{d}\boldsymbol{\theta}_m,$$

where $L(\boldsymbol{\theta}_m|\mathbf{y})$ is the likelihood of the data under model $m$ with parameters $\boldsymbol{\theta}_m$ and $p(\boldsymbol{\theta}_m)$ is the prior on the parameters in model $m$. Throughout the rest of this Section, conditioning on a particular model $m$ is omitted in notation to improve readability. The marginal likelihood is also known as the integrated likelihood, model evidence, normalizing constant or partition function.

For the most statistical models of interest, however, the marginal likelihood is analytically intractable, as it involves a high-dimensional integration over a complex function, and therefore, it must be approximated. This task, however, is not trivial.

A number of different approaches for estimation of the marginal likelihood have been proposed in both statistics and physics literature (where the problem is known as the free-energy calculation), sometimes independently and in parallel. For a comprehensive review, we refer the reader to e.g. (Friel and Wyse, 2012) and references therein. For example, the harmonic mean estimator (Newton and Raftery, 1994) can be seen as an importance sampling (IS) estimator which is easy to implement. IS approach estimates the marginal likelihood simply through the average of the importance weights. Nevertheless, the estimate is usually biased if the chain is not mixing well, due to an inefficient underlying sampler. This becomes more problematic for multimodal distributions.

A more sophisticated and successful IS was proposed by Neal (2001), who combined importance sampling with simulated annealing, leading to the Annealed Importance Sampling (AIS) method. An extension of AIS for the estimation of the marginal likelihood using HMC approach was developed by Sohl-Dickstein and Culpepper (2012).

Another popular technique makes use of the power posteriors (Friel and Pettitt, 2008), defined as in Equation (4.3). It is based on ideas of thermodynamic integration, which was first developed in statistical physics (Frenkel, 1986) and later extended for statistics within the path sampling approach (Gelman and Meng, 1998). Path sampling methods involve sampling from a sequence of power posteriors, i.e. distributions which connect the prior to the posterior distribution using a power of the likelihood. An estimator of the marginal likelihood is then obtained by integrating over these distributions.

For each of these approaches, the choice of the underlying sampler is crucial. MMHMC, being an efficient sampler, can be used as a base sampler within path sampling or annealing schemes to provide low variance estimates of the marginal likelihoods for model comparison.

For example, the parallel tempering MMHMC (PT-MMHMC) method, devised in the previous Section, can be readily combined with path sampling in the following way. PT-MMHMC draws samples from $K$ modified distributions $\tilde{\pi}_k$ with different temperatures $\beta_k, k = 1, \ldots, K$, as defined in Equation (4.4). We can then estimate the marginal likelihood by making use of the standard thermodynamic identity

$$\log p(\mathbf{y}) = \int_0^1 \mathbb{E}_{\pi_k} \left[ \log L(\boldsymbol{\theta}_{(k)} | \mathbf{y}) \right] \mathrm{d}\beta. \tag{4.10}$$

The expectation at each temperature $\beta_k$ is calculated as

$$\mathbb{E}_{\pi_k} \left[ \log L(\boldsymbol{\theta}_{(k)} | \mathbf{y}) \right] = \mathbb{E}_{\tilde{\pi}_k} \left[ w \cdot \log L(\boldsymbol{\theta}_{(k)} | \mathbf{y}) \right] \equiv E_k$$

due to importance sampling of MMHMC, as follows from (3.2), where $w$ is the importance weight function. We employ power posterior samples $\{\boldsymbol{\theta}^n_{(k)}\}_{n=1}^N$, drawn from the modified density $\tilde{\pi}_k$, to find Monte Carlo estimates $\hat{E}_k, k = 1, \ldots, K$ of expectations at each discrete temperature. Then a trapezoidal rule we can be used to approximate the marginal likelihood as

$$\log p(\mathbf{y}) \approx \sum_{k=1}^{K-1} (\beta_{k+1} - \beta_k) \frac{\hat{E}_{k+1} + \hat{E}_k}{2}.$$

However, this estimation introduces two sources of error. One of them appears due to discretization of temperature in the thermodynamic integral (4.10). Calderhead and Girolami (2009) characterized this error in terms of the Kullback-Leibler divergences between successive tempered distributions. Therefore, the optimal spacing of temperatures $\beta_k$, in terms of minimizing this source of error, should be chosen such that the Kullback-Liebler distances are minimized. The other source of error is the Monte Carlo error introduced from estimating the power posterior expectations $E_k$. Nevertheless, with MMHMC we expect this error to be reduced compared with the one resulting from a less sophisticated sampler.

## 4.5 Summary

In this chapter, we have introduced several extensions of the MMHMC method, which make it applicable to a broad range of problems. First, we have adapted MMHMC to sampling of constrained variables, by defining all quantities that take part in the calculation of modified Hamiltonians accounting for the transformation in the parameter space. In order to reduce the efforts of manual tuning of MMHMC simulation parameters, we then have devised two algorithms for automatic adaptation using Bayesian optimization approach. Also in Section 4.3 we have formulated the parallel tempering MMHMC method. The benefits of this method are twofold. Firstly, due to the use of an ensemble of chains it improves mixing

and enables sampling from the multimodal probability distributions. Secondly, it provides samples from all required power posteriors simultaneously, which then can be used for estimation of the marginal likelihood, as described in Section 4.4.

# 5

# Implementation

This chapter provides a description of the software package HaiCS (**Ha**miltonians **i**n **C**omputational **S**tatistics), developed along this dissertation within the Modelling and Simulation in Life and Materials Sciences group in the Basque Center for Applied Mathematics (BCAM).

There exist several open-source software packages with implemented Hamiltonian Monte Carlo (HMC) based methodologies, such as Stan (Stan Development Team, 2016), PyMC3 (Salvatier et al., 2016), LaplacesDemon (Statisticat LLC, 2013). None of them however includes implementation of modified Hamiltonians. Instead of implementing MMHMC in one of those packages, we decided to develop the in-house package from scratches in order to achieve (i) flexibility in methodology development and testing, and (ii) control over code performance and optimization.

## 5.1 Description

The HaiCS package is developed for statistical sampling of high dimensional and complex distributions and parameter estimation in different models through Bayesian inference using HMC based methods. The currently implemented models include multivariate Gaussian distribution, Bayesian Logistic Regression, and Stochastic Volatility, but new (hierarchical) models can be readily introduced through a template file for model implementation. Different existing and recently developed numerical integrators, strategies for momenta update and flips are available in the package for being employed within the HMC, Generalized Hamiltonian Monte Carlo (GHMC), Metropolis Adjusted Langevin Algorithm (MALA), second order Langevin Monte Carlo (L2MC), Mix & Match Hamiltonian Monte Carlo (MMHMC) methods and its variants. The package is suited for output analysis in

CODA (Plummer et al., 2006) – a widely used R toolkit for Markov Chain Monte Carlo (MCMC) diagnostics.

The HaiCS package is summarized below.

---

**Package summary**

*Package title:* Hamiltonians in Computational Statistics (HaiCS)
*No. of lines in the source code:* 8790
*Core programming language:* C
*Operating system:* UNIX certified (e.g. GNU/LINUX, OS X)
*RAM:* Dependable on application
*External libraries:* CBLAS, GSL
*Sampling engines:* HMC, MALA, GHMC, L2MC, MMHMC
*Benchmark models:* Multivariate Gaussian distribution, Bayesian Logistic Regression, Stochastic Volatility

---

The core functionality of HaiCS, namely performing statistical sampling, is implemented in C programming language and consists of 22 files. In addition, the package includes a template input file and 18 Bash and R scripts for running simulations or output data analysis. The HaiCS workflow is depicted in Figure 5.1.

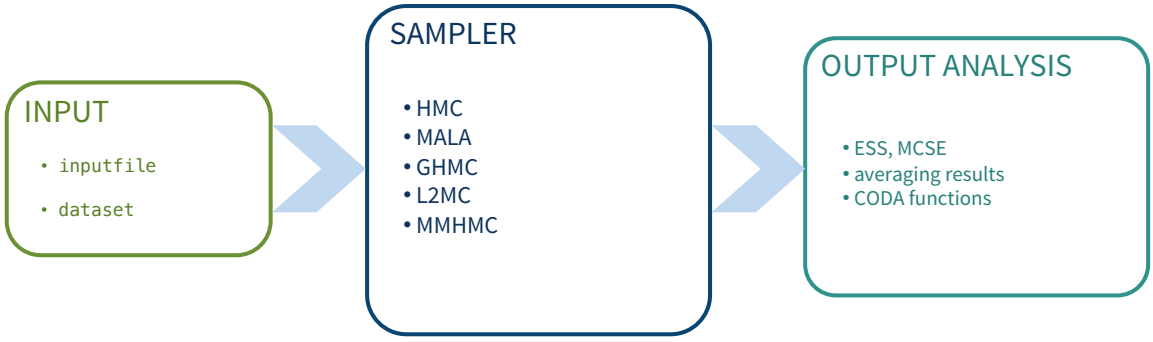


Figure 5.1: HaiCS workflow.

### 5.1.1 Structure of SAMPLER module

Subroutine dependencies of the HaiCS module SAMPLER are shown in Figure 5.2. All source files are located in the `HAICS/src/` directory. Below we list the main source files and explain their functions.

#### 5.1.1.1 Subroutine specification

**`main.c`** The main program. Passes the simulation `ID` (an argument given by a user at the time of execution) to the `read_input` subroutine for reading all input data. Calls the `hmc` subroutine for performing sampling.

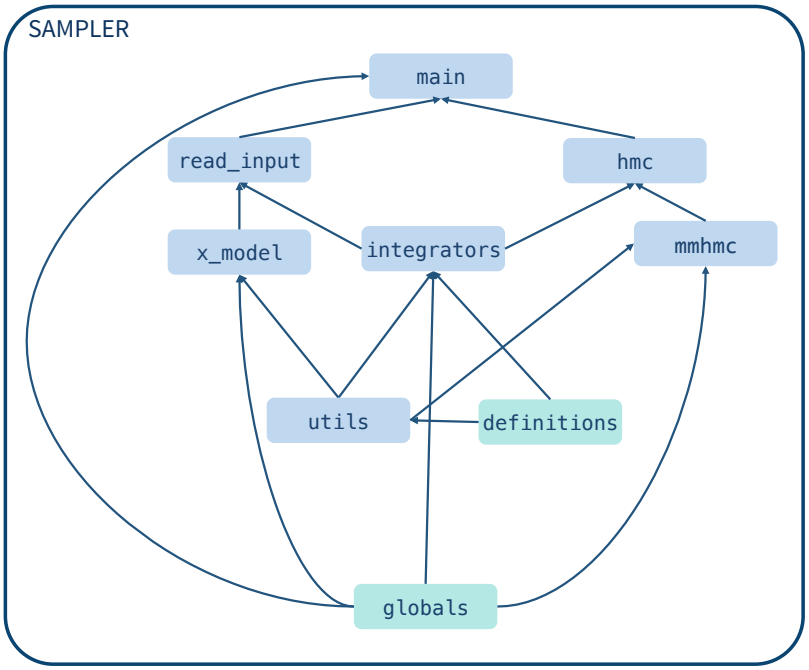


Figure 5.2: Structure of the HaiCS sampling module.

**`read_input.c`**

- According to the simulation `ID`, opens and reads the corresponding input file `HAICS/output/{ID}/inputfile{ID}`.
- Assigns all simulation parameters for all Monte Carlo iterations, according to `inputfile{ID}`.
- Assigns integrator(s)
- Defines the model.
- Prepares files for storing output data in the directory `HAICS/output/{ID}`.

**`x_model.c`** Contains model specific functions.

- For each model (`x={GD,BLR,SV}`), assigns functions for the logarithm of the prior, likelihood, as well as the gradient and Hessian (optional) of the log posterior, and the determinant of the Jacobian (optional).
- Reads input dataset from the directory `HAICS/benchmarks/{model}`.
- Initializes a state of the chain according to the prior distribution.

**`hmc.c`** Updates the Markov chain.

- For hierarchical models, samples parameters and hyper parameters in two phases, allowing for different simulation parameters to be used.
- For the chosen methodology, defines and calls subroutines for
  - momentum update,

    - momentum flips,
    - Hamiltonian dynamics integration,
    - calculation of Hamiltonians,
    - Metropolis test.
- Calculates CPU time for warm-up and production phase.

**`mmhmc.c`** Provides components specific to MMHMC:

- calculation of modified Hamiltonians of different types and orders,
- calculation of finite differences,
- partial momentum Monte Carlo step,
- calculation of importance weights.

**`integrators.c`**

- Implements the Verlet, two-, three- and four-stage integrators, both velocity and position versions.
- Includes optimized versions of integrators used within MMHMC.

**`utils.c`** Offers the utility functions:

- allocation/freeing of the memory for customized data types,
- efficient matrix and tensor allocation/freeing (with consecutive elements in the memory).

**`definitions.h`** Defines customized data types, macros, constants, used along the code. Particularly useful definitions are macros for efficient handling of column major order matrix operations, required for BLAS calculations.

**`globals.h`** Defines all global variables, pointers to arrays and functions, file pointers, used within the routines.

## 5.2 External libraries

**CBLAS** BLAS (Basic Linear Algebra Subprograms) is a library for performing fundamental linear algebra operations (`http://www.netlib.org/blas/`). CBLAS is a C interface to the Fortran BLAS library (`http://www.netlib.org/blas/#_cblas`). For an optimized performance, HaiCS makes use of a number of CBLAS functions, especially for handling symmetric and banded matrices. The link to the library (`-lcblas`) is provided in `Makefile` placed in the directory `HAICS/src/`.

**GSL** The GSL (GNU Scientific Library) is a C/C++ numerical library which provides over 1000 functions covering a broad range of mathematical areas (`https://www.gnu.org/software/gsl/`). It is a part of the GNU Project, conceived in 1996 by Dr. M. Galassi and Dr. J. Theiler of Los Alamos National Laboratory (Galassi et al., 2009).

Currently, only random number generators related routines are used in HaiCS. The link to the library (`-lgsl`) is included in the provided `Makefile`.

## 5.3 Installation

Installing HaiCS involves unpacking the software and building the executable (called `haics`) from source.

*System requirements*

The current version of HaiCS is intended for computers running Unix certified operating system (e.g. Linux, OS X). It requires C compilers as well as GSL and CBLAS libraries installed.

*Unpacking the software*

The software is stored in the form of a gzip'ed tar file which contains the HaiCS source code, parameter input file template, input data for three benchmark models, as well as scripts for running the code and scripts for post-processing. The package can be unpacked by typing the following command:

```
tar -xzvf HAICS.tgz
```

This will create a top-level directory called `HAICS` and subdirectories as shown in Figure 5.3.

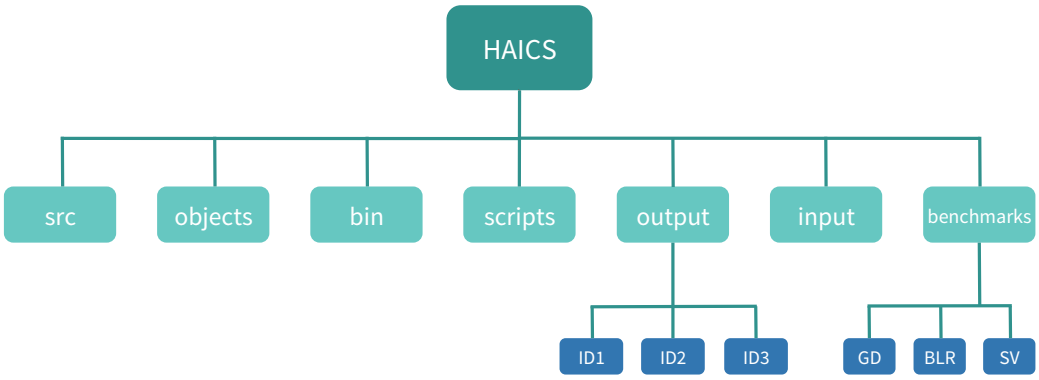


Figure 5.3: Detailed structure of the HAICS directory.

*Building `haics` executable from source*

Installation procedure is straightforward and can be successfully performed by the following steps:

1. Go to source directory by typing:

```
cd src
```

2. If necessary, change the values of environmental variables (the definition of the C compiler, `CC`, and its path, `COMP_PATH`, flags, `CFLAGS`, `IFLAGS` and `LDFLAGS`) in the `Makefile`.

3. Create the `haics` executable in the directory `HAICS/bin` by typing the command:
   `make`

## 5.4 Running HaiCS

Running HaiCS involves the following steps:

1. Set input data.
2. Execute `haics` in a single run or a sequence of runs.
3. Analyze the output data (optional).

### 5.4.1 Setting input data

Two input files are required for running calculations: a file with the dataset placed in the directory `HAICS/benchmarks/{model}/`, and a file with simulation parameters in the directory `HAICS/input/`. Columns of the dataset file correspond to variates and rows correspond to observations.

The template file for input simulation parameters, `HAICS/input/inputfile_tpl`, is self-explanatory and describes each input parameter to be specified by the user. An example is shown below.

```
#-------------------------------------------------------------------------------------------
# Input for model parameters
#-------------------------------------------------------------------------------------------
model         SV # SV; BLR; GD
data          2000 # SV - dimension; BLR - musk, sonar, secom, australian, german, heart,
   pima, ripley; GD - dimension
method        MMHMC # HMC; GHMC; MMHMC
seed          0 # 0 - shuffle seed; n - seed=n;
num_iter      30000 # total number of iterations
warm_up       10000 # number of iterations for warm-up
integrator    2S # V; 2S,3S,4S; ME2S,ME3S,ME4S
t_L           3 # type of the parameter number of int. steps: 0 - constant \
   1 - random \~U{0.8L,1.2L}  \ 2 - random ~N(L,0.0036L^2)  \ 3 - random ~U{1,L}
L             10 # number of int. steps; if type 0 - length \ 1,2 - mean \ 3 - max
t_stepsize    1 # type of the parameter step size; 0 - constant \ 1 - random ~U(0.8h,1.2h)
    \ 2 - random ~N(h,0.0036h^2)
stepsize      0.015 # step size; if type 0 - length \ 1,2 - mean
t_Phi         0 # type of the parameter phi; 0 - constant \ 1 - random ~U(0.8Phi,1.2Phi) \
    2 - random ~N(Phi,0.0025Phi^2) \ 3 - random ~U(0,Phi) \ 4 - random \~U{0.01,0.99}
Phi           0.5 # noise parameter phi, values from (0,1]; if method=HMC set to 1;
   if type 1,2 - mean \ 3 - max
flip          0 # 0 - automatic flip; 1 - reduced flip; 2 - no flip
thinning      1 # frequency of collecting posterior samples
mH            A # type of derivatives in modified Hamiltonian; A - analytical, N - numerical
mH_order      4 # order of modified Hamiltonian; 4 or 6
newPMMC       1 # 0 - no; 1 - yes
momTransf     0 # change of momenta variables; 0 - no; 1 - yes
#-------------------------------------------------------------------------------------------
```

```
# Input for latent variables
#-------------------------------------------------------------------------------------------
integrator     2S # V; 2S,3S,4S; ME2S,ME3S,ME4S
t_L_lat        3 # type of the parameter number of int. steps: 0 - constant \
   1 - random ~U(0.8L,1.2L)  \ 2 - random ~N(L,0.0025L)  \ 3 - random ~U(0,L)
L_lat          50 # number of int. steps; if 0 - length \ 1,2 - mean \ 3 - max
   t_stepsize_lat 1 # type of the parameter step size; 0 - constant \ 1 - random ~U(0.8h,1.2h)
    \ 2 - random ~N(h,0.0025h)
stepsize_lat   0.04 # step size; if 0 - length \ 1,2 - mean
t_Phi_lat      0 # type of the parameter phi; 0 - constant \ 1 - random ~U(0.8Phi,1.2Phi) \
    2 - random ~N(Phi,0.0025Phi) \ 3 - random ~U(0,Phi) \ 4 - random \~U{0.01,0.99}
Phi_lat        0.5 # noise parameter phi
```

Changes of input parameters should be made by the user in the already existing `inputfile_tpl`, without the need to copy the file elsewhere.

### 5.4.2 Executing a simulation

We recommend running a calculation in directory `HAICS/`. The script `runHAICS.sh` located in `HAICS/scripts/` directory automates calculations by typing the command

```
./scripts/runHAICS.sh {ID}
```

where `{ID}` is a chosen identification number for the simulation. The following assumptions are made in the script:

- the binary is placed in `HAICS/bin/`,
- input file `inputfile_tpl` is set and located in directory `HAICS/input/`,
- output files can be found in `HAICS/output/{ID}/` on completion of the calculation.

The run script `runHAICS.sh` does not require any tuning, editing or corrections in order to start the calculation. Provided that the input file `HAICS/input/inputfile_tpl` is prepared for calculations, `runHAICS.sh` takes care of the following steps in the following order:

- creates the directory `HAICS/output/{ID}/`,
- copies `inputfile_tpl` in `HAICS/output/{ID}/` under name `inputfile{ID}`,
- starts the simulation.

Alternatively, typing the command

```
./bin/haics {ID}
```

immediately starts the calculation and will be performed successfully if the corresponding directory and input file `HAICS/output/ID/inputfile{ID}` have been created previously.

In order to perform 10 repetitions of the same test (with same input parameters), one should use another run script by typing

```
./scripts/run10HAICS.sh {ID}
```

This will (i) create 10 different output directories in `HAICS/output/`, named `{ID}`,...,`{ID+9}` and containing corresponding input files `inputfile{ID}`,...,`inputfile{ID+9}`, and (ii) run 10 simulations.

### 5.4.3 Output data

On completion of the calculation, the following output files can be found in directory `HAICS/output/{ID}/`.

| Filename | Content |
|---|---|
| `art` | acceptance rate, CPU time (in sec.) |
| `dH(_lat)` | difference in Hamiltonians after integration, at each iteration |
| `ham(_lat)` | value of the Hamiltonian at the end of each iteration |
| `logfile{ID}` | log file |
| `logP` | logarithm of unnormalized posterior at each iteration |
| `samples(_lat)` | values of parameters at each iteration |
| `weights(_lat)` | importance weights for MMHMC |

If the binary is built with the debug option (`CFLAGS += -DDEBUG`) in `Makefile`, then a number of additional output files are created in directory `HAICS/output/{ID}/`. Those files contain detailed information at each iteration on e.g. Metropolis probability, simulation parameters, intermediate values of Hamiltonians, etc.

## 5.5 Summary

We have developed the user-friendly software package written in C HaiCS (Hamiltonians in Computational Statistics) targeted to computers running UNIX certified operating systems.

The code is intended for statistical sampling of high dimensional and complex distributions and parameter estimation in different models through Bayesian inference using Hamiltonian Monte Carlo based methods. The currently available sampling techniques include Hamiltonian Monte Carlo (HMC), Generalized Hamiltonian Monte Carlo (GHMC), Metropolis Adjusted Langevin Algorithm (MALA), second order Langevin Monte Carlo (L2MC) and Mix & Match Hamiltonian Monte Carlo (MMHMC), the method developed in this thesis.

The package benefits from efficient implementation of modified Hamiltonians, the accurate multi-stage splitting integration schemes (as previously proposed as the novel), the analysis tools compatible with CODA toolkit for MCMC diagnostics as well as the interface for implementing complex statistical models. The popular statistical models multivariate Gaussian distribution, Bayesian Logistic Regression and Stochastic Volatility are implemented in HaiCS.

# 6

# Applications

In this chapter we evaluate the performance of MMHMC method and compare it with the Random Walk Metropolis-Hastings (RWMH), Hamiltonian Monte Carlo (HMC), Generalized HMC (GHMC), Metropolis Adjusted Langevin Algorithm (MALA) and Riemann Manifold HMC (RMHMC) methods on a set of standard benchmark models used in the literature. Space exploration of an algorithm is examined on a banana-shaped distribution, while sampling efficiency is investigated on multivariate Gaussian distribution, Bayesian logistic regression model, and a stochastic volatility model. Before introducing the benchmark models and numerical results we outline measures for performance evaluation in the following section.

## 6.1 Performance evaluation

When assessing the performance of a method we focus on the following criteria:

- state **space exploration** by the chain;
- **sampling efficiency** – the ability of a method to produce more uncorrelated samples;
- **convergence** to the target distribution.

In order to evaluate these criteria we use the following metrics:

- Acceptance rate (AR);
- Effective Sample Size (ESS) and ESS normalized by the computational time in seconds (ESS/s);

- Efficiency Factor (EF) – relative ESS/s of a method with respect to ESS/s of the HMC method;
- Potential scale reduction factor ($\hat{R}$).

**Effective Sample Size** is a commonly used measure for sampling efficiency of an MCMC method. It indicates the number of effectively uncorrelated samples out of $N$ collected samples and is defined as

$$ESS = \frac{N}{1 + 2\sum_k \hat{\gamma}_k}, \tag{6.1}$$

where $\hat{\gamma}_k$ is the $k$-lag sample autocorrelation. For an estimate $\hat{I}$ of the expectation $\mathbb{E}_\pi(f)$, given by

$$\hat{I} = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{\theta}^n),$$

the effective sample size can be estimated using the initial monotone sequence variance estimator $\hat{\sigma}^2_{mono}$ of Geyer (1992) as

$$ESS = \frac{N\hat{\sigma}^2}{\hat{\sigma}^2_{mono}},$$

where $\hat{\sigma}^2 = \frac{1}{N-1}\sum_{n=1}^{N}(f(\boldsymbol{\theta}^n) - \hat{I})^2$ is the sample variance.

ESS is related to both Monte Carlo estimate of the variance $\widehat{\mathrm{Var}(\hat{I})}$ of the estimator $\hat{I}$ and Integrated Autocorrelation Time (IACT), which are two alternative measures of efficiency also used in the literature. Monte Carlo estimate of the variance of $\hat{I}$ indicates how much error is in the estimate due to the use of a Monte Carlo method. If the estimate is obtained from uncorrelated samples, its variance is given by $\hat{\sigma}^2/N$. Due to use of an MCMC method, this "naive" variance estimator has to be adjusted for autocorrelation leading to an estimate

$$\widehat{\mathrm{Var}(\hat{I})} = \frac{\hat{\sigma}^2}{ESS} = \frac{\hat{\sigma}^2_{mono}}{N}. \tag{6.2}$$

This estimator follows from the Central Limit Theorem (CLT), which states that $\sqrt{N}(\hat{I} - I) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$. It follows that the estimated variance of the estimator obtained with correlated samples is $\frac{N}{ESS} = \frac{\hat{\sigma}^2_{mono}}{\hat{\sigma}^2}$ times bigger than the estimated variance obtained with uncorrelated samples. Monte Carlo Standard Error (MCSE) is then just $\hat{\sigma}/\sqrt{ESS}$.

IACT, being the number of MC iterations needed on average for an independent sample to be drawn, is computed as

$$IACT = \frac{N}{ESS}.$$

Consequently, on average IACT correlated samples are needed to reduce the variance of the estimator by the same amount as a single uncorrelated sample.

In our experiments, we compute ESS of the mean estimator for each variate, i.e. we consider $f_i(\boldsymbol{\theta}^n) = \theta^n_i, i = 1, \ldots, D, n = 1, \ldots, N$. We report minimum, median, and maximum ESS across variates or just minimum ESS, as the most restrictive measure, calculated

using the collected posterior samples.

**Potential scale reduction factor** ($\hat{R}$) is a diagnostic for monitoring convergence of a chain to the stationary distribution (Gelman and Rubin, 1992; Brooks and Gelman, 1998). It forms a part of the CODA package (Plummer et al., 2006), developed for output analysis for MCMC methods, and is also used in Stan (Stan Development Team, 2016).

$\hat{R}$ is evaluated on $M > 1$ chains run with randomly assigned initial states until $N$ posterior samples are collected by each chain. Employing the mean of the sample variance within each chain

$$W = \frac{1}{M} \sum_{m=1}^{M} \hat{\sigma}_m^2$$

and the between-chain sample variance

$$B = \frac{N}{M-1} \sum_{m=1}^{M} \left( \bar{\theta}_m - \bar{\bar{\theta}} \right)^2,$$

calculated for each parameter $\theta_i, i = 1, \ldots, D$, where

$$\hat{\sigma}_m^2 = \frac{1}{N-1} \sum_{n=1}^{N} \left( \theta_m^n - \bar{\theta}_m \right)^2$$

and $\bar{\theta}_m$ are the variance and mean of the each chain, respectively, and $\bar{\bar{\theta}}$ is the mean of all chains combined, the sample variance $\hat{\sigma}^2$ from all chains combined is given as a weighted average of within-chain variance and between-chain variance as

$$\hat{\sigma}^2 = \left( 1 - \frac{1}{N} \right) W + \frac{1}{N} B.$$

The potential scale reduction factor for each parameter $\theta_i$ is then calculated as

$$\hat{R} = \sqrt{\frac{d+3}{d+1} \frac{\hat{V}}{W}} \tag{6.3}$$

where

$$\hat{V} = \hat{\sigma}^2 + \frac{B}{MN}$$

and $d$ is the number of degrees of freedom of a $t$-distribution with mean $\bar{\bar{\theta}}$ and variance $\hat{V}$, and is estimated by the method of moments

$$d \approx \frac{2\hat{V}^2}{\mathrm{Var}(\hat{V})}.$$

When $\hat{R}$ is high (e.g. greater than 1.1 or 1.2), the dependence on the initial point of the chain is still present, and one should increase the length of the chain to improve convergence to the stationary distribution.

This convergence diagnostic is suitable for inferences based on posterior means and variances. In case inference is determined by higher moments, other measures may be more appropriate.

We mention two alternative convergence tests proposed by Geweke (1992) and Hanson (2002). The Geweke test, also implemented in CODA package, is monitoring convergence using a single chain. The Hanson test is a convergence diagnostic specially designed for HMC and employes gradients of samples. For $i$th variate, $i = 1, \ldots, D$, the metric is computed as

$$R_i = \frac{\sum_n \left(\theta_i^n - \bar{\theta}_i^n\right)^3 \partial_{\theta_i} U(\boldsymbol{\theta}^n)}{3 \sum_n \left(\theta_i^n - \bar{\theta}_i^n\right)^2}.$$

In our experiments, however, we use only the potential scale reduction factor (6.3).

### 6.1.1 Efficiency evaluation for MMHMC

MMHMC is one of the methods which generate samples that are correlated (being an MCMC method) and weighted (being an importance sampler). Examples of such methods include SHMC (Izaguirre and Hampton, 2004), S2HMC (Sweet et al., 2009), GSHMC (Akhmatskaya and Reich, 2008), Gradient Importance Sampler (Schuster, 2015).

The estimation of the effective sample size for methods yielding correlated samples was reviewed in the previous section. For importance samplers, however, the effective sample size accounts for weighted samples and can be obtained in the following manner.

Let us assume that the function $f(\boldsymbol{\theta})$ and importance weight function $w(\boldsymbol{\theta})$ are independent under the importance probability $\tilde{\pi}(\boldsymbol{\theta})$. Variance of the mean estimator $\hat{I}$, in this case defined as the weighted average

$$\hat{I} = \frac{\sum_{n=1}^N w_n f(\boldsymbol{\theta}^n)}{\sum_{n=1}^N w_n}, \quad w_n = \frac{\pi(\boldsymbol{\theta}^n)}{\tilde{\pi}(\boldsymbol{\theta}^n)},$$

is given as

$$\mathrm{Var}_{\tilde{\pi}}(\hat{I}) = \frac{1}{N} \frac{\mathbb{E}_{\tilde{\pi}}\left[w(\boldsymbol{\theta})^2\right] \mathbb{E}_{\tilde{\pi}}\left[\left(f(\boldsymbol{\theta}) - \hat{I}\right)^2\right]}{\mathbb{E}_{\tilde{\pi}}[w(\boldsymbol{\theta})]^2}. \tag{6.4}$$

Its Monte Carlo estimate can be obtained as

$$\widehat{\mathrm{Var}(\hat{I})} = \frac{\hat{\sigma}^2}{N_e}, \tag{6.5}$$

where $\hat{\sigma}^2$ is the sample variance and

$$N_e = \frac{\left(\sum_{n=1}^N w_n\right)^2}{\sum_{n=1}^N w_n^2}$$

is the effective sample size, as first introduced by Kong et al. (1994). As noted by Neal (2001), we can write the expression (6.4) as

$$\begin{aligned} \mathrm{Var}_{\tilde{\pi}}(\hat{I}) &= \left(1 + \mathrm{Var}_{\tilde{\pi}}\left(\frac{w(\boldsymbol{\theta})}{\mathbb{E}_{\tilde{\pi}}[w(\boldsymbol{\theta})]}\right)\right)\frac{1}{N}\mathbb{E}_{\tilde{\pi}}\left[\left(f(\boldsymbol{\theta}) - \hat{I}\right)^2\right] \\ &= \left(1 + \mathrm{Var}_{\tilde{\pi}}\left(\frac{w}{\mathbb{E}_{\tilde{\pi}}[w]}\right)\right)\frac{1}{N}\mathrm{Var}_{\pi}(f(\boldsymbol{\theta})). \end{aligned} \tag{6.6}$$

We can easily see from (6.5) and (6.6) that

$$\left(1 + \mathrm{Var}_{\tilde{\pi}}\left(\frac{w}{\mathbb{E}_{\tilde{\pi}}[w]}\right)\right) \approx \frac{N}{N_e},$$

meaning that variance of the estimator obtained with weighted samples is $\frac{N}{N_e}$ times bigger than the variance obtained with unweighted samples. We also note that the effective sample size depends directly on variability in the normalized importance weights.

For the MMHMC method, the reduction in sampling efficiency due to use of importance sampling is expected to be minor. The reason for this is because the chosen importance density $\tilde{\pi}$ is a close approximation of the true density $\pi$ and therefore, normalized weights have values close to one. In case the Markov chain happens to draw samples from a region of the space in which the true Hamiltonian is not well approximated by the modified Hamiltonian, or in general, for importance sampling methods for which the importance density $\tilde{\pi}$ is not close enough to the target density $\pi$, high variability in the importance weights might occur. One should then use a metric for sampling efficiency that takes into account both correlations among samples and weights. To the best of our knowledge, a metric for samplers that generate correlated weighed samples has not been proposed, though the importance of such an objective criterion was discussed e.g. by Neal (2001) and Gramacy et al. (2010).

Here we propose a new metrics that addresses these issues. With the aim of estimating the variance from the CLT, $\sqrt{N}(\hat{I} - I) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$, we base our metric on the initial monotone sequence estimator by Geyer (1992) and weighted sample variance (Rimoldini, 2014) and covariance. More specifically, we start from an unbiased weighted sample variance

$$\hat{\sigma}_w^2 = \frac{\sum_{n=1}^N w_n}{(\sum_{n=1}^N w_n)^2 - \sum_{n=1}^N w_n^2} \sum_{n=1}^N w_n \left(f(\boldsymbol{\theta}^n) - \hat{I}\right)^2 \tag{6.7}$$

and weighted sample covariance

$$\hat{\gamma}_k = \frac{\sum_{n=1}^{N-k} \sqrt{w_n w_{n+k}}}{(\sum_{n=1}^{N-k} \sqrt{w_n w_{n+k}})^2 - \sum_{n=1}^{N-k} w_n w_{n+k}} \sum_{n=1}^{N-k} \sqrt{w_n w_{n+k}} \left(f(\boldsymbol{\theta}^n) - \hat{I}\right)\left(f(\boldsymbol{\theta}^{n+k}) - \hat{I}\right). \tag{6.8}$$

Following Geyer (1992), we define the variance estimator $\hat{\sigma}^2_{w,mono}$ as

$$\hat{\sigma}^2_{w,mono} = -\hat{\sigma}^2_w + 2\sum_{k=0}^{K} \hat{\Gamma}_k,$$

where $K$ is the largest integer such that

$$\hat{\Gamma}_k > 0, k = 0, \ldots, K,$$

and $\hat{\Gamma}_k$ are defined recursively as

$$\begin{aligned} \hat{\Gamma}_0 &= \hat{\gamma}_0 + \hat{\gamma}_1 \\ \hat{\Gamma}_k &= \min\left\{\hat{\Gamma}_{k-1}, \hat{\gamma}_{2k} + \hat{\gamma}_{2k+1}\right\}. \end{aligned}$$

Finally, we obtain the formula for estimating the effective sample size for importance sampling as

$$ESS = \frac{N\hat{\sigma}^2_w}{\hat{\sigma}^2_{w,mono}}. \tag{6.9}$$

The Monte Carlo variance of the estimator $\hat{I}$ in this case follows as

$$\widehat{\mathrm{Var}(\hat{I})} = \frac{\hat{\sigma}^2_{w,mono}}{N}. \tag{6.10}$$

## 6.2 Experimental results

The choice of the optimal simulations parameters remains an open question (Neal, 2011) and not the subject of this thesis. To make the comparison with other methods fair, we chose the following strategy. Since the stochastic volatility benchmark is studied well in literature and HMC and RMHMC were tuned previously for a particular dimension of this benchmark, we took the found set of optimal parameters as an initial guess and tuned it further. For Bayesian logistic regression and Gaussian models, especially for some data sets, such information is not available. In this case, we have located a range of reasonable parameters $L, h$ and $\varphi$ and performed the comparison for these sets. For each MC iteration we draw the number of integration steps uniformly from $\{1, \ldots, L\}$ for HMC, GHMC and MMHMC and step size uniformly from $(0.8h, 1.2h)$ for HMC, MALA, GHMC and MMHMC methods. Additionally, we tested MMHMC for a range of fixed noise parameters $\varphi$ or drawn a noise parameter uniformly from $(0, \varphi)$, but report here only results obtained with the best ones among tested values for each trajectory length $hL$. Smaller values of $\varphi$ tend to perform better for smaller values of the product $hL$ and vice versa. We then use the same values of $\varphi$ for simulations with the GHMC method. All our experiments are carried out with the identity mass matrix for HMC, MALA, GHMC and MMHMC. The computational time used for normalization of ESS and efficiency comparison is measured as CPU time

that each method takes to collect posterior samples. Except for the case of a banana-shaped distribution, for which we investigate trajectory of a single Markov chain, all results are averaged over ten independent runs. We examine the banana-shaped model with the Matlab code provided along with (Lan et al., 2015), in which we implemented the MMHMC method. The rest of experiments are carried out with the in-house software package HaiCS presented in Chapter 5.

Each test model has been prepared to sampling with MMHMC, which involves computation of derivatives of a model potential function.

### 6.2.1 Banana-shaped distribution

We begin with a comparison of space exploration achieved by MMHMC, Random Walk Metropolis-Hastings (RWMH), Hamiltonian Monte Carlo (HMC) and Riemann Manifold HMC (RMHMC) in sampling a 2-dimensional, non-linear target. Given data $\mathbf{y} = \{y_k\}_{k=1}^K$ we sample from a banana-shaped posterior distribution of the parameter $\boldsymbol{\theta} = (\theta_1, \theta_2)$ (Girolami and Calderhead, 2011b, discussion by Bornn and Cornebise) for which the likelihood and prior distributions are given as

$$\begin{aligned} y_k|\boldsymbol{\theta} &\sim \mathcal{N}(\theta_1 + \theta_2^2, \sigma_y^2),\ k = 1, \dots, K \\ \theta_1, \theta_2 &\sim \mathcal{N}(0, \sigma_\theta^2) \end{aligned}$$

respectively. Due to independence in the data and parameters, the posterior distribution is proportional to

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \prod_{k=1}^{K} p(y_k|\boldsymbol{\theta}) p(\theta_1) p(\theta_2).$$

The potential function becomes (see (2.15))

$$U(\boldsymbol{\theta}) = \frac{1}{2\sigma_y^2} \sum_{k=1}^{K} \left(y_k - \theta_1 - \theta_2^2\right)^2 + \log(\sigma_\theta^2 \sigma_y^{100}) + \frac{1}{2\sigma_\theta^2}(\theta_1^2 + \theta_2^2)$$

and its derivatives are

$$U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = -\frac{1}{\sigma_y^2} \left( \sum_{k=1}^{K} y_k - K(\theta_1 + \theta_2^2) \right) \begin{bmatrix} 1 \\ 2\theta_2 \end{bmatrix} + \frac{1}{\sigma_\theta^2} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

$$U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) = -\frac{1}{\sigma_y^2} \begin{bmatrix} K + \frac{1}{\sigma_\theta^2} & 2K\theta_2 \\ 2K\theta_2 & -2\left(\sum_{k=1}^{K} y_k - K\left(\theta_1 + 3\theta_2^2\right)\right) + \frac{1}{\sigma_\theta^2} \end{bmatrix}.$$

**Experimental setting.** We generate $K = 100$ data $\{y_k\}_{k=1}^K$ with $\theta_1 + \theta_2^2 = 1$, $\sigma_y = 2$ and $\sigma_\theta = 1$. Sampling with the MMHMC method is performed using the Verlet integrator, a fixed number of integration steps, a step size and a noise parameter with values $L = 7, h = 1/9, \varphi = 0.5$, respectively. We compare MMHMC with RWMH, HMC and RMHMC for which simulation parameters are chosen as suggested in (Lan et al., 2015).

**Results.** The dynamics of the four samplers is illustrated in Figure 6.1, in which we show sampling paths (lines) of the first 15 accepted proposals (dots). RWMH just started to explore the parameter space and is still located in the low-density tail. In contrast, other methods already visited high-density regions. As expected, RMHMC efficiently tracks a local curvature of the parameter space and is able to move along the ridge to its full extent. On the other hand, HMC and MMHMC tend to move across rather than along the ridge and therefore explore the space less efficiently. Figure 6.2 shows the coverage of the space

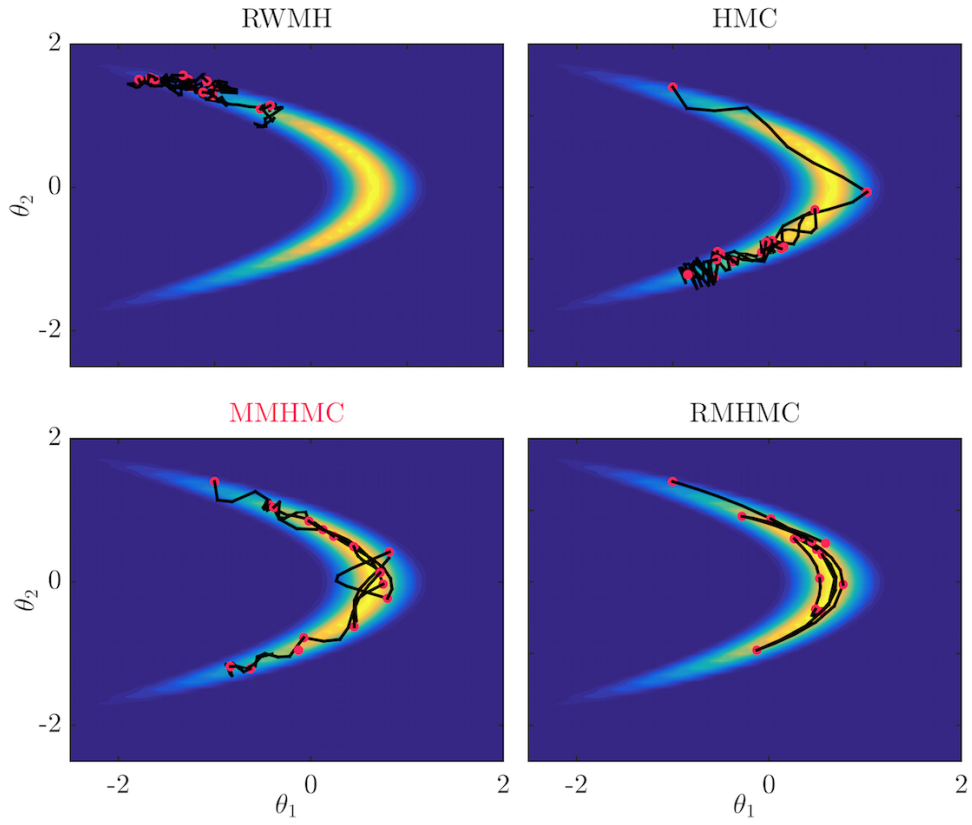


Figure 6.1: The first 15 Monte Carlo iterations with sampling paths (lines) and accepted proposals (dots) in sampling from a banana-shaped distribution with Random Walk Metropolis-Hastings (RWMH), Hamiltonian Monte Carlo (HMC), Mix&Match HMC (MMHMC) and Riemann Manifold HMC (RMHMC).

after 2000 iterations. We observe that RWMH[1] still did not cover the posterior distribution entirely. Other methods performed significantly better, though samples obtained with HMC did not reach the tails of the posterior, in contrast to MMHMC and RMHMC.

### 6.2.2 Multivariate Gaussian distribution

We take this experiment from (Hoffman and Gelman, 2014) for which the task is to sample from a $D$-dimensional Gaussian $\mathcal{N}(0, \Sigma)$. The precision matrix $\Sigma^{-1}$ is generated from a Wishart distribution with $D$ degrees of freedom and the $D$-dimensional identity scale matrix, which results in strong correlations among variates.

[1] RWMH was run $L$ times longer than other methods to somehow compensate for the cost of integration.

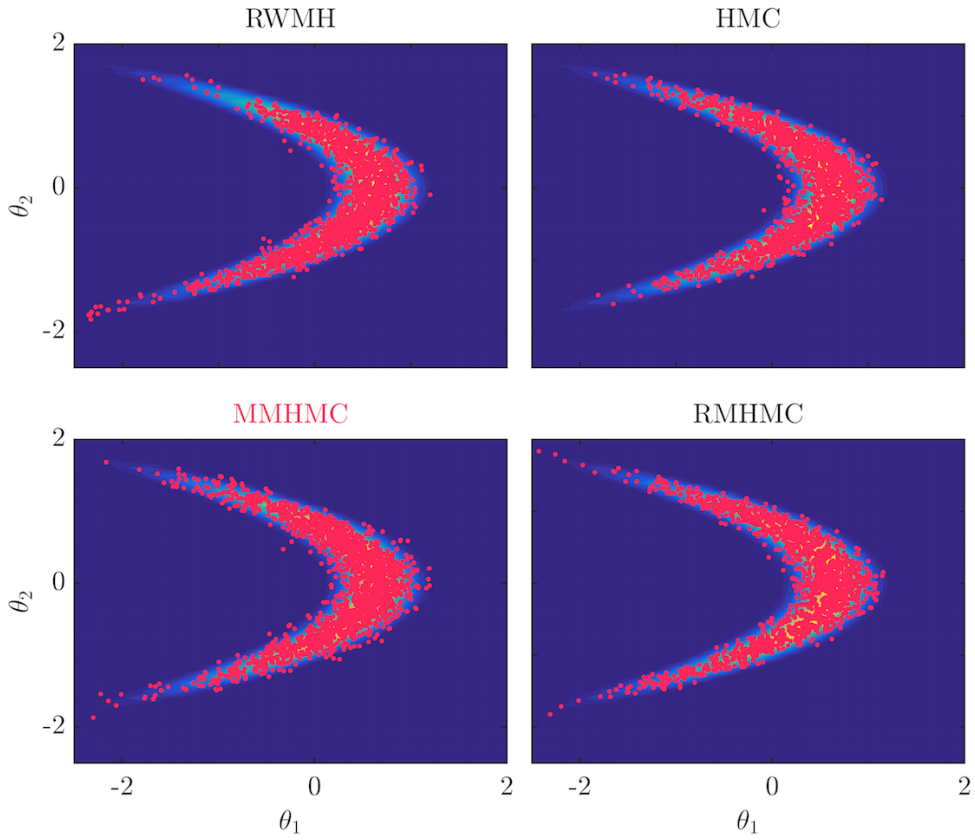


Figure 6.2: Exploration of space in sampling from a banana-shaped distribution achieved after 2000 samples obtained with Random Walk Metropolis-Hastings (RWMH), Hamiltonian Monte Carlo (HMC), Mix&Match HMC (MMHMC) and Riemann Manifold HMC (RMHMC). The red dots represent accepted points.

The potential function, its gradient and hessian are defined as

$$U(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T\Sigma^{-1}\boldsymbol{\theta}$$

$$U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \Sigma^{-1}\boldsymbol{\theta},$$

$$U_{\boldsymbol{\theta}\boldsymbol{\theta}}(\boldsymbol{\theta}) = \Sigma^{-1}.$$

**Experimental setting.** We perform tests for three different dimensions, $D = 100, 1000, 2000$, using the HMC and MMHMC methods and for $D = 100$ we additionally run GHMC. For the identity mass matrix, all three methods are invariant under rotations. Due to limited computational resources, we therefore choose for cases $D = 1000, 2000$ the covariance matrix $\Sigma$ to be diagonal with

$$\Sigma_{ii} = \sigma_i^2,$$

where $\sigma_i^2$ is the $i$th smallest eigenvalue of the original covariance matrix. Table 6.1 summarizes the integrators used for sampling with MMHMC, which were chosen according to the recommendations provided in Section 3.2.2. For two-stage integrators, we set a step size to $2h$ and a number of integration steps to $L/2$. We collect $10000$ samples with each method

| $D = 100$ | | $D = 1000$ | | $D = 2000$ | |
|---|---|---|---|---|---|
| $h$ | Integrator | $h$ | Integrator | $h$ | Integrator |
| $4 \cdot 10^{-2}$ | BCSS* | $8 \cdot 10^{-3}$ | ME* | $6 \cdot 10^{-3}$ | ME* |
| $5 \cdot 10^{-2}$ | BCSS* | $10 \cdot 10^{-3}$ | BCSS* | $8 \cdot 10^{-3}$ | BCSS* |
| $6 \cdot 10^{-2}$ | BCSS* | $12 \cdot 10^{-3}$ | BCSS* | $10 \cdot 10^{-3}$ | BCSS* |
| $7 \cdot 10^{-2}$ | Verlet | $14 \cdot 10^{-3}$ | BCSS* | $12 \cdot 10^{-3}$ | BCSS* |
| $8 \cdot 10^{-2}$ | Verlet | $16 \cdot 10^{-3}$ | BCSS* | | |

Table 6.1: Values of step size $h$ and corresponding integrators used for sampling from a $D$-dimensional Gaussian distribution with the MMHMC method.

and discard first $2000$ as a warm-up.

**Results.** Figure 6.3 compares the obtained acceptance rates (top) and the corresponding time-normalized minimum ESS across variates (bottom). While acceptance rates for HMC (and GHMC) drop considerably with increasing step size, especially for higher dimensions, MMHMC maintains very high acceptance. For $D = 100$, the acceptance rate for MMHMC starts to drop visibly but still stays reasonably high. As we noted before, the novel integrators do not improve over Verlet for small dimensions, and thus the Verlet integrator has been used for $D = 100$. It is interesting to note that although acceptance rates of GHMC are identical to those of HMC, the efficiency is considerably improved for smaller step sizes by just incorporating partial momenta update within HMC, as defined in the GHMC method. Bigger values of $L$ yield higher efficiency for HMC for all step sizes, however for MMHMC and GHMC this is not the case. For all tests, MMHMC demonstrates significantly higher sampling efficiency than HMC and GHMC, as can be seen from the inspection of ESS/s.

The results on sampling efficiency are summarized in Figure 6.4, from which we can appreciate the amount of improvement achieved with MMHMC compared to HMC. For a range of step sizes $h$ we show the efficiency factor (EF), i.e. relative time-normalized minimum ESS with respect to HMC, such that values above 1 indicate a superior performance of MMHMC. Each bar covers a range of numbers of integration steps $L$ tested for each step size $h$. The minimal EF value within a bar corresponds to the least difference in performance between HMC and MMHMC, whereas the maximal EF refers to the biggest improvement achieved by MMHMC over HMC. The improvement factor clearly increases with dimension. Depending on the choice of $h$ and $L$, the minimal improvement achieved is around 2 times (for the lowest dimension) and maximal one goes up to 40 times (for the highest dimension). Since optimal simulation parameters are not known a priori, we expect that sampling efficiency using MMHMC for this kind of problems will be at least 2 times better than using HMC, but very likely much higher.

Beside acceptance rates and sampling efficiency of the tested samplers, we are also interested in the effect that a choice of simulation parameters $L$ and $h$ has on the performance of each sampler. Figure 6.5 shows the maximal relative improvement in the time-normalized minimum ESS achieved for different choices of $L$ while keeping the step size constant for

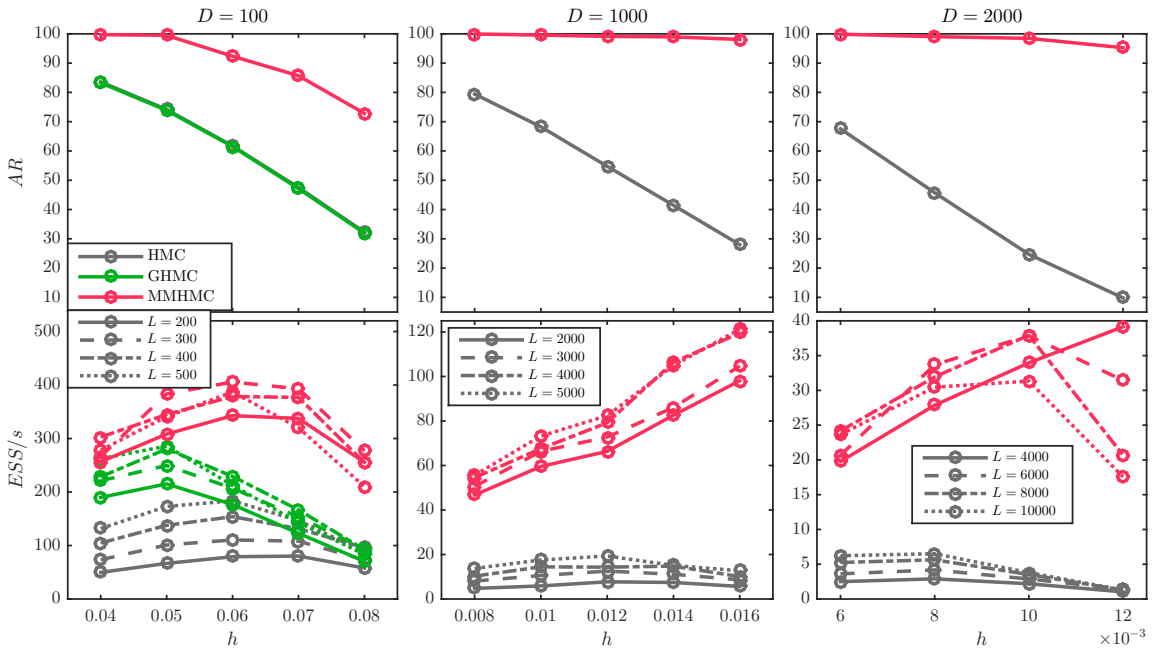


Figure 6.3: Acceptance rate (top) and time-normalized minimum ESS (bottom) for a range of step sizes $h$ and number of integration steps $L$, obtained in sampling from a $D$-dimensional Gaussian distribution with Hamiltonian Monte Carlo (HMC), Generalized HMC (GHMC) and Mix&Match HMC (MMHMC).

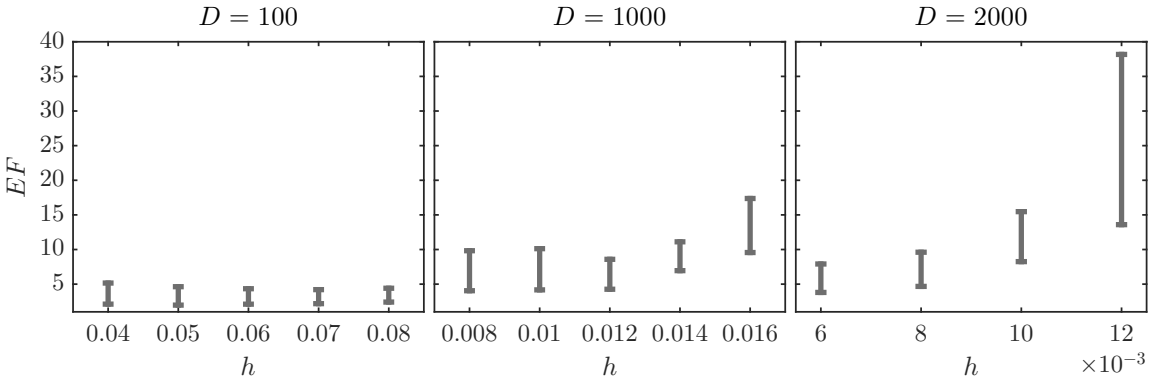


Figure 6.4: Relative sampling efficiency (EF) of MMHMC w.r.t. HMC for a range of step sizes $h$ in sampling from a $D$-dimensional Gaussian distribution. Each bar accounts for the data obtained with different choices of numbers of integration steps $L$.

each method. It demonstrates an additional advantage of MMHMC over the HMC method. Indeed, for HMC a "right" choice of $L$ can improve efficiency up to three times, i.e. a "wrong" choice can worsen efficiency up to three times, whereas for MMHMC the effect is almost always around $30\%$. Therefore, the problem of finding an optimal $L$ is less relevant to MMHMC than to HMC. This feature of MMHMC is particularly useful, as till now there is no universal criterion for finding an optimal value of $L$. The only exception appears in the case $D = 2000$ at the largest step size $h$, where MMHMC has a higher relative dependence on $L$ than HMC. The likely reason for this is because all choices of $L$ other than the smallest one induce trajectory lengths $hL$ that are too large and consequently, those trajectories are

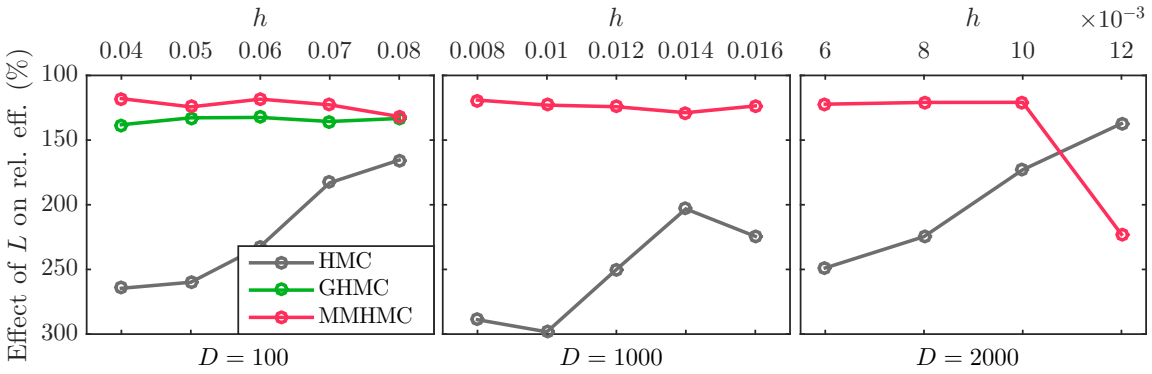


Figure 6.5: Effect of numbers of integration steps $L$ on sampling efficiency of HMC, GHMC and MMHMC for sampling from a $D$-dimensional Gaussian distribution. Y-axis shows the maximal relative improvement in time-normalized minimum ESS achieved when varying $L$ for a fixed step size $h$. MMHMC demonstrates superiority over HMC, while being less sensitive to changes in parameter $L$.

making turns and getting closer to their initial state while still performing integration steps and adding computational cost.

In the same fashion, in Figure 6.6 we show how the relative efficiency of the two tested methods is affected by the changes in the chosen step size $h$. In this case, a clear advantage of one method over another is not obvious, though we note that the effect of $h$ on HMC performance is quite high for the highest dimension. This is not surprising if we recall the drop in acceptance rate for HMC for increasing step size in higher dimensions. In contrast, the maximal improvement (or reduction) in the efficiency of MMHMC due to a choice of a step size stays around 2 times for all dimensions and numbers of integration steps $L$.

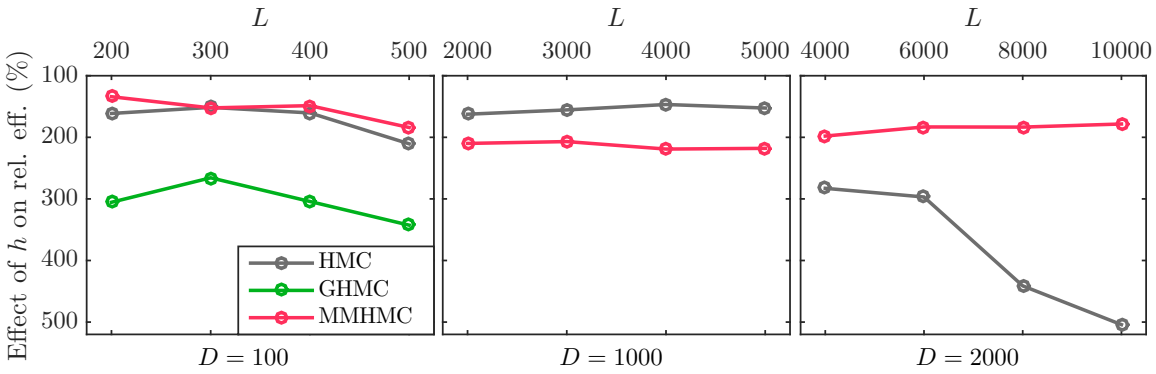


Figure 6.6: Effect of step size $h$ on sampling efficiency of HMC, GHMC and MMHMC for sampling from a $D$-dimensional Gaussian distribution. Y-axis shows maximal relative improvement in time-normalized minimum ESS achieved with different choices of $h$ and a fixed number of integration steps $L$.

### 6.2.3 Bayesian Logistic Regression model

The Bayesian Logistic Regression (BLR) model is used for solving binary classification problems appearing across various fields such as medical and social sciences, engineering, insurance, ecology, sports, etc.

Consider $K$ instances of data $\{\mathbf{x}_k, y_k\}_{k=1}^K$, where $\mathbf{x}_k$ are vectors of $D-1$ covariates and $y_k \in \{0,1\}$ are binary responses. In the BLR model, the response variable $\mathbf{y} = (y_1, \ldots, y_K)$ is governed by a Bernoulli distribution with parameter $\mathbf{p} = (p_1, \ldots, p_K)$. The unobserved probability $p_k$ of a particular outcome is linked to the linear predictor function through the logit function, i.e.

$$\operatorname{logit}(p_k) = \theta_0 + \theta_1 x_{1,k} + \cdots + \theta_{D-1} x_{D-1,k},$$

where $\operatorname{logit}(p) = \log\left(p/(1-p)\right)$ and $\boldsymbol{\theta} \in \mathbb{R}^D$ is the regression coefficient vector. The prior of the regression coefficient is given e.g. as $\boldsymbol{\theta} \sim \mathcal{N}(0, \alpha\mathbb{I})$, with a known $\alpha$.

If we construct the design matrix $X \in \mathbb{R}^{K,D}$ of input data as

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1,D-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{K1} & \cdots & x_{K,D-1} \end{bmatrix},$$

the likelihood function is given as

$$p(\mathbf{y}|X, \boldsymbol{\theta}) = \prod_{k=1}^K p(y_k|X_k, \boldsymbol{\theta}) = \prod_{k=1}^K \left(\frac{\mathrm{e}^{X_k\boldsymbol{\theta}}}{1+\mathrm{e}^{X_k\boldsymbol{\theta}}}\right)^{y_k} \left(\frac{1}{1+\mathrm{e}^{X_k\boldsymbol{\theta}}}\right)^{1-y_k},$$

where $X_k$ is the $k$th row of the matrix $X$. The corresponding posterior distribution over the regression coefficients is

$$\pi(\boldsymbol{\theta}|\mathbf{y}, \mathbf{x}) \propto \prod_{k=1}^K p(y_k|X_k, \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

with the prior

$$p(\boldsymbol{\theta}) \propto \exp\left\{-\frac{\boldsymbol{\theta}^T\boldsymbol{\theta}}{2\alpha}\right\}.$$

The potential function then reads

$$U(\boldsymbol{\theta}) = -\sum_{k=1}^K \left[ y_k \sum_{i=1}^D X_{ki}\theta_i - \log\left(1 + \exp\left(\sum_{i=1}^D X_{ki}\theta_i\right)\right)\right] + \frac{1}{2\alpha}\sum_{i=1}^D \theta_i^2$$

and its derivatives

$$\partial_{\theta_i} U(\boldsymbol{\theta}) = -\sum_{k=1}^K X_{ki}\left(y_k - \frac{\exp\left(\sum_{l=1}^D X_{kl}\theta_l\right)}{1+\exp\left(\sum_{l=1}^D X_{kl}\theta_l\right)}\right) + \frac{1}{\alpha}\theta_i$$

$$\partial_{\theta_i\theta_j} U(\boldsymbol{\theta}) = \sum_{k=1}^K \frac{X_{ki}X_{kj}\exp\left(\sum_{l=1}^D X_{kl}\theta_l\right)}{\left(1+\exp\left(\sum_{l=1}^D X_{kl}\theta_l\right)\right)^2} + \frac{\delta_{ij}}{\alpha},$$

with

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

**Experimental setting.** We use four different real datasets available from the University of California Irvine Machine Learning Repository (Lichman, 2013). The dataset characteristics, such as names, numbers of regression parameters ($D$) and observations ($K$) are summarized in Table 6.2.

| Dataset | $D$ | $K$ |
|---|---|---|
| German | 25 | 1000 |
| Sonar | 61 | 208 |
| Musk | 167 | 476 |
| Secom | 444 | 1567 |

Table 6.2: Datasets used for BLR model with corresponding number of regression parameters ($D$) and number of observations ($K$).

By following a common procedure, we normalize input data such that each covariate has zero mean and standard deviation of one. For each dataset, a diffuse Gaussian prior is imposed by setting $\alpha = 100$.

In all experiments, $N = 5000$ posterior samples were generated after discarding the first $5000$ as a warm-up. Apart from the comparison of MMHMC with HMC over the range of datasets, we also tested it against MALA on the German dataset and GHMC on the German and Musk datasets. We do not investigate the performance of RMHMC since as it was stated in (Girolami and Calderhead, 2011b), RMHMC does not outperform HMC for dimensions as high as for the German dataset, which in our case is the dataset of the smallest dimension.

In these experiments, we use MMHMC with the Verlet integrator, since dimensions of the four datasets may be too small to expect an improvement with the novel integrators derived in Section 3.2.2.

**Results.** Acceptance rate (top) and time-normalized minimum ESS across variates (bottom) obtained for BLR are presented in Figures 6.7 and 6.8. For all datasets, the acceptance rate is the highest for MMHMC, as is expected. Except MALA, which exhibits poor performance, all methods demonstrate comparable efficiency for the smallest dataset. The GHMC method improves HMC for the Musk dataset. Nevertheless, MMHMC outperforms both HMC and GHMC for a range of simulation parameters. We note that the parameter $L$ found to be the best for HMC is not necessarily the best for MMHMC. Actually, too long values of $L$ seem to result in poorer overall efficiency for MMHMC, although the computational overhead is smaller for larger $L$, due to a less frequent calculation of modified Hamiltonians. In contrast, longer trajectories are needed for HMC to achieve its full potential for larger datasets.

Figure 6.9 summarizes results on sampling efficiency in terms of relative improvement of MMHMC compared to HMC, for a range of step sizes $h$ and numbers of integration steps $L$ (included within bars). We note that MMHMC and HMC have comparable performance for the smallest dimension $D = 25$; however, the sampling efficiency grows with increasing

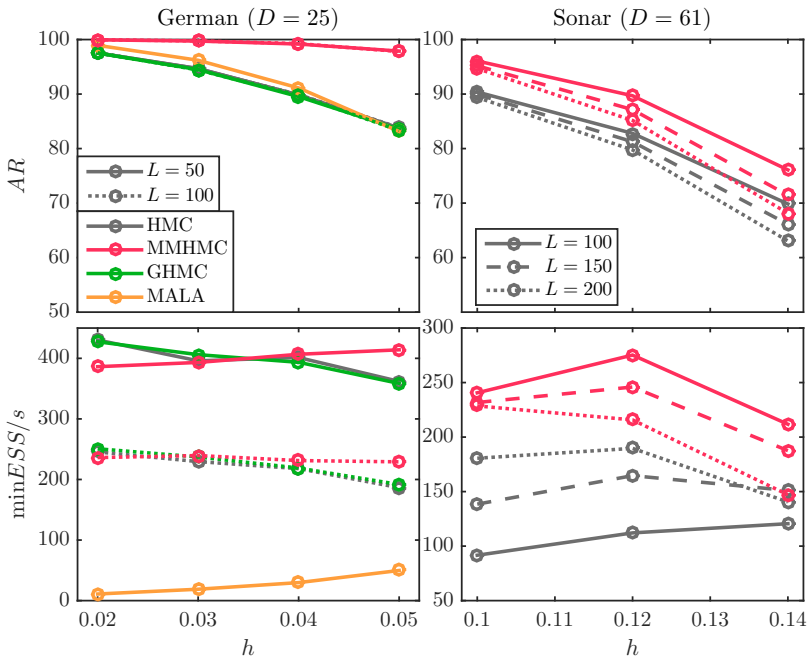


Figure 6.7: Acceptance rate (top) and time-normalized minimum ESS (bottom) for Bayesian logistic regression using Hamiltonian Monte Carlo (HMC), Mix&Match HMC (MMHMC), Generalized HMC (GHMC) and Metropolis Adjusted Langevin Algorithm (MALA), for a range of step sizes $h$ and numbers of integration steps $L$, for the German and Sonar datasets.

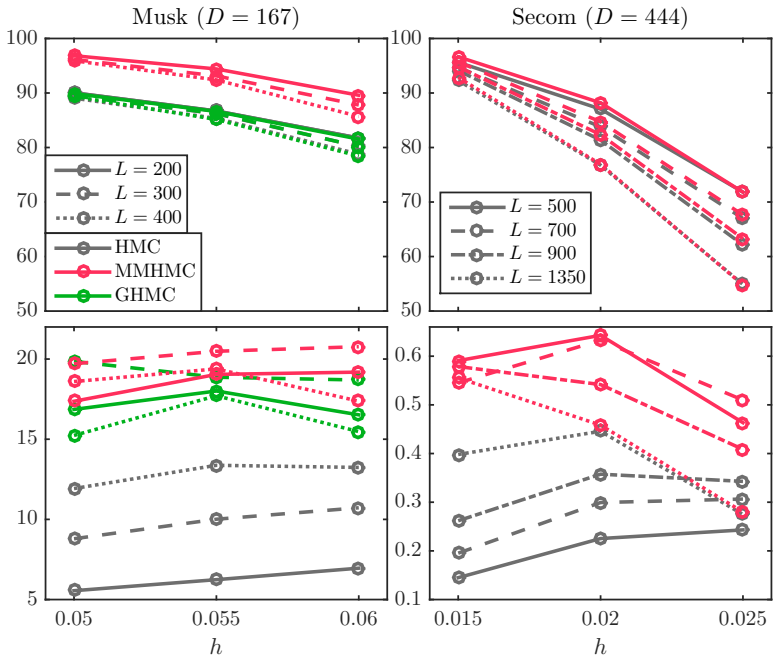


Figure 6.8: Acceptance rate (top) and time-normalized minimum ESS (bottom) for Bayesian logistic regression using Hamiltonian Monte Carlo (HMC), Mix&Match HMC (MMHMC), Generalized HMC (GHMC) and Metropolis Adjusted Langevin Algorithm (MALA), for a range of step sizes $h$ and numbers of integration steps $L$, for the Musk and Secom datasets.

dimension in favor of MMHMC. For BLR model and tested datasets, MMHMC demonstrates improvement over HMC of up to 4 times.

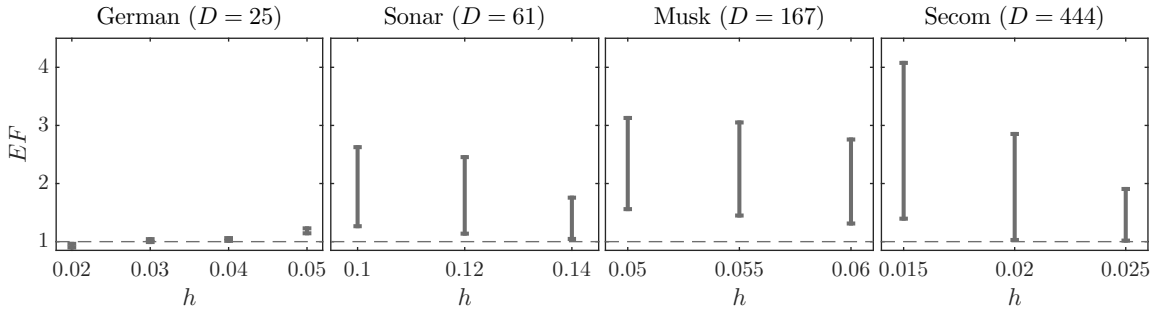


Figure 6.9: Relative sampling efficiency (EF) of MMHMC w.r.t. HMC for a range of step sizes $h$, in sampling of Bayesian logistic regression models. Each bar accounts for the data obtained with different choices of numbers of integration steps $L$.

### 6.2.4 Stochastic Volatility model

The volatility of price returns, as a magnitude of price fluctuation, is important for measuring the risk in empirical finance. Nevertheless, it is very difficult to extract the true volatility from asset price returns themselves. Stochastic volatility (SV) models turned out to be a useful tool for modeling time-varying volatility with significant potential for applications (e.g. risk management/risk prediction, pricing of financial derivatives). These models appear as discrete approximations to various diffusion processes in the theoretical finance literature on asset pricing (Hull and White, 1987) and have been extensively studied in both theoretical and empirical finance literature for more than 20 years.

We consider the standard SV model defined with the latent, log-volatilities following autoregressive AR(1) process. The model, as described by Kim et al. (1998), takes the following form

$$\begin{aligned} y_t &= \beta \exp(x_t/2)\epsilon_t, & \epsilon_t &\sim \mathcal{N}(0,1) \\ x_t &= \phi x_{t-1} + \sigma\eta_t, & \eta_t &\sim \mathcal{N}(0,1) \\ x_1 &\sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right) \end{aligned}$$

where $y_t$ are observed data of mean corrected log-returns, equidistantly spaced in time for $t = 1, \ldots, T$, and $x_t$ are latent variables of log-volatility assumed to follow a stationary process. This assumption leads to the constraint $|\phi| < 1$. The error terms $\epsilon_t$ and $\eta_t$ are serially and mutually uncorrelated white noise sequences with the standard normal distribution. The parameter $\beta$ of the model can be interpreted as the modal instantaneous volatility, $\phi$ as the persistence in the volatility and $\sigma$ as the volatility of the log-volatility, leading to the second constraint $\sigma > 0$.

Let denote the vector of model parameters as $\boldsymbol{\theta} = (\beta, \sigma, \phi)$. The difficulty in inferring SV parameters, i.e. sampling from the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{y})$ given a set of observed

log-returns $\mathbf{y} = (y_1, ..., y_T)$ lies in the fact that the likelihood function, defined as

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta}) \mathrm{d}\mathbf{x},$$

is an analytically intractable $T$-dimensional integral with respect to the unknown log-volatilities $\mathbf{x} = (x_1, ..., x_T)$. Fortunately, the MCMC simulation-based inference overcomes this difficulty. The first such analysis of the standard SV model was given by Jacquier et al. (1994) and the estimations using HMC based methodologies were later carried out by Chen et al. (2000), Liu (2008), Takaishi (2013), Girolami and Calderhead (2011b), Zhang and Sutton (2014), and Wang et al. (2013). In the MCMC approach, instead of sampling from $\pi(\boldsymbol{\theta}|\mathbf{y})$ we focus on the joint posterior distribution of both model parameters and latent volatilities, given through the conditional distributions as

$$\begin{aligned}
\pi(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y}) &\propto p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{x}) p(\boldsymbol{\theta}, \mathbf{x}) = p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{x}) p(\mathbf{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \\
&= p(\boldsymbol{\theta}) p(x_1|\boldsymbol{\theta}) \prod_{t=1}^{T} p(y_t|\boldsymbol{\theta}, x_t) \prod_{t=2}^{T} p(x_t|x_{t-1}, \boldsymbol{\theta}) \\
&= p(\beta) p(\sigma) p(\phi) p(x_1|\sigma, \phi) \prod_{t=1}^{T} p(y_t|x_t, \beta) \prod_{t=2}^{T} p(x_t|x_{t-1}, \sigma, \phi).
\end{aligned}$$

The expressions for the conditional distributions follow straightforwardly from the model specification as

$$p(x_1|\sigma, \phi) = \left(\frac{2\pi\sigma^2}{1-\phi^2}\right)^{-\frac{1}{2}} \exp\left\{-\frac{x_1^2(1-\phi^2)}{2\sigma^2}\right\}$$

$$\begin{aligned}
p(x_t|x_{t-1}, \sigma, \phi) &= p_{\eta_t}\left(\frac{x_t - \phi x_{t-1}}{\sigma}\middle| x_{t-1}, \sigma, \phi\right)\left|\frac{\mathrm{d}((x_t - \phi x_{t-1})/\sigma)}{\mathrm{d}x_t}\right| \\
&= (2\pi)^{-\frac{1}{2}} \frac{1}{\sigma} \exp\left\{-\frac{(x_t - \phi x_{t-1})^2}{2\sigma^2}\right\}
\end{aligned}$$

$$\begin{aligned}
p(y_t|x_t, \beta) &= p_{\epsilon_t}\left(\frac{y_t}{\beta \exp\{x_t/2\}}\middle| x_t, \beta\right)\left|\frac{\mathrm{d}(y_t/\beta \exp\{x_t/2\}}{\mathrm{d}y_t}\right| \\
&= (2\pi)^{-\frac{1}{2}} \exp\left\{-\frac{y_t^2}{2\beta^2 \exp\{x_t\}}\right\} \frac{1}{\beta} \exp\left\{-\frac{x_t}{2}\right\} \\
&= (2\pi)^{-\frac{1}{2}} \frac{1}{\beta} \exp\left\{-\frac{x_t}{2} - \frac{y_t^2}{2\beta^2 \exp\{x_t\}}\right\}.
\end{aligned}$$

We follow Liu (2008) and choose the priors as $p(\beta) \propto 1/\beta, \sigma^2 \sim$ Scale-inv-$\chi^2(10, 0.05), (\phi+1)/2 \sim$ Beta$(20, 1.5)$, leading to

$$\begin{aligned}
p(\beta) &\propto \frac{1}{\beta} \\
p(\sigma) &\propto \sigma^{-11} \exp\{-1/4\sigma^2\} \\
p(\phi) &\propto (\phi+1)^{19} (1-\phi)^{\frac{1}{2}}.
\end{aligned}$$

We employ HMC based samplers and instead of sampling jointly model parameters and latent volatilities from $\pi(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$, we follow a common procedure of cycling through the two full conditional distributions $\pi(\boldsymbol{\theta}|\mathbf{y}, \mathbf{x})$ and $\pi(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$.

Since HMC methods sample real valued parameters, we handle the constraints $\sigma^2 > 0$ and $-1 \leq \phi \leq 1$ by making use of the transformation $\mathcal{T} : \boldsymbol{\theta} \to \bar{\boldsymbol{\theta}}$ to the real line, defined as

$$\bar{\boldsymbol{\theta}} = \mathcal{T}(\boldsymbol{\theta}) = (\beta, \ln(\sigma), \operatorname{artanh}(\phi)) = (\beta, \gamma, \alpha)$$

with the Jacobian

$$\mathcal{J}_{\mathcal{T}} = \begin{bmatrix} \frac{\mathrm{d}\beta}{\mathrm{d}\beta} & 0 & 0 \\ 0 & \frac{\mathrm{d}\gamma}{\mathrm{d}\sigma} & 0 \\ 0 & 0 & \frac{\mathrm{d}\alpha}{\mathrm{d}\phi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma^{-1} & 0 \\ 0 & 0 & (1-\phi^2)^{-1} \end{bmatrix}.$$

The inverse transformation $\mathcal{T}^{-1}$ to the constrained parameters is

$$\boldsymbol{\theta} = \mathcal{T}^{-1}(\bar{\boldsymbol{\theta}}) = (\beta, e^{\gamma}, \tanh(\alpha)) \tag{6.11}$$

and its Jacobian is

$$\mathcal{J}_{\mathcal{T}^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{\gamma} & 0 \\ 0 & 0 & 1 - \tanh^2(\alpha) \end{bmatrix}. \tag{6.12}$$

Sampling is performed in the unconstrained space (see Section 4.1). First, we sample the transformed parameters from the distribution $\pi(\bar{\boldsymbol{\theta}}|\mathbf{x}, \mathbf{y})$, for which the potential function can be obtained as

$$\begin{aligned} \bar{U}(\bar{\boldsymbol{\theta}}) \quad = \quad & \sum_{t=1}^{T} \frac{x_t}{2} + \frac{1}{2\beta^2} \sum_{t=1}^{T} \frac{y_t^2}{\exp(x_t)} + \frac{1}{2\exp(2\gamma)} \sum_{t=2}^{T} (x_t - \tanh(\alpha) x_{t-1})^2 + \frac{x_1^2 (1 - \tanh(\alpha)^2)}{2\exp(2\gamma)} \\ & + T\log(\beta) + T\gamma - \frac{1}{2}\log((1 - \tanh(\alpha)^2)) \\ & \underbrace{+ \log(\beta) + 11\gamma + \frac{1}{4\exp(2\gamma)} - 19\log\left(\tanh(\alpha) + 1\right) - \frac{1}{2}\log\left(1 - \tanh(\alpha)\right)}_{prior} \\ & \underbrace{-\gamma - \log(1 - \tanh(\alpha)^2)}_{Jacobian}. \end{aligned}$$

The gradient of the potential function with respect to the transformed parameters follows from the chain rule (i.e. $\nabla_{\boldsymbol{\theta}} \bar{U} = \nabla_{\bar{\boldsymbol{\theta}}} \bar{U} \mathcal{J}_{\mathcal{T}}$) and reads

$$\nabla_{\bar{\boldsymbol{\theta}}} \bar{U} = \nabla_{\boldsymbol{\theta}} \bar{U} \mathcal{J}_{\mathcal{T}}^{-1}.$$

In particular, the partial derivatives are

$$\begin{aligned}
\partial_\beta \bar{U}(\bar{\boldsymbol{\theta}}) &= \frac{T+1}{\beta} - \sum_{t=1}^{T} \frac{y_t^2}{\beta^3 e^{x_t}} \\
\partial_\gamma \bar{U}(\bar{\boldsymbol{\theta}}) &= -\frac{1}{\exp(2\gamma)}\Big(\sum_{t=2}^{T}(x_t - \tanh(\alpha)x_{t-1})^2 + x_1^2(1-\tanh(\alpha)^2) + \frac{1}{2}\Big) + T + 10 \\
\partial_\alpha \bar{U}(\bar{\boldsymbol{\theta}}) &= -\frac{1-\tanh(\alpha)^2}{\exp(2\gamma)}\Big(\sum_{t=2}^{T}(x_t - \tanh(\alpha)x_{t-1})x_{t-1} + x_1^2\tanh(\alpha)\Big) \\
&\quad +22.5\tanh(\alpha) - 18.5.
\end{aligned}$$

Using the chain rule for higher derivatives

$$\big[\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}}\bar{U}\big]_{ij} = \sum_{\epsilon\in\{\beta,\gamma,\alpha\}} \partial_\epsilon \bar{U}\cdot\epsilon''|_{ij} + \sum_{\epsilon,\xi\in\{\beta,\gamma,\alpha\}} \partial_{\epsilon\xi}\bar{U}\cdot\epsilon'|_i\cdot\xi'|_j, \quad i,j\in\{\beta,\sigma,\phi\},$$

we obtain

$$\begin{aligned}
\partial_{\beta\beta}\bar{U}(\bar{\boldsymbol{\theta}}) &= \frac{1}{\beta^2}\Big(-T-1+\frac{3}{\beta^2}\sum_{t=1}^{T}\frac{y_t^2}{\exp\{x_t\}}\Big) \\
\partial_{\gamma\gamma}\bar{U}(\bar{\boldsymbol{\theta}}) &= \frac{2}{\exp(2\gamma)}\Big(\sum_{t=2}^{T}(x_t-\tanh(\alpha)x_{t-1})^2 + x_1^2(1-\tanh(\alpha)^2) + \frac{1}{2}\Big) \\
\partial_{\beta\gamma}\bar{U}(\bar{\boldsymbol{\theta}}) &= \partial_{\gamma\beta}\bar{U} = 0 \\
\partial_{\beta\alpha}\bar{U}(\bar{\boldsymbol{\theta}}) &= \partial_{\alpha\beta}\bar{U} = 0 \\
\partial_{\gamma\alpha}\bar{U}(\bar{\boldsymbol{\theta}}) &= \frac{2(1-\tanh(\alpha)^2)}{\exp(2\gamma)}\Big(\sum_{t=2}^{T}x_{t-1}(x_t-\tanh(\alpha)x_{t-1}) + x_1^2\tanh(\alpha)\Big) \\
\partial_{\alpha\alpha}\bar{U}(\bar{\boldsymbol{\theta}}) &= \frac{(1-\tanh(\alpha)^2)}{\exp(2\gamma)}\Bigg((1-\tanh(\alpha)^2)\sum_{t=2}^{T-1}x_t^2 + 2\tanh(\alpha)\Big(\sum_{t=2}^{T}(x_t-\tanh(\alpha)x_{t-1})x_{t-1} \\
&\quad +x_1^2\tanh(\alpha)\Big)\Bigg) + 22.5(1-\tanh(\alpha)^2).
\end{aligned}$$

We note that the second order partial derivatives can be expressed in terms of the first order derivatives, therefore for an efficient implementation of the SV model we use the following expressions

$$\begin{aligned}
\partial_{\beta\beta}\bar{U}(\bar{\boldsymbol{\theta}}) &= \Big(\frac{2(T+1)}{\beta} - 3\partial_\beta\bar{U}\Big)\frac{1}{\beta} \\
\partial_{\gamma\gamma}\bar{U}(\bar{\boldsymbol{\theta}}) &= 2(T+10-\partial_\gamma\bar{U}) \\
\partial_{\beta\gamma}\bar{U}(\bar{\boldsymbol{\theta}}) &= \partial_{\gamma\beta}\bar{U} = 0 \\
\partial_{\beta\alpha}\bar{U}(\bar{\boldsymbol{\theta}}) &= \partial_{\alpha\beta}\bar{U} = 0 \\
\partial_{\gamma\alpha}\bar{U}(\bar{\boldsymbol{\theta}}) &= 2(22.5\tanh(\alpha) - 18.5 - \partial_\alpha\bar{U})
\end{aligned}$$

$$\partial_{\alpha\alpha}\bar{U}(\bar{\boldsymbol{\theta}}) = \frac{(1-\tanh(\alpha)^2)^2}{\exp(2\gamma)}\sum_{t=2}^{T-1} x_t^2 + \tanh(\alpha)\partial_{\gamma\alpha}\bar{U} + 22.5(1-\tanh(\alpha)^2).$$

This concludes the preparation for the first sampling step.

The second sampling step consists of simulating latent volatilities from the distribution $\pi(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$. The priors for the model parameters are constant with respect to the volatilities and thus can be omitted from the potential function, which takes the form

$$\begin{aligned}\bar{U}(\mathbf{x}) = &\sum_{t=1}^{T}\frac{x_t}{2} + \frac{1}{2\beta^2}\sum_{t=1}^{T}\frac{y_t^2}{\exp(x_t)} + \frac{1}{2\exp(2\gamma)}\sum_{t=2}^{T}(x_t-\tanh(\alpha)x_{t-1})^2\\ &+\frac{x_1^2(1-\tanh(\alpha)^2)}{2\exp(2\gamma)} + T\log(\beta) + T\gamma - \frac{1}{2}\log((1-\tanh(\alpha)^2)).\end{aligned}$$

First order partial derivatives follow straightforwardly as

$$\begin{aligned}\partial_{x_1}\bar{U}(\mathbf{x}) &= \frac{1}{2} - \frac{y_1^2}{2\beta^2\exp\{x_1\}} + \frac{x_1-\tanh(\alpha)x_2}{\exp(2\gamma)}\\ \partial_{x_i}\bar{U}(\mathbf{x}) &= \frac{1}{2} - \frac{y_i^2}{2\beta^2\exp\{x_i\}} + \frac{x_i(\tanh(\alpha)^2+1)-\tanh(\alpha)(x_{i+1}+x_{i-1})}{\exp(2\gamma)},\quad i=2,\ldots,T-1\\ \partial_{x_T}\bar{U}(\mathbf{x}) &= \frac{1}{2} - \frac{y_T^2}{2\beta^2\exp\{x_T\}} + \frac{x_T-\tanh(\alpha)x_{T-1}}{\exp(2\gamma)}\end{aligned}$$

and second order derivatives as

$$\begin{aligned}\partial_{x_1^2}\bar{U}(\mathbf{x}) &= \frac{y_1^2}{2\beta^2\exp\{x_1\}} + \frac{1}{\exp(2\gamma)}\\ \partial_{x_i^2}\bar{U}(\mathbf{x}) &= \frac{y_i^2}{2\beta^2\exp\{x_i\}} + \frac{1+\tanh(\alpha)^2}{\exp(2\gamma)},\ i=2,\ldots,T-1\\ \partial_{x_T^2}\bar{U}(\mathbf{x}) &= \frac{y_T^2}{2\beta^2\exp\{x_T\}} + \frac{1}{\exp(2\gamma)}\\ \partial_{x_ix_{i+1}}\bar{U}(\mathbf{x}) &= -\frac{\tanh(\alpha)}{\exp(2\gamma)},\ i=1,\ldots,T-1\\ \partial_{x_ix_j}\bar{U}(\mathbf{x}) &= \partial_{x_jx_i}\bar{U}(\mathbf{x})\\ \partial_{x_ix_j}\bar{U}(\mathbf{x}) &= 0,\ j\neq i-1,i+1.\end{aligned}$$

As in the case of partial derivatives of model parameters, we can write second order derivatives of latent volatilities in terms of first order derivatives, thus simplifying the implementation with expressions

$$\begin{aligned}\partial_{x_1^2}\bar{U}(\mathbf{x}) &= -\partial_{x_1}\bar{U}(\mathbf{x}) + \frac{1}{2} + \frac{1+x_1-\tanh(\alpha)x_2}{\exp(2\gamma)}\\ \partial_{x_i^2}\bar{U}(\mathbf{x}) &= -\partial_{x_i}\bar{U}(\mathbf{x}) + \frac{1}{2} + \frac{(1+\tanh(\alpha)^2)(x_i+1)-\tanh(\alpha)(x_{i+1}+x_{i-1})}{\exp(2\gamma)},\\ &\quad i=2,\ldots,T-1\\ \partial_{x_T^2}\bar{U}(\mathbf{x}) &= -\partial_{x_T}\bar{U}(\mathbf{x}) + \frac{1}{2} + \frac{1+x_T-\tanh(\alpha)x_{T-1}}{\exp(2\gamma)}\end{aligned}$$

$$\begin{aligned}\partial_{x_i x_{i+1}} \bar{U}(\mathbf{x}) &= -\frac{\tanh(\alpha)}{\exp(2\gamma)},\ i = 1, \ldots, T-1 \\ \partial_{x_i x_j} \bar{U}(\mathbf{x}) &= \partial_{x_j x_i} \bar{U}(\mathbf{x}) \\ \partial_{x_i x_j} \bar{U}(\mathbf{x}) &= 0,\ j \neq i-1, i+1.\end{aligned}$$

**Experimental setting.** We examine sampling of the standard SV on simulated data with values $\beta = 0.65, \sigma = 0.15, \phi = 0.98$, for $T = 2000, 5000, 10000$ time points. This results in three experiments of dimensions $D = 2003, 5003, 10003$, which include three model parameters and $T$ latent volatility variables to sample. We run 10000 iterations as a warm-up and generate 200000 posterior samples collecting every 10th sample. We compare MMHMC with HMC, and for $D = 2003$ we additionally run the GHMC and RMHMC methods. The comparison with RMHMC was done indirectly by running HMC and RMHMC with the Matlab code by Girolami and Calderhead (2011a). The noise parameter for MMHMC and GHMC was tuned to values $\varphi_{\boldsymbol{\theta}} = 0.5, \varphi_{\mathbf{x}} = 0.8$, the number of integration steps for HMC, GHMC and MMHMC to $L_{\boldsymbol{\theta}} = 6, L_{\mathbf{x}} = 76$ and for RMHMC we took values from the corresponding paper, i.e. $L_{\boldsymbol{\theta}} = 6, L_{\mathbf{x}} = 50$. The step sizes used are summarized in Table 6.3. Naturally, for two-stage integrators, we set a step size to $2h$ and a number of integration steps to $L/2$.

| $D$ | Method | $h_{\boldsymbol{\theta}}$ | $h_{\mathbf{x}}$ |
|---|---|---|---|
| 2003 | HMC | 0.009 | 0.03 |
| | GHMC | 0.009 | 0.03 |
| | RMHMC | 0.5 | 0.1 |
| | MMHMC | 0.009 | 0.0225 |
| 5003 | HMC | 0.006 | 0.02 |
| | MMHMC | 0.006 | 0.0185 |
| 10003 | HMC | 0.004 | 0.02 |
| | MMHMC | 0.004 | 0.015 |

TABLE 6.3: Step size values used for the SV model experiments.

**Results.** We first show ESS for SV model parameters obtained using different integrators within the MMHMC sampler. The results are summarized in Table 6.4 and suggest the advantage of using novel integrators specifically derived for sampling with modified Hamiltonians. The rest of results presented in this section are obtained with the M-ME integrator.

We next investigate convergence to the stationary distribution of the tested samplers by calculating $\hat{R}$ as a function of a number of Monte Carlo iterations (see Figure 6.10). If we choose a commonly used threshold of $1.1$, or even $1.05$, we notice that for all methods the values of $\hat{R}$ drop quickly below the threshold, with a slightly slower convergence demonstrated by the HMC method in high dimensional experiments and the fastest one achieved by RMHMC. We note that here only MC iterations are taken into account and not the computational time. However, we stress that the computational cost per iteration in

| Integrator | ESS | | |
|---|---|---|---|
| | $\beta$ | $\sigma$ | $\phi$ |
| Verlet | 1332 | 1208 | 2308 |
| M-BCSS | 1335 | 1237 | 2411 |
| **M-ME** | 1544 | 1175 | 2454 |

Table 6.4: ESS for SV model parameters obtained using different integrators within the MMHMC method.

tested methodologies varies, for example, one iteration of RMHMC takes $66\%$ more time than needed for one iteration of HMC or $49\%$ more time than needed for one iteration of MMHMC.

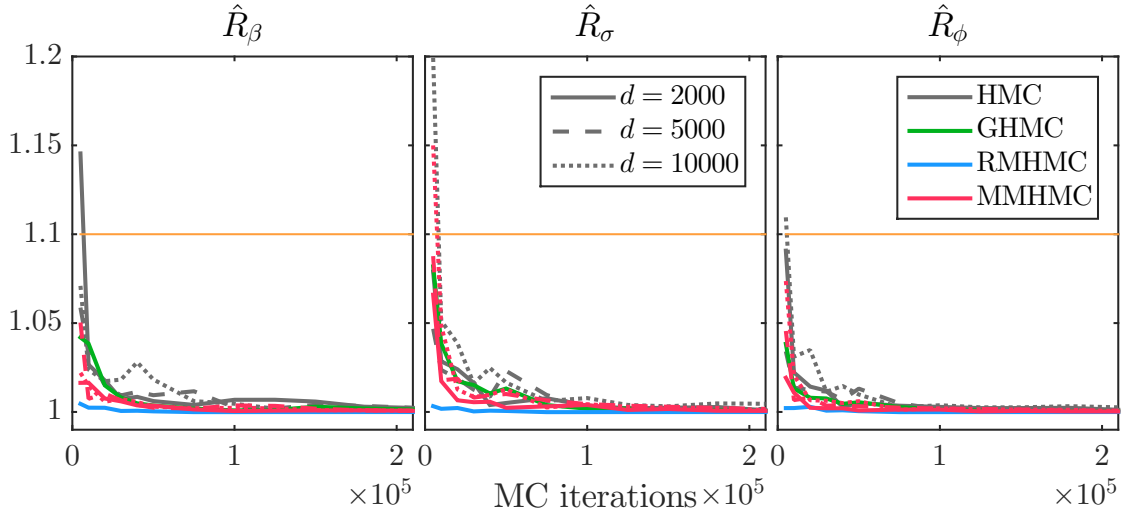


Figure 6.10: Convergence in terms of the potential scale reduction factor ($\hat{R}$) as a function Monte Carlo iterations for sampling the model parameters of the SV model.

Figures 6.11, 6.12 and 6.13 show sampling efficiency relative to HMC for experiments with $D = 2003, 5003, 10003$, respectively. Acceptance rates (shown in inset figures) are rather high for all methods. Nevertheless, there is no clear connection between obtained acceptance rates and ESS. Results demonstrate that all three methods, GHMC, RMHMC and MMHMC outperform HMC in terms of ESS. MMHMC and RMHMC show comparable performance – MMHMC is not more than 28% less efficient in sampling $\beta$ and latent variables than RMHMC and up to 35% more efficient than RMHMC in sampling $\sigma$ and $\phi$.

We recall here that in contrast to the RMHMC method, HMC, GHMC and MMHMC use the identity mass matrix. One way to improve the performance of these three methods compared to RMHMC would be to define the mass matrix from an estimate of global covariances in the warm-up phase and use it for obtaining the posterior samples.

We do not have access to the optimal parameters for RMHMC for dimensions higher than $D = 2003$. For $D = 5003, 10003$ we compare only MMHMC and HMC and observe that the superiority of MMHMC for sampling of model parameters and latent variables is

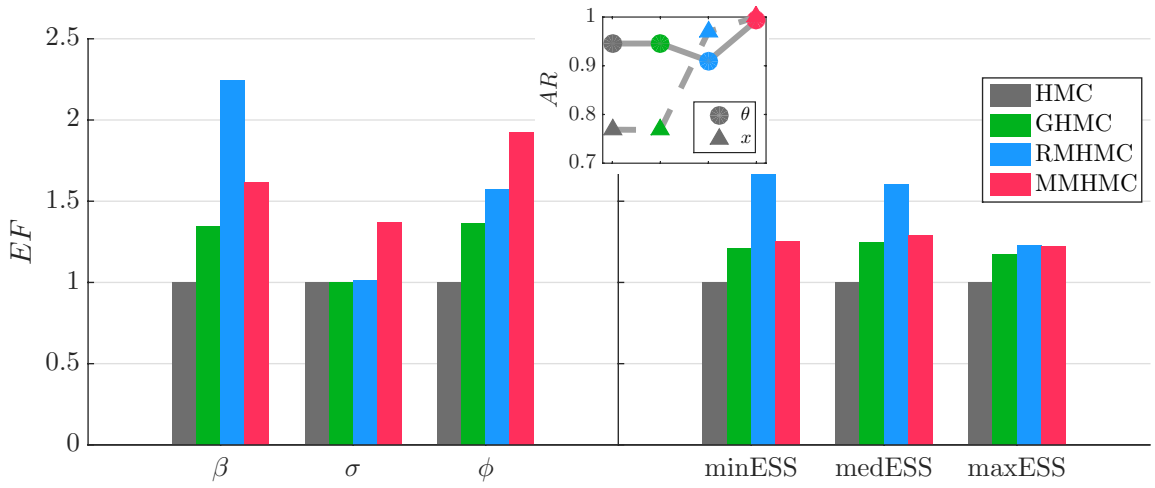


FIGURE 6.11: Sampling efficiency of GHMC, RMHMC and MMHMC relative to HMC for SV model parameters (left) and latent variables (right) and corresponding acceptance rates (inset) for dimension $D = 2003$.

maintained for higher dimensions.

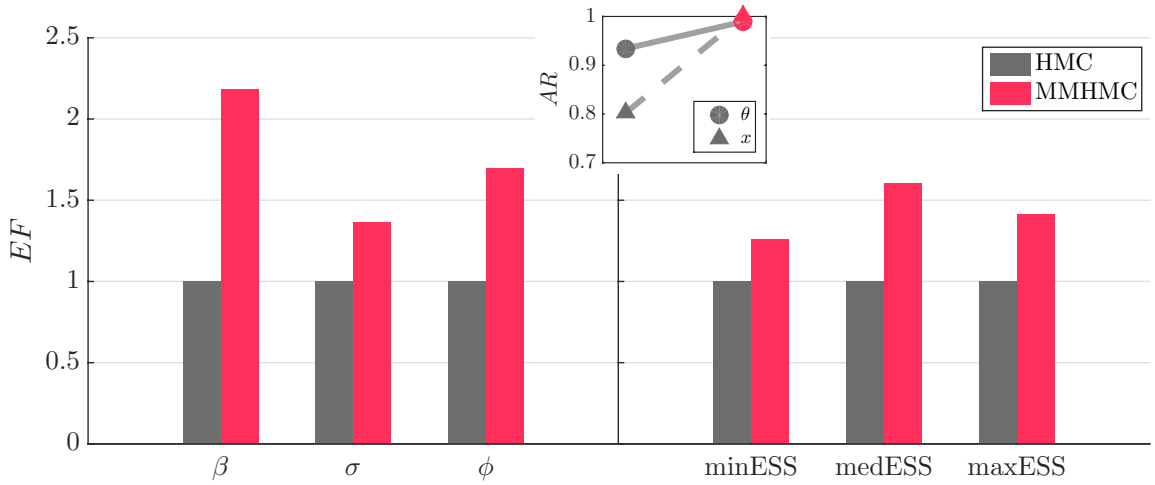


FIGURE 6.12: Sampling efficiency of MMHMC relative to HMC for SV model parameters (left) and latent variables (right) and corresponding acceptance rates (inset) for dimension $D = 5003$.

## 6.3 Summary

In this chapter we have examined performance of the MMHMC method on a set of standard benchmark statistical models and compared it with the popular sampling methods in computational statistics such as Random Walk Metropolis-Hastings (RWMH), Hamiltonian Monte Carlo (HMC), Generalized HMC (GHMC), Metropolis Adjusted Langevin Algorithm (MALA) and Riemann Manifold HMC (RMHMC).

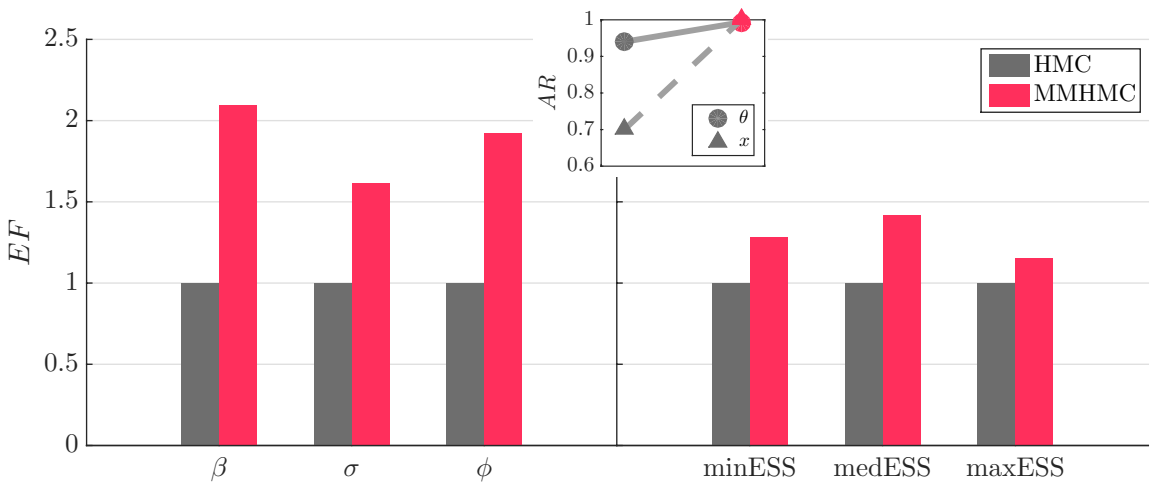


Figure 6.13: Sampling efficiency of MMHMC relative to HMC for SV model parameters (left) and latent variables (right) and corresponding acceptance rates (inset) for dimension $D = 10003$.

Space exploration has been inspected using an illustrative banana-shaped distribution. MMHMC accepts more proposals that result in better coverage of the space than with HMC. Although it uses the second-order information on the posterior, MMHMC does not follow its local curvature as obviously as it does RMHMC. Acceptance rate is higher for MMHMC than for other methods consistently for all experiments. Our tests demonstrate that in terms of sampling efficiency MMHMC and HMC perform comparably for small dimensional problems. However, the advantage of MMHMC over HMC increases with dimension – for a multivariate Gaussian problem MMHMC shows improvement of up to remarkable 20 times and for the BLR model up to 4 times. We expect even higher enhancement for problems of higher dimensions, as the new integrators specifically designed for MMHMC are particularly beneficial for high dimensional problems. The SV model experiments demonstrate the clear superiority of MMHMC and RMHMC over the HMC and GHMC methods. The sampling performance of MMHMC and RMHMC is comparable for this benchmark.

In addition to MMHMC performance evaluation with numerical experiments, in the beginning of this chapter we have proposed a new metric for ESS estimation. This metric is designed for MMHMC but can be applied for efficiency evaluation of any method that produces both correlated and weighted samples.

# 7

# Conclusions

## 7.1 Summary of contributions

Advances in Markov chain Monte Carlo (MCMC) methodology, together with increasing computing power, have led Bayesian inference becoming a widely used and powerful tool for dealing with complex statistical problems across a range of applications. The Bayesian approach provides a consistent and rigorous manner for dealing with uncertainty present in data and models, selection of a competing model among proposed and quantification of uncertainty in predictions. MCMC methods provide an arbitrarily level of accuracy in estimates by drawing as many samples from the target distribution as one requires. A very active research is being conducted in the development of efficient MCMC approaches but despite these efforts sampling in high dimensional problems and complex distributions, as appears in many problems with real data, remains a challenge. Besides, for problems that do not necessarily assume complex distributions but do involve large-scale simulations, e.g. geophysical, atmospheric, hydrodynamics models, it is crucial to reduce the number of forward model evaluations needed to achieve a particular level of accuracy in estimates.

In this thesis, we developed the methodology for enhanced statistical sampling, which offers higher sampling efficiency than the state-of-the-art MCMC method, Hamiltonian Monte Carlo. Our new approach, called Mix & Match HMC (MMHMC) arose as an extension of Generalized Shadow Hybrid Monte Carlo (GSHMC), proposed for molecular simulation, which has been published, patented and successfully tested on complex biological systems. The MMHMC introduces a number of modifications in GSHMC needed for efficient sampling in statistical applications. It can be defined as a generalized HMC importance sampler – momentum is updated in a general form and sampling is performed with respect to a modified density that is defined through modified Hamiltonians. To the best

of our knowledge, this is the first time that the method sampling with modified Hamiltonians has been implemented and applied to Bayesian inference problems in computational statistics.

We provided expressions for modified Hamiltonians of order 4 and 6 that can be readily employed within the newly derived splitting integrating schemes with two, three and four stages. In particular, the novel two-stage integrators derived in this thesis provide an outstanding improvement over the commonly used Verlet integrator that increases with the dimension of the problem. The improvement comes both in terms of acceptance rate and sampling efficiency over a range of simulation parameters. We also formulated and investigated different strategies for momentum update and momentum flip within the MMHMC method.

Being a method that generates both correlated and weighted samples, MMHMC requires a metric for sampling efficiency different from the one commonly used for MCMC. Here we suggested such a metric suitable for MCMC importance sampling based methods.

The method has been carefully tested and compared with the traditional and advanced sampling techniques for computational statistics such as Random Walk Metropolis-Hastings, Hamiltonian Monte Carlo, Generalized HMC, Riemann Manifold Hamiltonian Monte Carlo.

When compared to HMC and GHMC, the MMHMC method demonstrates superior performance, in terms of higher acceptance rate and bigger time-normalized ESS, for a range of applications, range of dimensions and choice of simulation parameters. It allows for bigger step sizes to be used without decreasing acceptance rate; moreover, it achieves better performance for larger step sizes. The improvements are more dramatic for high-dimensional problems – for a multivariate Gaussian problem MMHMC demonstrated an improvement over HMC of up to 40 times and for the BLR model up to 4 times. An additional advantage of MMHMC lays in the fact that it is less sensitive than HMC to the choice of a number of integration steps.

MMHMC and RMHMC demonstrate comparable sampling performance for the tested SV model. Nevertheless, in contrast to the original RMHMC, MMHMC does not require higher order derivative and inverse of the metric and thus is computationally less expensive. This issue becomes particularly important for high-dimensional problems with dense Hessian matrix. In addition, choices of integrators for RMHMC are limited due to the use of non-separable Hamiltonians, whereas MMHMC allows for the use of the novel efficient numerical integrators.

Several further extensions to the MMHMC method were designed in this thesis. These include the formulation of MMHMC for sampling of constrained variables, two algorithms for Bayesian adaptation of MMHMC simulation parameters, and Parallel tempering MMHMC offering efficient exploration of multimodal posterior distributions as well as estimation of the marginal likelihood.

The MMHMC method has been implemented in the in-house software package HaiCS (Hamiltonians in Computational Statistics), developed as a part of this thesis for statistical sampling of different models and distributions using Hamiltonian Monte Carlo based

methods.

MMHMC has been presented on following scientific events:

1. "Enhanced statistical sampling with GSHMC method", Radivojević T., Akhmatskaya E., Seminar Talk at Department of Statistics, University of Warwick, UK, May 5, 2015
2. "Employing modified Hamiltonians for sampling enhancement in statistical simulation", Radivojević T., Akhmatskaya E., International Conference on Scientific Computation and Differential Equations (SciCADE 2015), Potsdam, Germany, September 15, 2015, http://scicade2015.math.uni-potsdam.de/scicade2015/AbstractsBook_SciCADE2015.pdf
3. "Mix & Match Hamiltonian Monte Carlo", Akhmatskaya E., Radivojević T., 6th IMS-ISBA joint meeting, BayesComp at MCMSki V, Lenzerheide, Switzerland, January 5, 2016, http://www.pages.drexel.edu/~mwl25/mcmskiV/abstracts/Mix&MatchEA&TR.pdf
4. "Mix & Match Hamiltonian Monte Carlo", Akhmatskaya E., Radivojević T., ICMAT Workshop: Mathematical Perspectives in Biology, Madrid, Spain, February 3, 2016, https://www.icmat.es/congresos/2016/BBVA/BBVA-ICMAT-workshop_3-5Feb2016-final.pdf
5. "Hamiltonian Monte Carlo for high dimensional problems", Radivojević T., Akhmatskaya E., BCAM-IMUVA Summer School on Uncertainty Quantification for Applied Problems, Bilbao, Spain, July 7, 2016, http://www.bcamath.org/documentos_public/archivos/actividades_cientificas/Radivojevic_rev_.pdf
6. "Adaptive two-stage integrators for sampling algorithms based on Hamiltonian dynamics", Akhmatskaya E., Fernández-Pendás M., Radivojević T., Sanz-Serna J. M., ICERM Topical workshop Stochastic numerical algorithms, multiscale modeling and high-dimensional data analytics, ICERM, Brown University, RI, USA, July 21, 2016, https://icerm.brown.edu/materials/Slides/tw-16-5/Adaptive_two-stage_integrators_for_sampling_algorithms_based_on_Hamiltonian_dynamics_]_Elena_Akhmatskaya,_Basque_Center_for_Applied_Mathematics_-_BCAM.pdf

Development of enhanced sampling techniques in computational statistics was not the only research interest during my Ph.D. program. Besides, I had contributed to the implementation of the GSHMC method in the isobaric-isothermal statistical ensemble, which resulted in the following publication:

- Fernández Pendás M., Escribano B., **Radivojević T.**, Akhmatskaya E., *Constant pressure hybrid Monte Carlo simulations in GROMACS*, Journal of Molecular Modeling 20, 2487 (2014)

Another topic of my interest included a model development in finance and economy. In particular, we studied a simple model of the continuous double auction for high-frequency trading, and the results were published or submitted in:

- **Radivojević T.**, Anselmi J., Scalas E., *A stylized model for the continuous double auction*, Managing Market Complexity, Lecture Notes in Economics and Mathematical Systems 662, 115–125 (2012)
- **Radivojević T.**, Anselmi J., Scalas E., *Ergodic transition in a simple model of the continuous double auction*, PLoS ONE 9(2): e88095 (2014)
- Scalas E., Rapallo F., **Radivojević T.**, *Low-traffic limit and first-passage times for a simple model of the continuous double auction*, submitted

Furthermore, we studied the wealth distribution of economic agents using three different stochastic games and their combinations, resulting in publications:

- Garibaldi U., **Radivojević T.**, Scalas E., *Interplay of simple stochastic games as models for the economy*, Proceedings of Applications of Mathematics 2013, Institute of Mathematics, Academy of Sciences of the Czech Republic, Prague, 77–87 (2013)
- Scalas E., **Radivojević T.**, Garibaldi U., *Wealth distribution and the Lorenz curve: A finitary approach*, Journal of Economic Interaction and Coordination 10(1), 79–89 (2015)

More details on these models can be found in Appendix.

I had also participated in two industrial projects which resulted in two technical reports:

- **Radivojević T.**, Fernández Pendás M., Akhmatskaya E., Technical Report for the industrial project within the Math-in framework, (confidential) (2014)
- Arran M., Benham G., Dempsey L., Dubrovina E., Feier R., Fozard J., Lambert A., Maestri J., Miyajima N., **Radivojević T.**, Riley E., *Represent the Degree of Mimicry between Prosodic Behaviour of Speech Between Two or More People*, ESGI107 Technical Report (2015)

## 7.2 Ongoing and future work

The MMHMC method has a wide scope for further research. For example, we are already working on techniques that combine beneficial features of MMHMC with manifold methods, RMHMC and MMALA (Girolami and Calderhead, 2011b). These techniques rely on different integrators, and so different modified Hamiltonians, than in MMHMC. On the other hand, we are currently working on the development and testing of system-specific adaptive integrators to be used for sampling with the MMHMC method.

MMHMC can be extended in many other directions. We would like to adapt and implement new techniques for further performance and flexibility enhancing of MMHMC. These include alternative approaches for parameter adaptation, different techniques for efficiency improvement such as delayed rejections, zero-variance and quasi Monte Carlo, approximate computations based on stochastic gradients or proximal MCMC aiming to reduce the computational cost, alternative ways of calculating as well as making use of the second order information, extensions to particle filters and parallelization.

# A

# Appendix

## A.1 Contributions to model development

### A.1.1 Continuous double auction

**Radivojević T., Anselmi J., Scalas E., *A stylized model for the continuous double auction*, Managing Market Complexity, Lecture Notes in Economics and Mathematical Systems 662, 115–125 (2012)**

**Abstract:** A stylized phenomenological model for the continuous double auction is introduced. This model is equivalent to two uncoupled M/M/1 queues. The conditions for statistical equilibrium (ergodicity) are derived. The results of Monte Carlo simulations are presented on the behaviour of price differences and log-returns.

**Radivojević T., Anselmi J., Scalas E., *Ergodic transition in a simple model of the continuous double auction*, PLoS ONE 9(2): e88095. (2014)**

**Abstract:** We study a phenomenological model for the continuous double auction, whose aggregate order process is equivalent to two independent M/M/1 queues. The continuous double auction defines a continuous-time random walk for trade prices. The conditions for ergodicity of the auction are derived and, as a consequence, three possible regimes in the behavior of prices and logarithmic returns are observed. In the ergodic regime, prices are unstable and one can observe a heteroskedastic behavior in the logarithmic returns. On the contrary, non-ergodicity triggers stability of prices, even if two different regimes can be seen.

**Scalas E., Rapallo F., Radivojević T., *Low-traffic limit and first-passage times for a simple model of the continuous double auction*, submitted**

**Abstract:** We consider a simplified model of the continuous double auction where prices are integers varying from 1 to $N$ with limit orders and market orders, but quantity per order limited to a single share. For this model, the order process is equivalent to two M/M/1 queues. We study the behaviour of the auction in the low-traffic limit where limit orders are immediately transformed into market orders. In this limit, the distribution of prices can be computed exactly and gives a reasonable approximation of the price distribution when the ratio between the rate of order arrivals and the rate of order executions is below 1/2. This is further confirmed by the analysis of the first passage time in 1 or $N$.

### A.1.2 Wealth distribution

**Garibaldi U., Radivojević T., Scalas E., *Interplay of simple stochastic games as models for the economy*, Proceedings of Applications of Mathematics 2013, Institute of Mathematics, Academy of Sciences of the Czech Republic, Prague, 77–87 (2013)**

**Abstract:** Using the interplay among three simple exchange games, one may give a satisfactory representation of a conservative economic system where total wealth and number of agents do not change in time. With these games it is possible to investigate the emergence of statistical equilibrium in a simple pure-exchange environment. The exchange dynamics is composed of three mechanisms: a decentralized interaction, which mimics the pair-wise exchange of wealth between two economic agents, a failure mechanism, which takes into account occasional failures of agents and includes wealth redistribution favoring richer agents, and a centralized mechanism, which describes the result of a redistributive effort. According to the interplay between these three mechanisms, their relative strength, as well as the details of redistribution, different outcomes are possible.

**Scalas E., Radivojević T., Garibaldi U., *Wealth distribution and the Lorenz curve: A finitary approach*, Journal of Economic Interaction and Coordination 10(1), 79–89 (2015)**

**Abstract:** We use three stochastic games for the wealth of economic agents which may be at work in a real economy and we derive their statistical equilibrium distributions. Based on a heuristic argument, we assume that the expected observed wealth distribution is a mixture of these three distributions. We compare the Lorenz curves obtained from this conjecture with the empirical curves for a set of countries.

## A.2 Contributions to algorithm development

**"Momentum flips in generalized hybrid/Hamiltonian Monte Carlo methods", Radivojevic T., Akhmatskaya E., Seminar at Department of Applied Mathematics, Faculty of Sciences, University of Valladolid, Spain, January 22, 2014, `http://www.imuva.uva.es/en/actividades/ver/93`**

**Abstract:** Generalized hybrid / Hamiltonian Monte Carlo (GHMC) methods differ from hybrid / Hamiltonian Monte Carlo (HMC) techniques in the momentum update step, where a partial refreshment of momentum replaces a complete momentum reset. In order to satisfy detailed balance condition and ensure a stationary distribution, a momentum flip is required upon rejection of a Hamiltonian dynamics proposal step. These momentum swings induce reverse trajectories and might, in principle, slow down mixing and decorrelation of the chain. In this talk we analyze the effect of momentum flips on efficiency and accuracy of several versions of GHMC applied to molecular and statistical simulations and discuss possible ways for reducing potential negative effects of momentum flips.

**Fernández Pendás M., Escribano B., Radivojević T., Akhmatskaya E., *Constant pressure hybrid Monte Carlo simulations in GROMACS*, Journal of Molecular Modeling 20, 2487 (2014)**

**Abstract:** Adaptation and implementation of the Generalized Shadow Hybrid Monte Carlo (GSHMC) method for molecular simulation at constant pressure in the NPT ensemble are discussed. The resulting method, termed NPT-GSHMC, combines Andersen barostat with GSHMC to enable molecular simulations in the environment natural for biological applications, namely, at constant pressure and constant temperature. Generalized Hybrid Monte Carlo methods are designed to maintain constant temperature and volume and extending their functionality to preserving pressure is not trivial. The theoretical formulation of NPT-GSHMC was previously introduced. Our main contribution is the implementation of this methodology in the GROMACS molecular simulation package and the evaluation of properties of NPT-GSHMC, such as accuracy, performance, effectiveness for real physical systems in comparison with well-established molecular simulation techniques. Benchmarking tests are presented and the obtained preliminary results are promising. For the first time, the generalized hybrid Monte Carlo simulations at constant pressure are available within the popular open source molecular dynamics software package.

**"Employing modified Hamiltonians for sampling enhancement in statistical simulation", Radivojević T., Akhmatskaya E., International Conference on Scientific Computation and Differential Equations (SciCADE 2015), Potsdam, Germany, September 15, 2015, `http://scicade2015.math.uni-potsdam.de/scicade2015/`**

**Abstract:** Sampling with modified (shadow) Hamiltonians in hybrid Monte Carlo methods can dramatically improve efficiency of molecular simulation at different scales compared

with conventional molecular dynamics and hybrid Monte Carlo simulations. We introduce modified Hamiltonians in Hamiltonian Monte Carlo for enhancing sampling in statistical simulation, and demonstrate advantages of the proposed method in different statistical models through a comparison with well established Hamiltonian Monte Carlo based methods.

**"Mix & Match Hamiltonian Monte Carlo", Akhmatskaya E., Radivojević T., ICMAT Workshop: Mathematical Perspectives in Biology, Madrid, Spain, February 3, 2016, `https://www.icmat.es/congresos/2016/BBVA/BBVA-ICMAT-workshop_3-5Feb2016-final.pdf`**

**Abstract:** Hamiltonian (Hybrid) Monte Carlo (HMC) method, initially proposed in High Energy Physics, is becoming a popular tool for solving complex and intractable problems of statistical inference. We introduce multiple modifications in the original formulation of the HMC in order to enhance sampling from high-dimensional or strongly correlated target densities. The new features include the modified Metropolis test, the updated momentum refreshment step, the novel numerical integrating scheme. All alterations have been formulated and implemented within the Generalized Shadow Hybrid Monte Carlo framework, earlier proposed by the authors for simulation of molecular systems. The sampling efficiency of the resulting method is assessed by performing inference on standard statistical benchmark models, and compared with Random Walk Metropolis-Hastings, the original Hamiltonian Monte Carlo and Riemann Manifold Hamiltonian Monte Carlo methods.

**"Adaptive two-stage integrators for sampling algorithms based on Hamiltonian dynamics", Akhmatskaya E., Fernández-Pendás M., Radivojević T., Sanz-Serna J. M., ICERM Topical workshop Stochastic numerical algorithms, multiscale modeling and high-dimensional data analytics, ICERM, Brown University, RI, USA, July 21, 2016, `https://icerm.brown.edu/materials/Slides/tw-16-5/Adaptive_two-stage_integrators_for_sampling_algorithms_based_on_Hamiltonian_dynamics_]_Elena_Akhmatskaya,_Basque_Center_for_Applied_Mathematics_-_BCAM.pdf`**

**Abstract:** We present an alternative to the standard velocity Verlet integrator, known to be the state-of-the- art method for numerical integration of the Hamiltonian equations in molecular dynamics (MD) and hybrid / Hamiltonian Monte Carlo (HMC) simulations.

The novel methodology, which we call the Adaptive Integration Approach, or AIA, offers, for any chosen simulation problem and step size, a system-specific two-stage splitting integrator, which provides the best conservation of energy for harmonic forces. The proposed new family of numerical integrators can be viewed as a one-parameter two-stage splitting integrators family, with the parameter being a function of the simulation step size and the highest angular frequency present in the simulated system. In contrast, all numerical integrators for Hamiltonian dynamics used to date belong to the fixed parameters families.

The AIA has been formulated for a range of algorithms, which simulate either constrained or unconstrained dynamics, and sample with Hamiltonians or modified Hamiltonians. It can be implemented in a MD / HMC software code, without introducing computational overheads in the simulations.

Numerical tests show that the method successfully realises the fail-safe strategy. In all experiments, and for each of the criteria employed, the AIA is at least as good as, and often significantly outperforms the standard Verlet scheme, as well as fixed parameter, optimized two- stage integrators.

The ideas underlying the AIA can be also used for a rational choice of simulation parameters.

# Bibliography


Akhmatskaya, E. and S. Reich (2006). "The Targeted Shadowing Hybrid Monte Carlo (TSHMC) Method". In: *New Algorithms for Macromolecular Simulation, Lecture Notes in Computational Science and Engineering*. Vol. 49. Berlin: Springer-Verlag, pp. 141–153 (34).

— (2008). "GSHMC: An efficient method for molecular simulation". In: *Journal of Computational Physics* 227.10, pp. 4934–4954. DOI: http://dx.doi.org/10.1002/andp.19053221004 (vi, xiii, 34–36, 41, 57, 70, 73, 81, 110).

— (2011). "Meso-GSHMC: A stochastic algorithm for meso-scale constant temperature simulations". In: *Procedia Computer Science* 4, pp. 1353–1362 (35, 40).

— (2012). "New Hybrid Monte Carlo Methods for Efficient Sampling: from Physics to Biology and Statistics". In: *Progress in Nuclear Science and Technology* 2, pp. 447–462 (35, 40, 41).

Akhmatskaya, E., N. Bou-Rabee, and S. Reich (2009a). "A comparison of generalized hybrid Monte Carlo methods with and without momentum flip". In: *Journal of Computational Physics* 228, pp. 2256–2265 (75).

— (2009b). "Erratum to "A comparison of generalized hybrid Monte Carlo methods with and without momentum flip"". In: *Journal of Computational Physics* 228, pp. 7492–7496 (75).

Akhmatskaya, E., S. Reich, and R. Nobes (2009c). *Method, apparatus and computer program for molecular simulation*. GB patent (published) (35, 36).

Akhmatskaya, E., R. Nobes, and S. Reich (2011). *Method, apparatus and computer program for molecular simulation*. US patent (granted) (35).

Akhmatskaya, E., T. van Mourik, H. Früchtl, A. Heidenreich, K. Rademann, F. Emmerling, and E. Rössler (2013). "Computational study of polymorphism in drugs". In: *HPC-Europa Annual Report Book*, pp. 994–997 (35).

Andrieu, C., N. De Freitas, A. Doucet, and M. I. Jordan (2003). "An introduction to MCMC for machine learning". In: *Machine Learning* 50.1-2, pp. 5–43 (10).

Arnold, V. I. (1989). *Mathematical Methods of Classical Mechanics*. 2nd ed. New York Springer-Verlag (15).

Asua, J. M. and E. Akhmatskaya (2011). "Dynamical modelling of morphology development in multiphase latex particles". In: *European Success Stories in Industrial Mathematics*. Ed. by T. Lery et al. Springer (40).

Attia, A. and A. Sandu (2015). “A hybrid Monte Carlo sampling filter for non-gaussian data assimilation”. In: *AIMS Geosciences* 1, pp. 41–78 (27).

Bayes, T. and R. Price (1763). “An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFRS”. In: *Philosophical Transactions (1683–1775)* 53, pp. 370–418. URL: `http://www.jstor.org/stable/105741` (3).

Bernardo, J. M. and A. F. M. Smith (1994). *Bayesian Theory*. Wiley (5).

Beskos, A., F. J. Pinski, J. M. Sanz-Serna, and A. M. Stuart (2011). “Hybrid Monte Carlo on Hilbert spaces”. In: *Stochastic Processes and their Applications* 121.10, pp. 2201–2230 ( 24, 30).

Beskos, A., N. S. Pillai, G. O. Roberts, J. M. Sanz-Serna, and A. M. Stuart (2013). “Optimal tuning of the hybrid Monte Carlo algorithm”. In: *Bernoulli* 19.5A, pp. 1501–1534 (23).

Betancourt, M. (2010). “Nested Sampling with Constrained Hamiltonian Monte Carlo”. In: *arXiv:1005.0157v1* (30).

— (2013a). “A general metric for Riemannian manifold Hamiltonian Monte Carlo”. In: *Geometric Science of Information*. Springer, pp. 327–334 (29).

— (2013b). “Generalizing the No-U-Turn Sampler to Riemannian Manifolds”. In: *arXiv:1304.1920v1* (29).

— (2014). “Adiabatic Monte Carlo”. In: *arXiv:1405.3489v4* (30).

— (2016). “Identifying the Optimal Integration Time in Hamiltonian Monte Carlo”. In: *arXiv preprint arXiv:1601.00225* (28).

Betancourt, M. and M. Girolami (2015). “Hamiltonian Monte Carlo for hierarchical models”. In: *Current Trends in Bayesian Methodology with Applications* 79 (30).

Betancourt, M., S. Byrne, and M. Girolami (2014). “Optimizing The Integrator Step Size for Hamiltonian Monte Carlo”. In: *arXiv:1411.6669v2* (23).

Betancourt, M., S. Byrne, S. Livingstone, and M. Girolami (2016). “The Geometric Foundations of Hamiltonian Monte Carlo”. In: *Bernoulli* (20).

Blanes, S., F. Casas, and J. M. Sanz-Serna (2014). “Numerical integrators for the Hybrid Monte Carlo method”. In: *SIAM Journal on Scientific Computing* 36.4, A1556–A1580 ( 26, 27, 49, 57, 59, 62–64, 75).

Bou-Rabee, N. and J. M. Sanz-Serna (2015). “Randomized Hamiltonian Monte Carlo”. In: *arXiv preprint arXiv:1511.09382* (29).

Brooks, S. P. and A. Gelman (1998). “General methods for monitoring convergence of iterative simulations”. In: *Journal of Computational and Graphical Statistics* 7, pp. 434–455 (109).

Brooks, S., A. Gelman, G. Jones, and X.-L. Meng, eds. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press (vi, xii, 10).

Brubaker, M. A., M. Salzmann, and R. Urtasun (2012). “A Family of MCMC Methods on Implicitly Defined Manifolds”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 161–172 (30).

Calderhead, B. (2014). “A general construction for parallelizing Metropolis-Hastings algorithms”. In: *Proceedings of the National Academy of Sciences of the United States of America* 111.49, pp. 17408–17413. DOI: doi/10.1073/pnas.1408184111 (30).

Calderhead, B. and M. Girolami (2009). “Estimating Bayes factors via thermodynamic integration and population MCMC”. In: *Computational Statistics and Data Analysis* 53, pp. 4028–4045 (97).

Campos, C. M. and J. M. Sanz-Serna (2015). “Extra chance generalized hybrid Monte Carlo”. In: *Journal of Computational Physics* 281, pp. 365–374 (30, 75).

Chao, W.-L., J. Solomon, D. L. Michels, and F. Sha (2015). “Exponential Integration for Hamiltonian Monte Carlo”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37 (24, 30).

Chen, L., Z. Qin, and J. S. Liu (2000). “Exploring Hybrid Monte Carlo in Bayesian Computation”. In: *ISBA 2000, Proceedings* (123).

Chen, T., E. B. Fox, and C. Guestrin (2014). “Stochastic Gradient Hamiltonian Monte Carlo”. In: *Proceedings of the 31st International Conference on Machine Learning, Beijing, China* (30).

*Data Scientist: The Sexiest Job of the 21st Century* (2012). URL: https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century (2).

Dongarra, J. and F. Sullivan (2000). “Guest editors’ introduction: The top 10 algorithms”. In: *Computing in Science & Engineering* 2.1, pp. 22–23 (9).

Duane, S., A. D. Kennedy, B. J. Pendleton, and D. Roweth (1987). “Hybrid Monte Carlo”. In: *Physics Letters B* 195.2, pp. 216–222 (vi, xii, 10, 20).

Earl, D. J. and M. W. Deem (2005). “Parallel tempering: Theory, applications, and new perspectives”. In: *Physical Chemistry Chemical Physics* 7.3910–3916 (92, 95).

Engle, R. D., R. D. Skeel, and M. Drees (2005). “Monitoring energy drift with shadow Hamiltonians”. In: *Journal of Computational Physics* 206, pp. 432–452. DOI: doi:10.1016/j.jcp.2004.12.009 (49).

Escribano, B., E. Akhmatskaya, and J. I. Mujika (2013). “Combining stochastic and deterministic approaches within high efficiency molecular simulations”. In: *Central European Journal of Mathematics* 11.4, pp. 787–799 (35).

Escribano, B., E. Akhmatskaya, S. Reich, and J. M. Azpiroz (2014). “Multiple-time-stepping generalized hybrid Monte Carlo methods”. In: *Journal of Computational Physics* (35).

Fang, Y., J. M. Sanz-Serna, and R. D. Skeel (2014). “Compressible generalized hybrid Monte Carlo”. In: *The Journal of Chemical Physics* 140.17, p. 174108 (15, 86).

Fernández-Pendás, M., B. Escribano, T. Radivojević, and E. Akhmatskaya (2014). “Constant pressure hybrid Monte Carlo simulations in GROMACS”. In: *Journal of Molecular Modeling* 20.12, pp. 1–10 (35).

Frenkel, D. (1986). “Free-energy computation and first-order phase transitions”. In: *Molecular-Dynamics Simulation of Statistical-Mechanical systems*. Ed. by G. Ciccoti and W. G. Hoover. Vol. 97. North Holland, Amsterdam, pp. 151–188 (96).

Friel, N. and J. Wyse (2012). “Estimating the model evidence: a review”. In: *Statistica Neerlandica* 63.3, pp. 288–308 (96).

Friel, N. and A. N. Pettitt (2008). “Marginal likelihood estimation via power posteriors”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.3, pp. 589–607 (96).

Galassi, M., J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi (2009). “GNU Scientific Library Reference Manual (Network Theory Ltd., 2009)”. In: *URL http://www.gnu. org/s/gsl* (103).

Gelfand, A. E. and A. F. Smith (1990). “Sampling-based approaches to calculating marginal densities”. In: *Journal of the American Statistical Association* 85.410, pp. 398–409 (7).

Gelman, A. and D. B. Rubin (1992). “Inference from iterative simulation using multiple sequences”. In: *Statistical Science* 7.457-511 (109).

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2003). *Bayesian Data Analysis*. 2nd ed. Chapman & Hall / CRC (v, xi, 4, 5).

Gelman, A. and X.-L. Meng (1998). “Simulating normalizing constants: From importance sampling to bridge sampling to path sampling”. In: *Statistical Science*, pp. 163–185 ( 96).

Geman, S. and D. Geman (1984). “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images”. In: *IEEE Trans. Pattern Anal. Machine Intelligence* 6, pp. 721–741. URL: `http://www.stat.cmu.edu/~acthomas/724/Geman.pdf` (7, 9).

Geweke, J (1992). “Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments”. In: *Bayesian Statistics* 4, pp. 169–188 (110).

Geyer, C. J. (1991). “Markov chain Monte Carlo maximum likelihood”. In: *Computing Science and Statistics, ed. E. M. Keramides (Interface Foundation, Fairfax Station, Va.)* (92).

— (1992). “Practical Markov Chain Monte Carlo”. In: *Statistical Science,* 7.4, pp. 473–483 (8, 46, 108, 111, 112).

Girolami, M. and B. Calderhead (2011a). *Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods, Matlab code*. URL: `http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic-research/girolami/rmhmc/Stoch_Vol.zip` (127).

Girolami, M. and B. Calderhead (2011b). “Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.2, pp. 123–214 (29, 113, 120, 123, 134).

Gramacy, R., R. Samworth, and R. King (2010). “Importance tempering”. In: *Statistics and Computing* 20.1, pp. 1–7 (111).

Green, P. J., K. Łatuszyński, M. Pereyra, and C. P. Robert (2015). “Bayesian computation: a summary of the current state, and samples backwards and forwards”. In: *Statistics and Computing* 25.4, pp. 835–862 (2, 5).

Gupta, S., A. Irb ack, F. Karsch, and B. Petersson (1990). “The acceptance probability in the hybrid Monte Carlo method”. In: *Physics Letters B* 242.437–443 (23).

Hairer, E., C. Lubich, and G. Wanner (2006). *Geometric Numerical Integration*. 2nd ed. Springer-Verlag (13, 49).

Hamze, F., N. Dickson, and K. Karimi (2010). “Robust parameter selection for parallel tempering”. In: *International Journal of Modern Physics C* 21.05, pp. 603–615 (95).

Hanson, K. M. (2002). “Use of probability gradients in hybrid MCMC and a new convergence test”. In: *Los Alamos Report LA-UR-02-4105* (110).

Hastings, W. K. (1970). “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57.1, pp. 97–109 (vi, xii, 9).

Hoffman, M. D. and A. Gelman (2014). “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: *Journal of Machine Learning Research* 15, pp. 1593–1623 (28, 29, 114).

Horowitz, A. M. (1991). “A generalized guided Monte Carlo algorithm”. In: *Physics Letters B* 268, pp. 247–252 (31).

Hukushima, K. and Y. Nemoto (1996). “Exchange Monte Carlo method and application to spin glass simulations”. In: *Journal of the Physical Society of Japan* 65 (92).

Hull, J. C. and A. White (1987). “The Pricing of Options on Assets with Stochastic Volatilities”. In: *Journal of Finance* 42, pp. 281–300 (122).

Izaguirre, J. A. and S. S. Hampton (2004). “Shadow hybrid Monte Carlo: an efficient propagator in phase space of macromolecules”. In: *Journal of Computational Physics* 200, pp. 581–604 (vi, xii, 34, 42, 73, 110).

Jacquier, E., N. G. Polson, and P. E. Rossi (1994). “Bayesian Analysis of Stochastic Volatility Models”. In: *Journal of Business & Economic Statistics* 12.4 (123).

Jeffreys, H. (1961). *Theory of Probability*. Third. Oxford University Press (4).

Kahn, H. and A. W. Marshall (1953). “Methods of Reducing Sample Size in Monte Carlo Computations”. In: *Journal of the Operations Research Society of America* 1.5, pp. 263–278 (6).

Kass, R. E. and A. E. Raftery (1995). “Bayes factors”. In: *Journal of the American Statistical Association* 90.430, pp. 773–795 (4).

Katzgraber, H. G., S. Trebst, D. A. Huse, and M. Troyer (2006). “Feedback-optimized parallel tempering monte carlo”. In: *Journal of Statistical Mechanics: Theory and Experiment* 3 ( 95).

Kennedy, A. D. and B. Pendleton (2001). “Cost of the Generalised Hybrid Monte Carlo Algorithm for Free Field Theory”. In: *Nuclear Physics B* 607 (3), pp. 456–510. DOI: 10.1016/S0550-3213(01)00129-8 (23, 31, 33).

Kennedy, A. D. (1990). “The theory of hybrid stochastic algorithms”. In: *Probabilistic Methods in Quantum Field Theory and Quantum Gravity*. Springer, pp. 209–223 (31).

Kim, S., N. Shephard, and S. Chib (1998). “Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models”. In: *Review of Economic Studies* 65, pp. 361–393 (122).

Kong, A., J. S. Liu, and W. H. Wong (1994). “Sequential Imputations and Bayesian Missing Data Problems”. In: *Journal of the American Statistical Association* 89.425, pp. 278–288 (111).

Lan, S., J. Streets, and B. Shahbaba (2014a). “Wormhole Hamiltonian Monte Carlo”. In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (30, 91).

Lan, S., V. Stathopoulos, B. Shahbaba, and M. Girolami (2015). “Lagrangian Dynamical Monte Carlo”. In: *Journal of Computational and Graphical Statistics* 24.2 (29, 113).

Lan, S. and B. Shahbaba (2012). “Split HMC for Gaussian Process Models”. In: *arXiv preprint arXiv:1201.3973* (24).

Lan, S., B. Zhou, and B. Shahbaba (2014b). “Spherical Hamiltonian Monte Carlo for Constrained Target Distributions”. In: *Proceedings of the 31st International Conference on Machine Learning*, pp. 629–637 (30).

Laplace, P. de (1820). “Theorie analytique des probabilites”. In: *M.V. Courcier* (3).

Leimkuhler, B. and S. Reich (2005). *Simulating Hamiltonian Dynamics*. Cambridge, UK: Cambridge University Press (13).

Lichman, M. (2013). *UCI Machine Learning Repository*. URL: `http://archive.ics.uci.edu/ml` (120).

Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing*. New York: Springer Series in Statistics, Springer. URL: `https://books.google.es/books?id=jwrSBwAAQBAJ&printsec=frontcover&dq=monte+carlo+strategies+in+scientific+computing&hl=en&sa=X&ved=0ahUKEwjB8Jjyg7bLAhWLtBQKHRPXAjUQ6AEIHTAA#v=onepage&q=monte%20carlo%20strategies%20in%20scientific%20computing&f=false` (10, 29, 123).

Liu, J. S., F. Liang, and W. H. Wong (2000). “Multiple-Try Method and Local Optimization in Metropolis Sampling”. In: *Journal of the American Statistical Association* 95.449, pp. 121–134 (73).

Livingstone, S., M. Betancourt, S. Byrne, and M. Girolami (2016). “On the geometric ergodicity of Hamiltonian Monte Carlo”. In: *arXiv:1601.08057v1* (20, 29).

Mackenzie, P. B. (1989). “An improved hybrid Monte Carlo method”. In: *Physics Letters B* 226, pp. 369–371 (29).

Mahendran, N., Z. Wang, F. Hamze, and N. De Freitas (2012). “Adaptive MCMC with Bayesian Optimization”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 22, pp. 751–760 (87).

McLachlan, R. I. and P. Atela (1992). “The accuracy of symplectic integrators”. In: *Nonlinearity* 5.541-562 (26).

McLachlan, R. I. (1995). “On the numerical integration of ordinary differential equations by symmetric composition methods”. In: *SIAM Journal on Scientific Computing* 16.1, pp. 151–168 (26, 27, 57, 58).

Meeds, E., R. Leenders, and M. Welling (2015). “Hamiltonian ABC”. In: *arXiv:1503.01916v1* (30).

Meent, J.-W. van de, B. Paige, and F. Wood (2014). “Tempering by Subsampling”. In: *arXiv:1401.7145v1* (30).

Metropolis, N. and S. Ulam (1949). “The Monte Carlo method”. In: *Journal of the American Statistical Association* 44, pp. 335–341 (v, xi, 5).

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, M. Teller, and E. Teller (1953). “Equation of State Calculations by Very Fast Computing Machines”. In: *Journal of Chemical Physics* 21, pp. 1087–1092 (vi, xii, 7, 9, 10).

Metropolis, N. (1987). “The beginning of the Monte Carlo method”. In: *Los Alamos Science* 15.584, pp. 125–130 (5).

Mira, A. (2001). “On Metropolis-Hastings algorithms with delayed rejection”. In: *Metron* 59, pp. 231–241 (30).

Moan, P. C. and J. Niesen (2014). “On an asymptotic method for computing the modified energy for symplectic methods”. In: *Dynamical Systems* 34.3, pp. 1105–1120 (49).

Murua, A and J. Sanz-Serna (1999). “Order conditions for numerical integrators obtained by composing simpler integrators”. In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 357.1754, pp. 1079–1100 ( 20).

Neal, R. M. (1994). “Bayesian Learning for Neural Networks”. PhD thesis. Dept. of Computer Science, University of Toronto (10, 20).

— (2001). “Annealed Importance Sampling”. In: *Statistics and Computing* 11, pp. 125–139 (96, 111).

— (2011). “MCMC using Hamiltonian dynamics”. In: *Handbook of Markov Chain Monte Carlo*. Vol. 2. Chapman & Hall / CRC Press, pp. 113–162 (vi, xii, 20, 112).

Newton, M. A. and A. E. Raftery (1994). “Approximate Bayesian inference with the weighted likelihood bootstrap”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 56.1, pp. 3–48 (96).

Nishimura, A. and D. B. Dunson (2015). “Recycling intermediate steps to improve Hamiltonian Monte Carlo”. In: *arXiv:1511.06925v1* (31).

Omelyan, I. P., I. M. Mryglod, and R. Folk (2002). “Construction of high-order force-gradient algorithms for integration of motion in classical and quantum systems”. In: *Physical Review E* 66.2 (49, 50).

Ottobre, M., N. S. Pillai, F. J. Pinski, and A. M. Stuart (2016). “A function space HMC algorithm with second order Langevin diffusion limit”. In: *Bernoulli* 22.1, pp. 60–106 ( 24).

Pakman, A. and L. Paninski (2013). “Auxiliary-variable Exact Hamiltonian Monte Carlo Samplers for Binary Distributions”. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2490–2498 (30).

— (2014). “Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians”. In: *Journal of Computational and Graphical Statistics* 23.2, pp. 518–542 (24).

Pasarica, C. and A. Gelman (2010). “Adaptively scaling the Metropolis algorithm using expected squared jumped distance”. In: *Statistica Sinica* 20.1 (91).

Plummer, M., N. Best, K. Cowles, and K. Vines (2006). “CODA: Convergence Diagnosis and Output Analysis for MCMC”. In: *R News* 6.1, pp. 7–11 (100, 109).

Predescu, C., M. Predescu, and C. V. Ciobanu (2004). “The incomplete beta function law for parallel tempering sampling of classical canonical systems”. In: *The Journal of Chemical Physics* 120.9, pp. 4119–4128 (95).

Rimoldini, L. (2014). “Weighted skewness and kurtosis unbiased by sample size and Gaussian uncertainties”. In: *Astronomy and Computing* 5, pp. 1–8 (111).

Robert, C. and G. Casella (2005). *Monte Carlo Statistical Methods*. Springer New York (6, 10).

Roberts, G. O. and J. S. Rosenthal (2007). "Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms". In: *Journal of Applied Probability*, pp. 458–475 (89).

Salvatier, J., T. V. Wiecki, and C. Fonnesbeck (2016). "Probabilistic programming in Python using PyMC3". In: *PeerJ Computer Science* 2, e55 (99).

Sanz-Serna, J. M. and M. Calvo (1994). *Numerical Hamiltonian Problems*. London: Chapman and Hall (13, 19, 20, 49, 58).

Schuster, I. (2015). "Gradient Importance Sampling". In: *arXiv:1507.05781v1* (110).

Shahbaba, B., S. Lan, W. O. Johnson, and R. M. Neal (2014). "Split Hamiltonian Monte Carlo". In: *Statistics and Computing* 24, pp. 339–349 (24, 30).

Skeel, R. D. and D. J. Hardy (2001). "Practical construction of modified Hamiltonians". In: *SIAM Journal on Scientific Computing* 23.4, pp. 1172–1188 (34, 49).

Sohl-Dickstein, J. (2012). "Hamiltonian Monte Carlo with Reduced Momentum Flips". In: *arXiv:1205.1939v1* (33, 75).

Sohl-Dickstein, J. and B. J. Culpepper (2012). "Hamiltonian Annealed Importance Sampling for partition function estimation". In: *arXiv preprint arXiv:1205.1925* (30, 96).

Sohl-Dickstein, J., M. Mudigonda, and M. Deweese (2014). "Hamiltonian Monte Carlo Without Detailed Balance". In: *Proceedings of the 31st International Conference on Machine Learning*, pp. 719–726 (30, 33, 75).

Srinivas, N., A. Krause, M. Seeger, and S. M. Kakade (2010). "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design". In: *Proceedings of the 27th International Conference on Machine Learning*, pp. 1015–1022 (91).

Stan Development Team (2016). *Stan Modeling Language Users Guide and Reference Manual*. Version 2.9.0 (20, 29, 85, 99, 109).

Statisticat LLC (2013). "LaplacesDemon: Complete Environment for Bayesian Inference". In: *R package version* 13.4. URL: `https://web.archive.org/web/20141224051720/http://www.bayesian-inference.com/index` (99).

Strathmann, H., D. Sejdinovic, S. Livingstone, Z. Szabo, and A. Gretton (2015). "Gradient-free Hamiltonian Monte Carlo with efficient kernel exponential families". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 955–963 (30).

Sugita, Y. and Y. Okamoto (1999). "Replica-exchange molecular dynamics method for protein folding". In: *Chemical Physics Letters* 314.1, pp. 141–151 (92, 93).

Sweet, C. R., S. S. Hampton, R. D. Skeel, and J. A. Izaguirre (2009). "A separable shadow Hamiltonian hybrid Monte Carlo method". In: *The Journal of Chemical Physics* 131, 174106 (17). DOI: `doi:10.1063/1.3253687` (34, 70, 110).

Swendsen, R. H. and J. S. Wang (1986). "Replica Monte Carlo simulation of spin-glasses". In: *Physical Review Letters* 57, pp. 2607–2609 (92).

Takaishi, T. (2013). "Empirical Analysis of Stochastic Volatility Model by Hybrid Monte Carlo Algorithm". In: *Journal of Physics: Conference Series*. Vol. 423. 1. IOP Publishing, pp. 12021–12030 (123).

— (2014). “Bayesian estimation of realized stochastic volatility model by Hybrid Monte Carlo algorithm”. In: *Journal of Physics: Conference Series*. Vol. 490. 1. IOP Publishing, pp. 12092–12095 (26).

Takaishi, T. and P. de Forcrand (2006). “Testing and tuning symplectic integrators for the hybrid Monte Carlo algorithm in lattice QCD”. In: *Physical Review E* 73.036706 (26).

Terterov, I., B. Escribano, M. Dubina, and E. Akhmatskaya (2013). “Implementation of Meso-GSHMC Algorithm in Gromacs Molecular Simulation Package”. In: *Proceedings of the International Workshop on Computational and Theoretical Modeling of Biomolecular Interactions*. Vol. 75. Moscow-Izhevsk: Institute of Computer Science, Russia (35).

Verlet, L. (1967). “Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard–Jones molecules”. In: *Physical Review* 159, pp. 98–103 (17, 23).

Wagoner, J. A. and V. S. Pande (2012). “Reducing the effect of Metropolization on mixing times in molecular dynamics simulations”. In: *Journal of Chemical Physics* 137.21, p. 214105 (vii, xiv, 75, 76, 81).

Wang, Z. and N. de Freitas (2011). “Predictive adaptation of hybrid Monte Carlo with Bayesian parametric bandits”. In: *NIPS Deep Learning and Unsupervised Feature Learning Workshop* (30).

Wang, Z., S. Mohamed, and N. de Freitas (2013). “Adaptive Hamiltonian and Riemann manifold Monte Carlo samplers”. In: *Proceedings of the 30th International Conference on Machine Learning*, pp. 1462–1470 (30, 87–91, 123).

Wee, C. L., M. S. Sansom, S. Reich, and E. Akhmatskaya (2008). “Improved sampling for simulations of interfacial membrane proteins: Application of generalized shadow hybrid Monte Carlo to a peptide toxin/bilayer system”. In: *The Journal of Physical Chemistry B* 112.18, pp. 5710–5717 (35, 40).

Zhang, C., B. Shahbaba, and H. Zhao (2015a). “Hamiltonian Monte Carlo Acceleration Using Neural Network Surrogate functions”. In: *arXiv:1506.05555v2* (30).

— (2015b). “Precomputing Strategy for Hamiltonian Monte Carlo Method Based on Regularity in Parameter Space”. In: *arXiv:1504.01418v1* (30).

Zhang, Y. and C. Sutton (2014). “Semi-separable Hamiltonian Monte Carlo for inference in Bayesian hierarchical models”. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 10–18 (30, 123).

Zhang, Y., Z. Ghahramani, A. J. Storkey, and C. A. Sutton (2012). “Continuous relaxations for discrete Hamiltonian Monte Carlo”. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3194–3202 (30).

Zhang, Y., X. Wang, C. Chen, K. Fan, and L. Carin (2016). “Monomial Gamma Monte Carlo Sampling”. In: *arXiv preprint arXiv:1602.07800* (31).

# *Acknowledgements*

I would like to express my gratitude to those who have helped me during my graduate years in making this work possible and these years memorable. My special thanks go to:

My supervisor Prof. Elena Akhmatskaya, for her endless support, guidance and encouragement. I particularly appreciate her availability for joint work, constructive feedback and overall approach, as well as her invaluable scientific expertise and wisdom. Above all, I thank her for the friendship and countless thoughtful conversations, which have made me grow both professionally and personally.

My supervisor Prof. Enrico Scalas, for introducing me to interdisciplinary research and demonstrating its importance and prolificacy, for the time we spent working together giving me the opportunities to learn from him, and for his patience and inspiring conversations.

Dr. Jonatha Anselmi, Prof. Ubaldo Garibaldi, Prof. Fabio Rapallo and Mario Fernández Pendás, for having the opportunity to collaborate with them.

Prof. Jesús María Sanz-Serna, Prof. Mark Girolami, Dr. Michael Betancourt, Dr. Ben Calderhead, for insightful and interesting discussions.

Dr. Ángel Rodríguez-Rozas, for his help with the code development.

Dr. Ben Calderehad, Mario Fernández Pendás, Prof. Jose Antonio Lozano, and Dr. Andrijana Radivojević, for their valuable comments on the Thesis manuscript.

Prof. Jesús María Sanz-Serna, Prof. Mari Paz Calvo and Prof. Jose Antonio Lozano, for serving as members of my Thesis Committee.

MSLMS group members, Mario, Simone, Ariel and Bruno, for always being available for friendly chats, either technical or non-technical.

Dedicated staff of BCAM, especially Miguel Benítez, for being available and resolving all the administrative issues appearing in my way.

Eneko&Eneko, for providing the IT support whenever needed.

All present and former members of BCAM, for many breakfasts, lunches, coffee breaks, interesting discussions, and for creating such a pleasant team atmosphere.

Felipe, Martin, Cristi, Fabio, Luigi, Ivan, Mario, Simone, Julia, for being great office-mates of the *pejor!* Y19, and for the memorable moments we have shared outside BCAM.

María, Felipe, Maialen, Vincent, Javi, Tamás, Alejandro, Imanol, Julia, and of course, Osolomo, for being honorable BCAMtatoes and bearing all adventures the title assumed.

Amaia, Aitor, Carlos, Asier, Peio, for being amazing flatmates and introducing me to the Basque culture.

Mario, my scientific brother, for being my Spanish teacher, for all the help with translations, and in particular, for his generous friendship.

Goran and Arrate, for being my euskal family, for the great times we have shared (and we will share) and for being by my side in all ups and downs along this journey. Hvala, prijatelju! Eskerrik asko, lagun maitea!

Ana, Jovana, Sonja, Tamara, Saša, Miroslav, Milan, for their most sincere friendship and innumerable online conversations. Bogatstvo je imati prijatelje kao što ste vi! Hvala na svoj energiji koju mi prenosite!

Ángel, for his love and devotion, and for withstanding heroically the final phase of my PhD work. ¡Om, vtq!

My sister Andrijana, my mom Nada and my dad Cojo, for their unconditional love and endless support in everything that I do. Ova teza je posvećena vama, mojim najdražima.

Tijana Radivojević,
Bilbao, September 2016