

Symmetrically processed splitting integrators for enhanced Hamiltonian Monte Carlo sampling

S. Blanes,^{*} M.P. Calvo,[†] F. Casas,[‡] and J.M. Sanz-Serna[§]

November 10, 2020

Abstract

We construct integrators to be used in Hamiltonian (or Hybrid) Monte Carlo sampling. The new integrators are easily implementable and, for a given computational budget, may deliver five times as many accepted proposals as standard leapfrog/Verlet without impairing in any way the quality of the samples. They are based on a suitable modification of the processing technique first introduced by J.C. Butcher. The idea of modified processing may also be useful for other purposes, like the construction of high-order splitting integrators with positive coefficients.

AMS numbers: 65L05, 65C05, 37J05

Keywords: Hamiltonian Monte Carlo method, splitting integrators, processing

1 Introduction

In this paper we show how to construct symmetrically processed splitting algorithms for efficient Hamiltonian (or Hybrid) Monte Carlo (HMC) sampling.

^{*}Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, 46022-Valencia, Spain. E-mail: serblaza@imm.upv.es

[†]Departamento de Matemática Aplicada e IMUVA, Facultad de Ciencias, Universidad de Valladolid, Spain. E-mail: mariapaz.calvo@uva.es

[‡]Departament de Matemàtiques and IMAC, Universitat Jaume I, E-12071 Castellón, Spain. E-mail: casas@uji.es

[§]Departamento de Matemáticas, Universidad Carlos III de Madrid, Avenida de la Universidad 30, E-28911 Leganés (Madrid), Spain. E-mail: jmsanzserna@gmail.com

HMC is a widely used sampling technique; introduced in the physics literature [17], it has become very popular in statistics [28] and may provide large improvements over alternative approaches [14]. All the many variants of HMC share the need for integrating numerically a system of Hamiltonian differential equations at each step of the Markov chain [34] and in fact the gradient evaluations required by the numerical integrator dominate the computational cost of obtaining the samples. It is therefore useful to identify existing integrators or to construct new ones that are efficient in HMC sampling. Even though the leapfrog/Störmer/Verlet method is the integrator usually chosen, it is possible to cut down substantially the computational cost of the integrations *without impairing in any way* the quality of the sampling by using (multistage) *splitting integrators* which may be implemented as easily as leapfrog and are *time reversible* and *volume preserving*, two essential requirements for HMC use [5].

There are several families of possible splitting schemes and each family includes free parameters. The paper [9] suggested a methodology for choosing the splitting parameters so as to optimize the efficiency in the HMC setting. As distinct from other works (see [26, 30, 31, 20, 22] among many others), where the choice of parameters is determined by the behaviour of the integrator as the step size h approaches 0, in [9] h is not assumed to be small; it rather ranges in a suitable interval $(0, \bar{h})$. This corresponds to the fact that successful HMC simulations operate with rather large values of h [28, 5]. Because free parameters are not used to boost the accuracy in the limit $h \rightarrow 0$, all the integrators constructed in [9] are second order. This implies that they yield *average* energy errors of order four [4, 5]. Extensive numerical experiments [9, 3, 18, 13, 25, 29, 1, 12, 20] in a variety of applications that range from Bayesian statistics to the Auxiliary Field Quantum Monte Carlo method and theoretical results presented in [12] endorse the soundness of the approach in [9]. In particular [12] shows experimentally that for splitting formulas that use three gradient evaluations per step, the parameter choice in [9] leads to better HMC sampling than any other parameter choices. The references [2, 32] extended the technique in [9] to modified HMC algorithms that combine the basic idea of HMC with importance sampling.

The idea of *processing* numerical integrators [7, Section 3.5] is due to J.C. Butcher [10]. Given a one-step integrator, sometimes called the *kernel*, a processed integration requires (i) preprocessing the initial condition, (ii) integrating with the kernel and (iii) postprocessing the solution. Pre and postprocessing should have negligible complexity, so that the cost of an integration with processing is essentially the cost of integrating with the

kernel. The interest of the idea lies in cases where the accuracy of the processed integration is higher than the accuracy the kernel alone would provide. For instance, the processed algorithm may converge with an order ν higher than the order μ of the kernel; when this happens the kernel is said to possess *effective order* ν . Processing did not become popular when it was first suggested, due to the difficulties of its combination with variable time steps. It reappeared [23, 24, 8] in the geometric integration scenario, where the emphasis is in constant time steps [11]. Since processing may considerably increase the efficiency of an integrator, it is natural to study whether the splitting kernels for HMC applications successfully used in [9, 3, 18, 13, 25, 29, 1, 12, 20] may be processed. Unfortunately the standard approach to processing, where the postprocessing map just inverts the action of the preprocessor, will not work, as it leads to integrations that are *not* time reversible.

In this paper:

1. We introduce (Section 2) *symmetric (modified) processing*, a modification of standard processing under which time reversible kernels provide time reversible integrations.
2. We provide (Section 3) a methodology to determine the parameters in the kernel and the pre and postprocessor to optimize the performance of the integrator in HMC sampling. This methodology extends the material in [9].
3. We construct (Section 4) specific symmetric processed algorithms to be applied within HMC simulations.
4. We show (Section 5) by means of numerical experiments that the sampling efficiency of the new symmetric processed integrators may improve on the standard velocity Verlet integrator by a factor of five or more.

HMC sampling is not the only application where symmetric modified processing may be useful and the final Section 6 briefly discusses another possible application area: the construction of higher-order splitting methods with positive coefficients.

2 Symmetric (modified) processing

2.1 Definition

Given a system of differential equations $(d/dt)x = f(x)$ in \mathbb{R}^D , each one-step integrator is specified by a map $\psi_h : \mathbb{R}^D \rightarrow \mathbb{R}^D$ that advances the numerical solution over a time-interval of length h . For instance $\psi_h(x) = x + hf(x)$ corresponds to Euler's rule. If ψ_h is an integrator (sometimes called the kernel) and $\pi_h : \mathbb{R}^D \rightarrow \mathbb{R}^D$ is a map, one may consider the corresponding so-called *processed integrator* defined by the map

$$\widehat{\psi}_h = \pi_h^{-1} \circ \psi_h \circ \pi_h$$

(the superscript $^{-1}$ denotes inverse map and \circ means composition). N consecutive steps of the processed integrator correspond to the transformation

$$\widehat{\psi}_h^N = \underbrace{\widehat{\psi}_h \circ \cdots \circ \widehat{\psi}_h}_{N \text{ times}} = \underbrace{(\pi_h^{-1} \circ \psi_h \circ \pi_h) \circ \cdots \circ (\pi_h^{-1} \circ \psi_h \circ \pi_h)}_{N \text{ times}},$$

that is,

$$\widehat{\psi}_h^N = \pi_h^{-1} \circ \psi_h^N \circ \pi_h. \quad (1)$$

In other words, to perform N steps of the processed method, one successively (i) applies once the map π_h (preprocessing), (ii) takes N steps of the kernel ψ_h and (iii) applies once the map π_h^{-1} (postprocessing). Since π_h and its inverse are applied only once per integration leg, the computational complexity of $\widehat{\psi}_h$ is not very different from that of ψ_h . Processing is advantageous in situations, among others, where $\widehat{\psi}_h$ is more accurate than the unprocessed ψ_h (for instance $\widehat{\psi}_h$ may have higher order of convergence or smaller error constants than ψ_h).

Recall that the true solution flow ϕ_t of the system being integrated is *time reversible (or symmetric)*, in the sense that ϕ_t^{-1} (which maps the final state into the initial condition) coincides with ϕ_{-t} (which moves the initial condition backwards in time). Correspondingly, an integrator ψ_h is said to be time reversible (or symmetric) [35, Section 3.6] if $(\psi_h^N)^{-1} = \psi_{-h}^N$. From (1) it is easily concluded that, even if the kernel ψ_h is time reversible, the processed $\widehat{\psi}_h$ may not be expected to be so. The symmetric (modified) processing approach suggested in the present paper is a modification of the idea of processing that makes it possible to obtain time reversible integrators.

To perform an integration leg spanning a time interval of length Nh with a symmetric modified processed integrator one applies the map

$$\widetilde{\psi}_{N,h} = \pi_h^* \circ \psi_h^N \circ \pi_h, \quad (2)$$

where π_h^* denotes the *adjoint* of π_h , i.e. the map such that $\pi_{-h}^* = \pi_h^{-1}$ (see e.g. [35, Section 3.6] or [7, Section 1.2]). This differs from standard processing (see (1)) in that the adjoint rather than the inverse is used as a postprocessor. Since

$$(\tilde{\psi}_{N,h})^{-1} = \pi_h^{-1} \circ (\psi_h^N)^{-1} \circ (\pi_h^*)^{-1}$$

and

$$\tilde{\psi}_{N,-h} = \pi_{-h}^* \circ \psi_{-h}^N \circ \pi_{-h} = \pi_h^{-1} \circ \psi_{-h}^N \circ (\pi_h^*)^{-1},$$

the symmetric modified processed integrator (2) will be time reversible if ψ_h is time reversible.

For processing integrators, the map (1) that advances the solution over a time interval $[0, Nh]$ is the N -th power of the one-step map $\hat{\psi}_h$. For symmetric processing the map $\tilde{\psi}_{N,h}$ in (2) is not obtained by means of a similar N -fold composition of a one-step map.

2.2 The case of splitting integrators

Although the idea of symmetric processing is completely general, our attention is restricted to the case where in (2) ψ_h and π_h are constructed via splitting (see e.g. the monographs [35, 21, 7] and the survey [27]). If the system being integrated may be written in the split form

$$\frac{d}{dt}x = f(x) = f^A(x) + f^B(x),$$

and ϕ_t^A and ϕ_t^B represent the exact flows of the split systems, we deal with splitting kernels

$$\psi_h = \phi_{b_1 h}^B \circ \phi_{a_1 h}^A \circ \cdots \circ \phi_{a_{r-1} h}^A \circ \phi_{b_r h}^B \circ \phi_{a_r h}^A \circ \phi_{b_r h}^B \circ \phi_{a_{r-1} h}^A \circ \cdots \circ \phi_{a_1 h}^A \circ \phi_{b_1 h}^B. \quad (3)$$

(Since it is possible to set $a_r = 0$, this format includes integrators where the central flow is ϕ^B rather than ϕ^A . Similarly one may set $b_1 = 0$ to have integrators where the extreme flows are ϕ^A .) The palindromic structure of (3) ensures time reversibility. This integrator is consistent (in fact of order ≥ 2 due to symmetry) if

$$2a_1 + \cdots + 2a_{r-1} + a_r = 1, \quad 2b_1 + \cdots + 2b_{r-1} + 2b_r = 1, \quad (4)$$

and in what follows we always assume that these conditions hold.

Similarly, we choose π_h to be a composition of $2s$ flows of the form

$$\pi_h = \phi_{c_s h}^A \circ \phi_{d_s h}^B \circ \cdots \circ \phi_{c_1 h}^A \circ \phi_{d_1 h}^B, \quad (5)$$

which leads to

$$\pi_h^* = \phi_{d_1 h}^B \circ \phi_{c_1 h}^A \circ \cdots \circ \phi_{d_s h}^B \circ \phi_{c_s h}^A. \quad (6)$$

We assume that

$$c_1 + \cdots + c_s = 0, \quad d_1 + \cdots + d_s = 0, \quad (7)$$

which imply that π_h and π_h^* differ from the identity map by $\mathcal{O}(h^2)$ terms. In this way, it is clear that (2) is a time reversible integrator with even order of accuracy ≥ 2 .

We use the abbreviations

$$(b_1, a_1, \dots, a_{r-1}, b_r, a_r, b_r, a_{r-1}, \dots, a_1, b_1),$$

and

$$(c_s, d_s, \dots, c_1, d_1), \quad (d_1, c_1, \dots, d_s, c_s)$$

to refer to (3), (5), and (6) respectively. In this way, (2) is denoted as

$$(d_1, c_1, \dots, d_s, c_s)(b_1, a_1, \dots, b_r, a_r, b_r, \dots, a_1, b_1)^N(c_s, d_s, \dots, c_1, d_1); \quad (8)$$

the palindromic structure is apparent.

2.3 The Hamiltonian case

We have in mind the integration of Hamiltonian systems of the form

$$\frac{d}{dt}q = M^{-1}p, \quad \frac{d}{dt}p = -\nabla V(q), \quad (9)$$

where M is a symmetric, positive definite $d \times d$ mass matrix and V denotes the potential. Under the familiar q/p or potential/kinetic splitting, the split flow ϕ_t^A is the solution flow of the system

$$\frac{d}{dt}q = M^{-1}p, \quad \frac{d}{dt}p = 0$$

and ϕ_t^B corresponds to

$$\frac{d}{dt}q = 0, \quad \frac{d}{dt}p = -\nabla V(q).$$

In molecular dynamics the transformations ϕ_t^A and ϕ_t^B are known as drifts and kicks respectively. Thus an integration leg with the symmetric processed integrator (2) is a succession kick, drift, kick, \dots , kick with a palindromic pattern and therefore time reversible. In addition the map (2) is volume preserving as a composition of kicks and drifts. Time reversibility and volume preservation ensure that (2) may be used in HMC applications with the simple standard recipe for the accept/reject probability applied with leapfrog [5].

3 Choosing the parameters

The paper [9] suggested a technique to identify “good” parameter values in families of numerical integrators for the Hamiltonian system (9) in the HMC context. Extensive numerical experiments in statistical and molecular dynamics problems reported in [13, 18, 12] show clearly the soundness of the approach in [9] and it is this approach that we follow here. However the material in that paper cannot be directly applied to symmetric modified processed algorithms and in this section we present the necessary modifications.

The technique is based on discriminating between integrators by applying them to the one-degree-of-freedom model Hamiltonian $p^2/2 + q^2/2$ for which the equations of motion correspond to the standard harmonic oscillator and the target probability density function is $\propto \exp(-(p^2 + q^2)/2)$ so that q and p are independent with a standard normal distribution. (The results obtained for the model problem apply to more general situations [5, 12].) For the harmonic oscillator, N steps of a given palindromic splitting integrator (3) define a linear transformation $(q_0, p_0) \mapsto (q_N, p_N)$ of the form

$$\begin{bmatrix} q_N \\ p_N \end{bmatrix} = \begin{bmatrix} C & \chi S \\ -\chi^{-1} S & C \end{bmatrix} \begin{bmatrix} q_0 \\ p_0 \end{bmatrix}. \quad (10)$$

Here (q_0, p_0) is the initial condition, (q_N, p_N) the numerical solution at the end of the integration leg, C and S are abbreviations for $\cos(N\theta_h)$ and $\sin(N\theta_h)$ respectively and $\chi = \chi_h$, $\theta = \theta_h$ are quantities that change with the step length h . A priori, θ_h may be complex-valued, but if h is of sufficiently small magnitude, then the 2×2 matrix above is power bounded (stability) and, as shown in [9], this corresponds to θ_h being real, something we assume hereafter. (In fact the format (10) is not specific to splitting integrators; it is shared by any reasonable time reversible, volume preserving unprocessed integrator for (9), see [9, 5].) If N varies, the points (q_N, p_N) defined by (10) move on an ellipse of the (q, p) -plane, whose eccentricity is governed by χ_h . For $\chi_h = 1$, the transformation in (10) is a rotation, the ellipse becomes a circle and the energy error in the numerical simulation $(1/2)(q_N^2 + p_N^2) - (1/2)(q_0^2 + p_0^2)$ vanishes: all proposals are then accepted.

Similarly to (10), for the standard harmonic oscillator, the preprocessor π_h and the postprocessor π_h^* in (5) and (6) are respectively associated with 2×2 matrices of the form

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}, \quad \begin{bmatrix} \delta & \beta \\ \gamma & \alpha \end{bmatrix}, \quad (11)$$

where $\alpha = \alpha_h$, $\beta = \beta_h$, $\gamma = \gamma_h$, $\delta = \delta_h$ are polynomials in h , with $\alpha_h \delta_h - \beta_h \gamma_h = 1$ by conservation of volume. In addition α_h and δ_h are even in h and β_h and γ_h are odd so that

$$\begin{bmatrix} \delta_{-h} & \beta_{-h} \\ \gamma_{-h} & \alpha_{-h} \end{bmatrix} = \begin{bmatrix} \delta_h & -\beta_h \\ -\gamma_h & \alpha_h \end{bmatrix} = \begin{bmatrix} \alpha_h & \beta_h \\ \gamma_h & \delta_h \end{bmatrix}^{-1},$$

as needed for a map and its adjoint. (It is perhaps useful to observe that in standard processing (1), the matrix of the postprocessor would be

$$\begin{bmatrix} \alpha_h & \beta_h \\ \gamma_h & \delta_h \end{bmatrix}^{-1} = \begin{bmatrix} \delta_h & -\beta_h \\ -\gamma_h & \alpha_h \end{bmatrix},$$

which differs from the postprocessing matrix for symmetric processing given in (11) in the sign of the non-diagonal entries; these entries are $\mathcal{O}(h)$ as $h \rightarrow 0$.)

Combining (10) with (11), one integration leg for (2) is then given by

$$\begin{bmatrix} q_N \\ p_N \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{A} \end{bmatrix} \begin{bmatrix} q_0 \\ p_0 \end{bmatrix}, \quad (12)$$

with

$$\begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{A} \end{bmatrix} = \begin{bmatrix} \delta & \beta \\ \gamma & \alpha \end{bmatrix} \begin{bmatrix} C & \chi S \\ -\chi^{-1} S & C \end{bmatrix} \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}.$$

We multiply out the matrices to find:

$$\begin{aligned} \mathcal{A} &= C(\alpha\delta + \beta\gamma) + S(\gamma\delta\chi - \alpha\beta\chi^{-1}), \\ \mathcal{B} &= C(2\beta\delta) + S(\delta^2\chi - \beta^2\chi^{-1}), \\ \mathcal{C} &= C(2\alpha\gamma) + S(\gamma^2\chi - \alpha^2\chi^{-1}). \end{aligned}$$

The ‘‘ideal’’ preprocessor has $\alpha_h = \chi_h^{1/2}$, $\delta_h = 1/\alpha_h$, $\beta_h = \gamma_h = 0$ leading to $\mathcal{A} = C$, $\mathcal{B} = S$, $\mathcal{C} = -S$. With this ideal processing, (12) is a rotation, the symmetrically processed integrator conserves energy exactly and there are no rejections in HMC sampling. Unfortunately such ideal preprocessor cannot be realized by means of a splitting formula of the form (5) and our aim now is to identify processors so that (12) is, in a suitable sense, as close to a rotation as possible.

At this stage, we assume that q_0 and p_0 are independent random variables with standard normal distribution (i.e. that the Markov chain is at stationarity) and consider the change in energy

$$\Delta(q_0, p_0) = \frac{1}{2}(q_N^2 + p_N^2) - \frac{1}{2}(q_0^2 + p_0^2)$$

over one integration leg. Here (q_N, p_N) and therefore $\Delta(q_0, p_0)$ are deterministic functions of the random initial condition and therefore random variables themselves. By conservation of energy $\Delta(q_0, p_0)$ would vanish if the integrator were exact; in HMC simulations small values of Δ correspond to high acceptance probability. We have the following result:

Proposition 1 *With the preceding notation, $\mathbb{E}(\Delta) = (1/2)(\mathcal{B} + \mathcal{C})^2$.*

Proof. An elementary computation yields:

$$2\Delta = (\mathcal{A}^2 + \mathcal{C}^2 - 1)q_0^2 + 2(\mathcal{A}\mathcal{B} + \mathcal{C}\mathcal{A})q_0p_0 + (\mathcal{B}^2 + \mathcal{A}^2 - 1)p_0^2$$

and therefore taking expectations

$$2\mathbb{E}(\Delta) = 2\mathcal{A}^2 + \mathcal{B}^2 + \mathcal{C}^2 - 2.$$

The result follows because $\mathcal{A}^2 - \mathcal{B}\mathcal{C} = 1$ by conservation of volume. ■

Note that $\mathbb{E}(\Delta) \geq 0$, a well-known fact in HMC simulations [4]. The proposition leads to our next result that extends to the situation at hand the bound in [9, Proposition 4.3] valid for the unprocessed case.

Proposition 2 *The expectation of Δ may be bounded above as follows:*

$$\mathbb{E}(\Delta) \leq \rho_h,$$

with

$$\rho_h = 2(\alpha\gamma + \beta\delta)^2 + \frac{1}{2}[(\delta^2 + \gamma^2)\chi - (\alpha^2 + \beta^2)\chi^{-1}]^2.$$

Proof. From the expressions for \mathcal{B} and \mathcal{C}

$$(\mathcal{B} + \mathcal{C})^2 = \left(2C(\alpha\gamma + \beta\delta) + S[(\delta^2 + \gamma^2)\chi - (\alpha^2 + \beta^2)\chi^{-1}]\right)^2.$$

In the right hand-side we have the standard dot product of the vector $\mathbf{v}_1 \in \mathbb{R}^2$ with components $2(\alpha\gamma + \beta\delta)$ and $(\delta^2 + \gamma^2)\chi - (\alpha^2 + \beta^2)\chi^{-1}$ and the *unit* vector $\mathbf{v}_2 \in \mathbb{R}^2$ with components $C = \cos(N\theta_h)$ and $S = \sin(N\theta_h)$. The magnitude of the inner product can then be bounded above by the length of \mathbf{v}_1 and the result follows easily. ■

For a fixed symmetric processed integrator, the upper bound ρ_h is *independent of the number of steps N* ; it only depends on h and does so through the quantity χ associated with the kernel and with the quantities $\alpha, \beta, \gamma, \delta$ associated with the pre and postprocessor. Of course, in the case where a *family* of integrators is considered, for each h the quantity ρ_h changes with

the specific choice of algorithm within the family. According to the strategy in [9], one should pick up a value \bar{h} that represents the maximum value of h to be used in the simulations of the model problem and then prefer the member of the family that *minimizes* the expected energy-error metric

$$\|\rho\|_{\bar{h}} = \max_{0 < h < \bar{h}} \rho_h.$$

It is proved in [12] that this is equivalent to *maximizing* the expected acceptance rate. In [9] it is recommended to set \bar{h} equal to the number of evaluations of ∇V necessary to perform a single time step. Note that this implies that making the integrator more computationally intensive by increasing the number of gradient evaluations per step increases the value of $\|\rho\|_{\bar{h}}$. (It is also possible [18] to adapt the value of \bar{h} to the specific Hamiltonian under consideration, but that line of thought will not be pursued here.)

4 Specific integrators

Several specific (unprocessed) integrators were constructed in [9] by means of the methodology introduced there. In the next section we will report numerical results for a method (to be referred to as BiCaSa) of the form (3) with $r = 2$. A single time step of BiCaSa uses four evaluations of ∇V , but, since the evaluation of ∇V at the first kick of the next time step coincides with the evaluation at the last kick of the current step, the computational cost is essentially three gradient evaluations per time step. Integrators that use two or four evaluations per time step were also constructed in [9], but that reference found three evaluations per time step to be preferable. With two evaluations, the resulting larger value $\|\rho\|_{\bar{h}}$ offsets the benefit of the smaller computational cost per time step. Four evaluations lead to a marginal improvement of $\|\rho\|_{\bar{h}}$, which does not really compensate for the extra complication. In [12] BiCaSa was found to clearly outperform in HMC applications other integrators of the family (3) with $r = 2$. For these reasons we will use BiCaSa as a measuring rod to assess the efficiency of the symmetric processed integrators to be constructed.

We focus our attention on symmetric processed integrators of the form (8) with $r = 2$. After imposing the consistency requirement (4), we may regard $a = a_1$ and $b = b_2$ as free parameters for the kernel. It is shown in [13] that unless $a = b/(6b - 1)$ the stability interval of the kernel is very short, which makes the integrator uncompetitive. Therefore we impose this relation and deal with a one-parameter family of kernels. To keep pre and

\bar{h}	b	c	d	$\ \rho\ _{\bar{h}}$	h_s
3	0.381120	-----	-----	7×10^{-5}	4.602
3	0.348674	-0.075640	0.069720	6×10^{-8}	4.985
3.5	0.346660	-0.079510	0.070171	5×10^{-7}	5.010
4	0.343684	-0.084690	0.071880	5×10^{-6}	5.048
4.5	0.340200	-0.093500	0.072800	5×10^{-5}	5.095

Table 1: The last four rows give information on symmetric processed integrators for HMC applications. The methods have been found by minimizing the expected energy-error metric $\|\rho\|_{\bar{h}}$ for the values of \bar{h} displayed in the leftmost column. The first row corresponds to the BlCaSa integrator. The table provides the parameter values to run the integrators and, in the last column, the length of the linear stability interval of the kernel.

postprocessing as simple as possible, we set $s = 2$, the lowest value for which the consistency relations (7) have a nontrivial solution. We thus work with a three parameter family of integrators of the form

$$(d, c, -d, -c) \left(\frac{1}{2} - b, a, b, 1 - 2a, b, a, \frac{1}{2} - b \right)^N (-c, -d, c, d),$$

where $a = b/(6b - 1)$. An integration leg requires a total of $3N + 5$ gradient evaluations (two of them within the pre or postprocessing). Of course, if N is large this is approximately $3N$.

Initially we determined the values of the parameters b, c, d by minimizing $\|\rho\|_{\bar{h}}$ with $\bar{h} = 3$, the value suggested in [9]. The values we obtained may be seen in Table 4, along with the length h_s of the stability interval of the kernel, i.e. the supremum of the step sizes h for which the matrix in (10) may be bounded independently of N thus guaranteeing that errors do not grow exponentially as N increases. (Note that the stability of the kernel determines the stability of the overall integrator, since the pre and postprocessor are applied only once.) For comparison we have also included in the first row of the Table the data corresponding to BlCaSa: symmetric processing results in a reduction of $\|\rho\|_3$ *by three orders of magnitude*. This reduction is achieved at the price of only four additional gradient evaluations per integration leg.

The extremely small value of $\|\rho\|_3$ that may be achieved in this way, prompted us to explore larger values of \bar{h} , so as to obtain integrators meant to operate with larger step sizes. Our results with $\bar{h} = 3.5, 4$ or 4.5 are also reported in the table. The integrator specified in the last row, run with

$0 < h < 4.5$, is (on the Gaussian model) as accurate as BICaSa when run with $0 < h < 3$.

5 Numerical results

As in [9] our first test problem is the model multivariate Gaussian target with density (q_j is the j -component of q)

$$\propto \exp\left(-\frac{1}{2}\sum_{j=1}^d j^2 q_j^2\right).$$

We have used the dimensions $d = 256$, $d = 1024$ and $d = 4096$: our interest is in problems of large dimensionality, those where efficiency is more important (for targets in very low dimension leapfrog/Verlet performs very satisfactorily as a consequence of its optimal linear stability properties [5]). As in [12], integrations were performed, for different stable choices of h , in the interval $0 \leq t \leq 5$ and we generated Markov chains of 5,000 elements initialized from the target distribution. As discussed in detail in [12], the efficiency of the algorithm and the *quality of the samples* is entirely determined by the acceptance rate and therefore we will focus on this metric. The conclusions to be drawn as to the merit of the different integrators based on the behaviour of the acceptance rate are the *same* that may be obtained by considering other metrics such as mean square displacement, effective sample sizes of the different components of q , etc.

Our results are summarized in Figure 1, where we compare the symmetrically processed integrators with $b = 0.348674$ and $b = 0.340200$ (see Table 4) against the integrator BICaSa and standard velocity leapfrog/Verlet. The intermediate values $b = 0.346660$ and $b = 0.343684$ in the table were also run; the results interpolate between those of $b = 0.348674$ and those of $b = 0.340200$ and are not reported so as to not blur the plots. For the symmetrically processed methods, the reported gradient evaluation count includes the evaluations required by the pre and postprocessing.

The *left* subplots give, for the four integrators, acceptance rate as a function of the number of gradient evaluations per integration leg. Of course, for each integrator, more gradient evaluations per leg (corresponding to smaller values of h) provide higher acceptance rates. The advantage of the three multistage integrators over Verlet is clearly borne out, and this advantage becomes more pronounced as the dimensionality increases (i.e. as it becomes more important to have efficient algorithms). For the integrator with

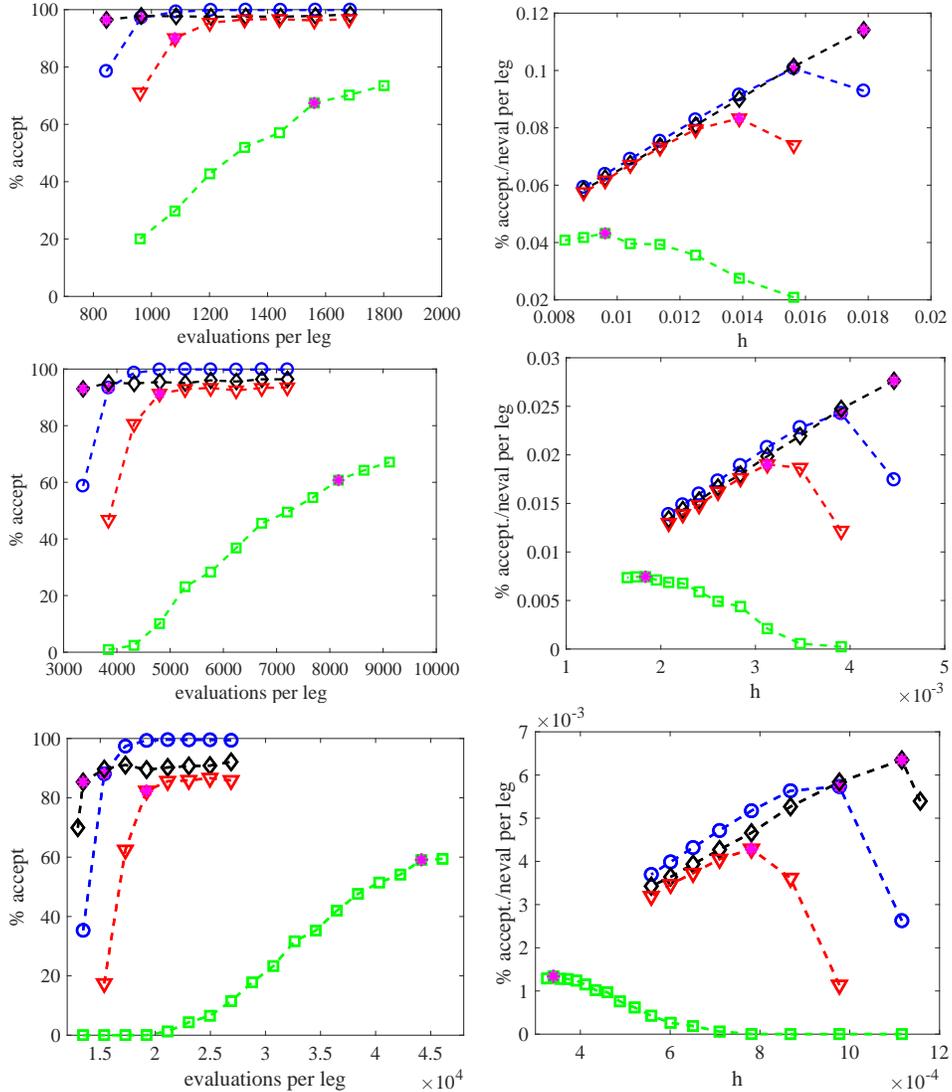


Figure 1: Gaussian model. Comparison of the symmetrically processed methods with $b = 0.348674$ (blue circles) and $b = 0.340200$ (black diamonds) with the integrators BICaSa (red triangles) and leapfrog (green squares). The top, middle and bottom panels have $d = 256$, $d = 1024$ and $d = 4096$ respectively. On the left, acceptance percentage as a function of the number of evaluations per integration leg. On the right, acceptance percentage divided by number of gradient evaluations for different values of the step size h . For each integrator and d a star symbol on the marker identifies the most efficient run.

$b = 0.348674$, the runs with more gradient evaluations deliver acceptance rates of virtually 100%, which is in agreement with the low energy errors that correspond to the extremely low value of $\|\rho\|_{\bar{h}}$ reported in the table. Also note that $b = 0.340200$ operates very well for runs with fewer evaluations (larger h) and this matches the fact that this method was derived using a larger value of \bar{h} .

Even though very low acceptance rates are unwelcome, very high acceptance rates are also undesirable. In fact, it is well understood [4, 5, 12] that, for a given integrator, a very high acceptance rate signals that the value of h being employed is too small: one would do better by using the available computational budget to obtain longer Markov chains by getting proposals at a lower computational cost per leg, even if that implies rejecting more proposals. For this reason, it is not easy to assess the efficiency of the different integrators by examining the left subplots we have been discussing. This efficiency is best assessed from the *right* subplots that give, for different values of h , the acceptance rate *per unit computational cost*, i.e. the result of dividing the empirical acceptance rate achieved in a simulation by the number of gradient evaluations required. Larger values of this metric correspond to more efficient sampling. The figure shows that according to our discussion, for a given integrator, the best efficiency (i.e. the highest marker) is not obtained when h is too small or too large. In the figure the most efficient run for each of the four integrators has been indicated by a purple star on the corresponding marker. The right panels make it clear that BICaSa is far more efficient than Verlet and the gap in efficiency increases with the dimensionality. For $d = 4096$ the optimal value of h for Verlet is $\approx 2 \times 10^{-4}$ and then the acceptance rate divided by the number of gradient evaluations is ≈ 1 ; for BICaSa the optimal value of h is substantially larger $\approx 8 \times 10^{-4}$ and yields an acceptance rate per unit computational cost ≈ 4 , approximately a fourfold improvement on Verlet. In turn the performance of the symmetrically processed integrators clearly improves on BICaSa. For $d = 4096$, the integrator with $b = 0.340200$ is roughly five times more efficient than Verlet and approximately 50% more efficient than BICaSa.

We now present results (Figure 2) for a well-known Log-Gaussian Cox problem in Bayesian inference, often used as a test [16, 19, 12]. The dimension is $d = 4096$. The details of the sampling (length of the chain, burn-in, initialization, step sizes, time length of the integration legs, etc.) are exactly as in [12] and will not be reproduced here. The general pattern of the results is very similar to that in Figure 1. Again BICaSa is roughly four times more efficient than Verlet. Now $b = 0.348674$ and $b = 0.340200$

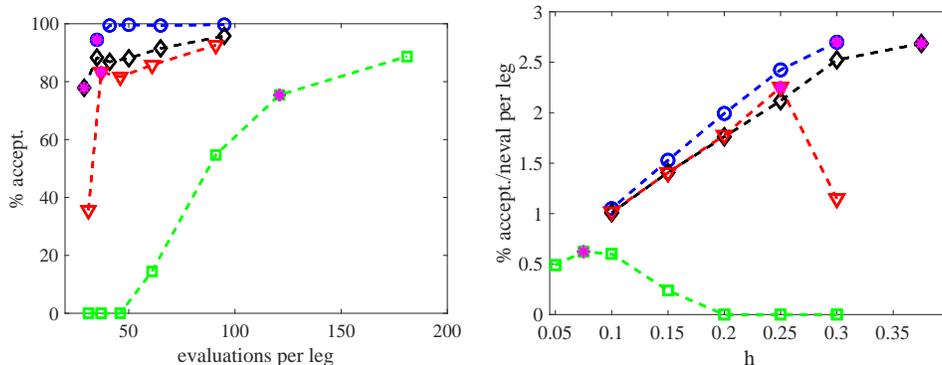


Figure 2: Log-Gaussian Cox problem. Comparison of the symmetrically processed methods with $b = 0.348674$ (blue circles) and $b = 0.340200$ (black diamonds) with the integrators BICaSa (red triangles) and leapfrog (green squares). On the left, acceptance percentage as a function of the number of evaluations per integration leg. On the right, acceptance percentage divided by number of gradient evaluations for different values of the step size h . For each integrator a star symbol on the marker identifies the most efficient run.

are equally efficient and their common efficiency is roughly 25% higher than that of BICaSa.

Since the number of gradient evaluations is $3N + 1$ for BICaSa and $3N + 5$ for the symmetrically processed integrators, the advantages of processing will decrease if the integration legs use very few time steps and become more marked when many time steps are taken.

6 Other uses of symmetric modified processing

As we will discuss now, HMC sampling is not the only application where symmetric modified processing may be useful.

In several problems, including the time-integration of parabolic partial differential equations or the Schrödinger equation in imaginary time (as used in path integral computations), all the coefficients appearing in a splitting algorithm have to be positive (or complex with nonnegative real part). As is well known (see [6] and its references) this sets an upper bound of two to the order of accuracy that may be achieved by unprocessed splitting integrators. On the other hand, by using modified potentials, it is possible [15, 33, 36] to construct kernels with effective order four and positive coefficients and this raises the question of how to construct suitable pre and post-processors with positive coefficients. Unfortunately, for the standard processing format

given by (1) and (5), the relations (7) imply that at least one c_i and one d_j have to be negative. This difficulty may be circumvented by symmetric modified processing as follows. If $N \geq 2$, the kernel is time reversible and we set $\kappa_h = \psi_h \circ \pi_h$, then the definition in (2) may obviously be rewritten as

$$\tilde{\psi}_{N,h} = \kappa_h^* \circ \psi_h^{N-2} \circ \kappa_h.$$

If now the map κ_h is sought in the form

$$\kappa_h = \phi_{c'_s h}^A \circ \phi_{d'_s h}^B \circ \cdots \circ \phi_{c'_1 h}^A \circ \phi_{d'_1 h}^B,$$

consistency demands

$$c'_1 + \cdots + c'_s = 1, \quad d'_1 + \cdots + d'_s = 1,$$

and these relations may be satisfied with $c_i, d_i \geq 0$, $i = 1, \dots, s$. Details will be presented in a forthcoming article.

Acknowledgement. S.B., F.C. and J.M.S.-S. have been supported by project PID2019-104927GB-C21 (AEI/FEDER, UE). M.P.C. has been supported by projects PID2019-104927GB-C22 (GNI-QUAMC) (AEI/FEDER, UE) and VA105G18 (Junta de Castilla y Leon, ES) co-financed by FEDER funds.

References

- [1] M. ALEARDI, A. SALUSTI, *Hamiltonian Monte Carlo algorithms for target- and interval-oriented amplitude versus angle inversions*, *Geophysics*, 85 (2020), R177.
- [2] E. AKHMATSKAYA, M. FERNÁNDEZ-PENDÁS, T. RADIVOJEVIC, AND J.M. SANZ-SERNA, *Adaptive splitting integrators for enhancing sampling efficiency of modified Hamiltonian Monte Carlo methods in molecular simulation*, *Langmuir*, 32 (2017), pp. 11530-11542.
- [3] A. ATTIA, AND A. SANDU, *A Hybrid Monte Carlo sampling filter for non-Gaussian data assimilation*, *AIMS Geosciences* 1 (2015), 41–78.
- [4] A. BESKOS, N.S. PILLAI, G.O. ROBERTS, AND J.M. SANZ-SERNA, *Optimal tuning of hybrid Monte-Carlo*, *Bernoulli J.*, 19 (2013), pp. 1501–1534.

- [5] N. BOU-RABEE AND J.M. SANZ-SERNA, *Geometric integrators and the Hamiltonian Monte Carlo method*, Acta Numerica, 27 (2018), pp. 113–206.
- [6] S. BLANES AND F. CASAS, *On the necessity of negative coefficients for operator splitting schemes of order higher than two*, Appl. Numer. Math., 54 (2005), pp. 23–37.
- [7] S. BLANES AND F. CASAS, *A Concise Introduction to Geometric Numerical Integration*, Chapman and Hall/CRC, Boca Raton, 2016.
- [8] S. BLANES, F. CASAS, AND J. ROS, *Symplectic integrators with processing: a general study*, SIAM J. Sci. Comput., 21 (1999), pp. 711–727.
- [9] S. BLANES, F. CASAS, AND J.M. SANZ-SERNA, *Numerical integrators for the Hybrid Monte Carlo method*, SIAM J. Sci. Comput., 36,4(2014), A1556—A1580.
- [10] J.C. BUTCHER, *The effective order of Runge-Kutta methods*, in Proceedings of the Conference on the Numerical Solution of Differential Equations, J. Ll. Morris ed., Lecture Notes in Mathematics, vol. 109, Springer, 1969, pp. 133–139.
- [11] M.P. CALVO AND J.M. SANZ-SERNA, *The development of variable-step symplectic integrators with application to the two-body problem*, SIAM J. Sci. Stat. Comput., 14 (1993), pp. 936–952.
- [12] M.P. CALVO, D. SANZ-ALONSO, AND J.M. SANZ-SERNA, *HMC: avoiding rejections by not using leapfrog and some results on the acceptance rate*, arXiv 1912.03253.
- [13] C.M. CAMPOS AND J.M. SANZ-SERNA, *Palindromic 3-stage splitting integrators, a roadmap*, J. Comput. Phys., 346 (2017), pp. 340–355.
- [14] E. CANCÈS, F. LEGOLL, AND G. STOLZ, *Theoretical and numerical comparison of some sampling methods for molecular dynamics*, ESAIM M2AN 41 (2007), pp. 351–389.
- [15] S. CHIN, *Symplectic integrators from composite operator factorizations*, Phys. Lett. A, 226 (1997), pp. 344–348.
- [16] O.F. CHRISTENSEN, G.O. ROBERTS, AND J. S. ROSENTHAL, *Scaling limits for the transient phase of local Metropolis–Hastings algorithms*, J. of the Royal Stat. Soc. B, 67 (2005), pp. 253–268.

- [17] S. DUANE, A.D. KENNEDY, B. PENDLETON, AND D. ROWETH, *Hybrid Monte Carlo*, Phys. Lett. B., 195 (1987), pp. 216–222.
- [18] M. FERNÁNDEZ-PENDAS, E. AKHMATSKAYA, AND J.M. SANZ-SERNA, *Adaptive multi-stage integrators for optimal energy conservation in molecular simulations*, J. Comput. Phys., 327 (2016), pp. 434–449.
- [19] M. GIROLAMI AND B. CALDERHEAD, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, J. Royal Stat. Soc. B, 73, (2011), pp. 123–214.
- [20] F. GOTH, *Higher order order auxiliary field quantum Monte Carlo methods*, arXiv 2009.04491v1.
- [21] E. HAIRER, C. LUBICH, AND G. WANNER, *Geometric Numerical Integration. Structure Preserving Algorithms for Ordinary Differential Equations*, 2nd ed., Springer 2006.
- [22] M. HERNÁNDEZ-SÁNCHEZ, F.-SH. KITaura, M. ATA, AND C. DALLA VECCHIA, *Higher Order Hamiltonian Monte Carlo Sampling for Cosmological Large-Scale Structure Analysis*, arXiv 1911.02667v3.
- [23] M.A. LÓPEZ-MARCOS, J.M. SANZ-SERNA, AND R.D. SKEEL, *Cheap enhancement of symplectic integrators*, Proceedings of the Dundee Conference on Numerical Analysis, D. F. Griffiths and G. A. Watson eds., Longman, 1966.
- [24] M.A. LÓPEZ-MARCOS, J. M. SANZ-SERNA, AND R. D. SKEEL, *Explicit symplectic integrators using Hessian-vector products*, SIAM J. Sci. Comput., 18 (1997), pp. 223–238.
- [25] J. MANNSETH, T.S. KLEPPE, H.J. SKANG, *On the application of improved symplectic integrators in Hamiltonian Monte Carlo*, Commun. Stats., Simul. and Comp., 147 (2018), pp. 500-509
- [26] R.I. McLACHLAN, *On the numerical integration of ordinary differential equations by symmetric composition methods*, SIAM J. Sci. Comp. 16 (1995), pp. 151-168.
- [27] R. McLACHLAN AND R. QUISPel, *Splitting methods*, Acta Numerica, 11 (2002), pp. 341–434.
- [28] R.M. NEAL, *MCMC using Hamiltonian dynamics*, in Handbook of Markov Chain Monte Carlo, S. Brooks, A. Gelman, G. Jones, X. L.

- Meng, eds., Chapman and Hall/CRC Press, London, 2011, pp. 113–162.
- [29] A. NISHIMURA, D.B. DUNSON, J. LU, *Discontinuous Monte Carlo for discrete parameters and discontinuous likelihoods*, *Biometrika* 107 (2020), pp. 365–380
- [30] I. OMELYAN, I. MRYGOLD, AND R. FOLK, *Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations*, *Comput. Phys. Comm.*, 151 (2003), pp. 273–314.
- [31] C. PREDESCU, R.A. LIPPERT, M.P. EASTWOOD, D. IERARDI, H. XU, M. Ø. JENSEN, K.J. BOWERS, J. GULLINGSRUD, C.A. RENDLEMAN, R. O. DROR, ET AL, *Computationally efficient molecular dynamics integrators with improved sampling accuracy*, *Molec. Phys.*, 110 (2012), pp. 967–983.
- [32] T. RADIVOJEVIC, M. FERNÁNDEZ-PENDÁS, J.M. SANZ-SERNA, AND E. AKHMATSKAYA, *Multi-stage splitting integrators for sampling with modified Hamiltonian Monte Carlo methods*, *J. Comput. Phys.*, 373 (2018), pp. 900–916.
- [33] G. ROWLANDS, *A numerical algorithm for Hamiltonian systems*, *J. Comput. Phys.*, 97 (1991), pp. 235–239.
- [34] J.M. SANZ-SERNA, *Markov Chain Monte Carlo and numerical differential equations*, in *Current Challenges in Stability Issues for Numerical Differential Equations*, L. Dieci and N. Guglielmi, eds., Springer, Cham, 2013, pp. 39–88.
- [35] J.M. SANZ-SERNA AND M.P. CALVO, *Numerical Hamiltonian Problems*, Chapman and Hall, London, 1994.
- [36] T. TAKAHASHI AND M. IMADA, *Montecarlo calculation of quantum system II. Higher order correction*, *J. Phys. Soc. Japan*, 53 (1984), pp. 3765–3769.