

Adaptive tuning of Hamiltonian Monte Carlo methods

Elena Akhmatskaya ^{a,b,*}, Lorenzo Nagar ^{a,c,*}, Jose Antonio Carrillo ^d,
Leonardo Gavira Balmaciz ^{a,c}, Hristo Inouzhe ^e, Martín Parga Pazos ^f,
María Xosé Rodríguez Álvarez ^g

^a BCAM – Basque Center for Applied Mathematics, Bilbao, Spain

^b Ikerbasque – Basque Foundation for Science, Bilbao, Spain

^c UPV/EHU – Universidad del País Vasco / Euskal Herriko Unibertsitatea, Basque Country, Spain

^d Mathematical Institute, University of Oxford, Oxford, United Kingdom

^e Departamento de Matemáticas, Universidad Autónoma de Madrid, Madrid, Spain

^f CIC bioGUNE – Center for Cooperative Research in Biosciences, Derio, Spain

^g Departamento de Estatística e Investigación Operativa, Universidade de Vigo, Vigo, Spain

ARTICLE INFO

Keywords:

Hamiltonian Monte Carlo
Generalized Hamiltonian Monte Carlo
Adaptive hyperparameters tuning
Adaptive multi-stage splitting integration
Randomization intervals
Biomedical applications

ABSTRACT

With the recently increased interest in probabilistic models, the efficiency of an underlying sampler becomes a crucial consideration. Hamiltonian Monte Carlo is one popular option for models of this kind. Performance of the method, however, strongly relies on a choice of parameters associated with an integration approach for Hamiltonian equations. Up to date, such a choice remains mainly heuristic or introduces time complexity. We propose a novel computationally inexpensive and flexible approach (we call it Adaptive Tuning or ATune) that, by combining a theoretical analysis of the multivariate Gaussian model with simulation data generated during a burn-in stage of a Hamiltonian Monte Carlo simulation, detects a system specific splitting integrator with a set of reliable sampler's hyperparameters, including their credible randomization intervals, to be readily used in a production simulation. The method automatically eliminates those values of simulation parameters which could cause undesired extreme scenarios, such as resonance artifacts, low accuracy or poor sampling. The new approach is implemented in the in-house software package HaiCS, with no computational overheads introduced in a production simulation, and can be easily incorporated in any package for Bayesian inference with Hamiltonian Monte Carlo. The tests on popular statistical models reveal the superiority of adaptively tuned standard and generalized Hamiltonian Monte Carlo methods in terms of stability, performance and accuracy over conventional Hamiltonian Monte Carlo tuned heuristically and coupled with the well-established integrators. We also claim that the generalized Hamiltonian Monte Carlo is preferable for achieving high sampling performance. The efficiency of the new methodology is assessed in comparison with state-of-the-art samplers, e.g. the No-U-Turn-Sampler, in real-world applications, such as endocrine therapy resistance in cancer, modeling of cell-cell adhesion dynamics and influenza A epidemic outbreak.

* Corresponding authors.

E-mail addresses: eakhmatskaya@bcamath.org (E. Akhmatskaya), lnagar@bcamath.org (L. Nagar), carrillo@maths.ox.ac.uk (J.A. Carrillo), lgavira@bcamath.org (L. Gavira Balmaciz), hristo.inouze@uam.es (H. Inouzhe), mparga@cicbiogune.es (M. Parga Pazos), mxrodriguez@uvigo.gal (M.X. Rodríguez Álvarez).

<https://doi.org/10.1016/j.apm.2026.116892>

Received 20 September 2025; Received in revised form 3 March 2026; Accepted 4 March 2026

Available online 8 March 2026

0307-904X/© 2026 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Hamiltonian Monte Carlo (HMC) [1] has emerged as a powerful and widely adopted methodology for sampling from high-dimensional complex probability distributions, making it invaluable in Bayesian inference applications. By leveraging the Hamiltonian equations of motion for generating Monte Carlo proposals, HMC exploits gradient information on posterior distributions to enhance the convergence of the Markov chain and enables efficient exploration of the parameter space.

Further enhancing the sampling performance of HMC has attracted growing attention within the computational statistics community in recent years [2–9].

The Markov kernel of HMC based on Hamiltonian dynamics induces a set of hyperparameters and tuning blocks that critically impact sampling performance and include a numerical integrator for solving Hamiltonian equations, an integration step size, a number of integrations per iteration, a mass matrix. For instance, while a smaller step size ensures high acceptance rates and reduces computational waste, it also shortens proposal moves, increasing correlation between samples. Conversely, a larger number of integration steps per iteration enables longer trajectories and more independent samples but comes at the cost of increased computational overhead due to additional gradient evaluations per iteration. Consequently, significant research efforts have been devoted to identifying optimal parameter settings for HMC to balance computational efficiency and sampling performance.

Early strategies for tuning the integration step size focused on targeting an optimal acceptance rate for the Metropolis test [10,11] or employing on-the-fly adaptation using primal-dual averaging algorithms [12]. Some sophisticated integration schemes have also been proposed, replacing the standard Verlet algorithm [13] with more accurate multi-stage splitting integrators to enhance sampling performance and efficiency [14–19]. Comprehensive reviews on numerical integration in HMC are available in [20,21].

Regarding the number of integration steps per iteration, a common recommendation is randomization [22], though more advanced algorithms have been proposed, such as automated trajectory length selection based on the double-backing criterion [12].

Additional contributions to HMC tuning include position-dependent mass matrix adaptation [23] and methods for determining the proper number of Monte Carlo iterations to ensure convergence [24].

More recent advancements in optimization algorithms for HMC hyperparameters address both integration step size [3,6–9,18] and number of integration steps per iteration [2,4,6,8,9].

In this study, we propose a computationally inexpensive adaptive tuning procedure that identifies an optimal numerical integrator along with a set of HMC hyperparameters tailored to a given simulation system. We call this procedure ATune and refer to an HMC method reinforced with such tuning as Adaptively Tuned HMC (AT-HMC).

Furthermore, we extend our focus to Generalized Hamiltonian Monte Carlo (GHMC) [25], a promising but relatively unexplored variant of standard HMC.

The key distinction between HMC and GHMC lies in the update mechanism for the momenta. In HMC, momenta are completely discarded after the Metropolis test and resampled at the beginning of each iteration from a normal distribution with mass matrix covariance. On the contrary, GHMC performs a partial momentum update (PMU) [26]. The PMU introduces an additional hyperparameter which, if not carefully chosen, can degrade sampling performance. This likely explains why GHMC has been less commonly used than HMC.

However, when properly tuned, GHMC can offer significant improvements in sampling accuracy and efficiency. There are two primary reasons for that.

First, in contrast to HMC, GHMC produces irreversible chains by construction [27]. This usually implies a faster convergence to equilibrium and a reduction in asymptotic variance [28–30]. Second, the partial momentum update in GHMC with a small momentum mixing angle makes a choice of the number of steps per integration leg less critical than in the case of HMC, enabling efficient sampling with shorter Hamiltonian trajectories [31], thereby enhancing both the speed and efficiency of the sampling process.

Although theoretical studies recognize potential of GHMC, little effort has been devoted to developing effective tuning strategies. Only recently, GHMC has begun to attract growing interest within the scientific community [32–35].

Building on these insights, we extend our Adaptive Tuning to GHMC to derive AT-GHMC. Apart from the previously discussed in the literature choices of HMC and GHMC simulation parameters, we add to our optimal settings list randomization intervals for simulation parameters. Indeed, randomization of simulation parameters is part of any HMC-based algorithm [36], but until now its choice remains purely heuristic.

In summary, we aim to find optimal, or close to optimal choices for:

- an integration scheme for the Hamiltonian equations of motion
- a dimensional integration step size and its randomization interval
- a number of integration steps per Monte Carlo iteration and its randomization interval
- a noise control parameter (along with the appropriate randomization interval) in the PMU.

Our approach relies on the adaptive s-AIA integration method [17], which provides useful guidance on choices of integrators and optimal step size locations. According to [17], s-AIA integrators outperform the standard Velocity Verlet and other fixed-parameter multi-stage integrators in terms of both integration accuracy and their impact on sampling efficiency of HMC, for any simulation step size within a system-specific stability interval. Additionally, a key observation in [17] is that the highest performance of HMC is achieved near the center of the stability interval, provided a stability limit is accurately estimated. This also was suggested in [15,37]. Using these findings and the analysis underlying the s-AIA approach, we identify optimal integration schemes and step size randomization intervals for both AT-HMC and AT-GHMC, as well as an optimal randomization scheme for a number of integration steps in AT-GHMC.

Moreover, we revisit the optimization strategies previously proposed for Modified HMC in [38], and adapt them to GHMC, refining the momentum refreshment procedure to derive an optimal, system-dependent noise parameter in the PMU. Using the proposed settings, GHMC achieves efficient phase-space exploration even with a small number of integration steps [31].

The new tuning methodology does not incur additional computational overhead, as the optimal values are either system-specific (and can be determined a priori) or computed during a burn-in stage. Numerical experiments on standard benchmarks confirm the positive effect of proposed optimal settings on HMC and GHMC performance.

Additionally, we assess the efficiency of AT-HMC and AT-GHMC on three real-world applications, each relying on a different probabilistic model, and compare the results with those obtained using the state-of-the-art samplers commonly employed in Bayesian inference on such models. The applications include:

- Patient resistance to endocrine therapy in breast cancer – a Bayesian Logistic Regression (BLR) model
- Cell-cell adhesion dynamics – a Partial Differential Equation (PDE) model
- Influenza A (H1N1) epidemics outbreak – an Ordinary Differential Equation (ODE) model.

The paper is organized as follows. Section 2 presents the essential background, covering HMC, GHMC and the numerical integration of Hamiltonian dynamics. Section 3 revisits the numerical experiments in [17], showing that optimal performance is consistently achieved near the center of a stability interval. In Section 4, we present the refined randomization interval for the optimal simulation step size and support our choice by numerical experiments.

Sections 5 and 6 introduce the optimization procedures for the random noise parameter in the PMU and the number of integration steps per iteration. The Adaptive Tuning algorithm ATune is outlined in Section 7, validated on the set of benchmarks in Section 8, and systematically tested in the three real-world application studies in Section 9. Section 10 summarizes our conclusions. Additional technical details and simulations are provided in [39], which includes the appendices.

2. Hamiltonian Monte Carlo

2.1. Standard formulation

Hamiltonian Monte Carlo [1] is a Markov Chain Monte Carlo (MCMC) method for obtaining N correlated samples $\{\theta_n\}_{n=1,\dots,N}$ from a target probability distribution $\pi(\theta)$ in \mathbb{R}^D , D being the dimension of θ . HMC generates a Markov chain in the joint phase space $\Omega \subseteq \mathbb{R}^{2D}$ with invariant distribution

$$\pi(\theta, p) = \pi(\theta)n(p) \propto \exp(-H(\theta, p)), \tag{1}$$

and recovers $\pi(\theta)$ by marginalizing out the auxiliary momentum variable p (distributed according to $n(p)$).

In (1), $H(\theta, p)$ is the Hamiltonian

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

where M is the symmetric positive definite mass matrix, and $U(\theta)$ is the potential energy, which contains information about the target distribution $\pi(\theta)$:

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

HMC generates new proposals (θ', p') by alternating momentum update steps, where p is drawn from a Gaussian $\mathcal{N}(0, M)$, with the steps that update position θ and momenta p through the numerical integration of the Hamiltonian equations

$$\frac{d\theta}{dt} = M^{-1}p, \quad \frac{dp}{dt} = -\nabla_{\theta}U(\theta), \tag{2}$$

performed L times using an explicit symplectic and reversible integrator Ψ_h [40] (h being an integration step size). Thus,

$$(\theta', p') = \underbrace{\Psi_h \circ \dots \circ \Psi_h}_{L \text{ times}}(\theta, p). \tag{3}$$

Symplecticity and reversibility of Ψ_h ensure that the generated Markov chain is ergodic, thus that $\pi(\theta, p)$ is an invariant measure for the generated Markov chain.

Due to numerical integration errors, the Hamiltonian energy and hence the target density (1) are not exactly preserved. To address this issue, the invariance of the target density is secured through a Metropolis test with the acceptance probability

$$\alpha = \min\{1, \exp(-\Delta H)\},$$

where

$$\Delta H = H(\theta', p') - H(\theta, p)$$

is the energy error resulting from the numerical integration (3). In case of acceptance, θ' becomes the starting point for the subsequent iteration. Conversely, if the proposal is rejected, the previous state θ is retained for the next iteration. Regardless of acceptance or rejection, the momentum variable is discarded, and a new momentum p is drawn from $\mathcal{N}(0, M)$.

2.2. Generalized Hamiltonian Monte Carlo

Generalized Hamiltonian Monte Carlo [25] incorporates a partial momentum update (PMU) [26] in standard Hamiltonian Monte Carlo to replace the full refreshment of auxiliary momenta. The PMU reduces random-walk behavior by partially updating the auxiliary momenta p through mixing with an independent and identically distributed (i.i.d) Gaussian noise $u \sim \mathcal{N}(0, M)$:

$$p \leftarrow \sqrt{1 - \varphi} p + \sqrt{\varphi} u. \tag{4}$$

Here, $\varphi \in (0, 1]$ is a random noise parameter which determines the extent to which the momenta are refreshed. When $\varphi = 1$, the update (4) reduces to the standard HMC momentum resampling scheme.

Notice that the PMU produces configurations that preserve the joint target distribution, as the orthogonal transformation in (4) maintains the distributions of $p \sim \mathcal{N}(0, M)$. Since the momenta are not fully discarded, GHMC performs a momentum flip after each rejected proposal. As shown in [36] and formally proved in [31], the three core steps of GHMC (Hamiltonian dynamics combined with the Metropolis test, the PMU, and momentum flip) each individually satisfy detailed balance and maintain reversibility, while their combination does not. This means that the resulting Markov chains are not reversible. Nevertheless, [31] demonstrated that GHMC fulfils the weaker modified detailed balance condition, ensuring the stationarity of the generated Markov chains.

2.3. Numerical integration of Hamiltonian dynamics

The most commonly used numerical scheme for solving the Hamiltonian Eqs. (2) is the *Velocity Verlet* integrator [13]. This scheme is widely favored due to its simplicity, good stability, and energy conservation properties. The algorithm reads

$$\begin{aligned} p &\leftarrow p - \frac{h}{2} \nabla_{\theta} U(\theta), \\ \theta &\leftarrow \theta + h M^{-1} p, \\ p &\leftarrow p - \frac{h}{2} \nabla_{\theta} U(\theta). \end{aligned} \tag{5}$$

Here, h represents the dimensionless integration step size. We will use a notation Δt for its dimensional counterpart. The updates of momenta and position in (5) are often called momentum *kick* and position *drift*, respectively.

More sophisticated integration schemes have been proposed in literature, including *multi-stage palindromic splitting integrators* [21]. These methods perform multiple splits of the separable Hamiltonian system (2), applying a sequence of kicks and drifts (5) with arbitrary chosen step sizes, in contrast to the half-kick, drift, half-kick structure of the standard Verlet method. By denoting a position drift and a momentum kick, respectively, with

$$\varphi_h^A(\theta, p) = (\theta + h M^{-1} p, p), \quad \varphi_h^B(\theta, p) = (\theta, p - h \nabla_{\theta} U(\theta)),$$

the family of k -stage splitting integrators is defined as [20]

$$\Psi_h^k = \varphi_{b_1 h}^B \circ \varphi_{a_1 h}^A \circ \dots \circ \varphi_{a_{k'} h}^A \circ \varphi_{b_{k'+1} h}^B \circ \varphi_{a_{k'} h}^A \circ \dots \circ \varphi_{a_1 h}^A \circ \varphi_{b_1 h}^B, \quad b_i, a_j \in \mathbb{R}^+, \tag{6}$$

if $k = 2k'$, and

$$\Psi_h^k = \varphi_{b_1 h}^B \circ \varphi_{a_1 h}^A \circ \dots \circ \varphi_{b_{k'} h}^B \circ \varphi_{a_{k'} h}^A \circ \varphi_{b_{k'} h}^B \circ \dots \circ \varphi_{a_1 h}^A \circ \varphi_{b_1 h}^B, \quad b_i, a_j \in \mathbb{R}^+, \tag{7}$$

if $k = 2k' - 1$. The coefficients b_i, a_j in (6)–(7) have to satisfy the conditions $2 \sum_{i=1}^{k'} b_i + b_{k'+1} = 2 \sum_{j=1}^{k'} a_j = 1$, and $2 \sum_{j=1}^{k'-1} a_j + a_{k'} = 1$, respectively, to ensure an integration step of length h . A term *stage* (k) refers to the number of gradient evaluations per integration. It is straightforward to verify that the only 1-stage splitting integrator of the form (7) is the Velocity Verlet (VV) scheme (5), which can be expressed as

$$\Psi_h^{VV} = \varphi_{\frac{h}{2}}^B \circ \varphi_{\frac{h}{2}}^A \circ \varphi_{\frac{h}{2}}^B.$$

Various choices of integration coefficients b_i, a_j in (6)–(7) yield integrators that reach their pick performance at different specific values of a simulation step size h . To address the step size dependency, adaptive integration approaches AIA [16] and s-AIA [17] were developed for molecular simulations and Bayesian inference applications, respectively. For a given system and simulation step size h , these methods select the most suitable 2- (AIA, s-AIA2) and 3- (s-AIA3) stage splitting integrator ensuring optimal energy conservation for harmonic forces. Examples of 2- and 3-stage integrators, including those used in this study, are reviewed in [15,17,41].

3. Tracking the location of optimal step size

At first, we aim to demonstrate that top performance of HMC is typically achieved with a step size being near the center of the longest stability interval (CoLSI) for an integrator in use, as previously suggested in [37] and lately confirmed in [15,17]. We will use notations h_{CoLSI} and Δt_{CoLSI} for such dimensionless and dimensional step sizes, respectively. We recall that for a k -stage integrator, the length of the longest dimensionless stability interval equals $2k$, hence, $h_{\text{CoLSI}} = k$ [15].

Table 1

Multi-stage splitting integrators considered for this study. $\mathbb{E}[\Delta H]$ is the expected energy error resulting from the numerical integration of Hamiltonian dynamics, $h_{\text{CoLSI}} = k$ is the centre of the longest stability interval.

Integrator	N. of stages (k)	Feature	References
VV	1	longest stability interval	[13]
BCSS k	2, 3	minimizes $\mathbb{E}[\Delta H]$ in $(0, h_{\text{CoLSI}}) = k$	[15]
ME k	2, 3	minimizes truncation error for $h \rightarrow 0$	[42,43]
sAIA k	2, 3	minimizes $\mathbb{E}[\Delta H] \forall h \in (0, 2k)$	[17]

Table 2

Best grad/minESS (gESS*) performance, detected in the HMC simulations in [17] for the three benchmarks, using best performed integrator at h_{gESS^*} integration step size (normalized to the number of stages k).

Benchmark	k	gESS*	h_{gESS^*} / k	Integrator
G1000	2	6402.3	1.0	s-AIA2
	3	5177.1	1.1	BCSS3
German	2	28.8	0.9	AIA2
	3	26.4	0.8	ME3
Musk	2	249.3	0.8	BCSS2
	3	252.8	0.9	s-AIA3

We consider the numerical experiments presented in [17] for the three benchmarks – 1000-dimensional multivariate Gaussian model, G1000, and two BLR models, German and Musk – performed with the numerical integrators summarized in Table 1. For further details on benchmarks, see Table 4 and [17].

For each benchmark, we identify the integrator and integration step size which lead to the lowest ratio between the number of gradient evaluations, grad (which represents the bulk of the computational cost in an HMC simulation), and the minimum effective sample size [44] across variates, minESS, computed for HMC production chains of the length of $N_{\text{conv}} + 1000$ iterations. Here N_{conv} is the number of samples in the production chains, required for reaching convergence [45]. We choose minESS (calculated through the effectiveSize function of the coda package of R [46]) as the most demanding one among the metrics considered in [17].

We refer to the performance metric introduced in such a way, i.e. grad/minESS, as gESS. In Section 8, we provide a detailed discussion of this metric and explore more performance metrics for validation of the proposed methodology. We emphasize that lower values of grad and higher minESS suggest faster convergence and better sampling efficiency, respectively, meaning that lower gESS indicates the improved overall performance.

Table 2 displays the best grad/minESS performance, i.e. the lowest $\text{gESS} := \text{gESS}^*$ achieved for each benchmark with k -stage integrators, $k = 2, 3$, and specifies the integrators along with the corresponding step sizes (normalized to the number of stages for clarity, i.e. normalized h_{CoLSI} is equal to 1 for all k -stage integrators) responsible for such performance. The results are visualized in Fig. 1 using normalized step sizes.

From Table 2, one can observe that top performance is consistently achieved around the CoLSI, $h_{\text{CoLSI}} = 1$, and thus h_{CoLSI} is an obvious candidate for an optimal step size. The deviation of h_{gESS^*} from h_{CoLSI} is not critical at this stage as an optimal step size is to be randomized yet, and only performance at a randomized step matters.

Furthermore, 3-stage integrators always outperform 2-stage ones, as already noticed in [47] and [17]. Hence, it is natural to search for an optimal integrator within the class of 3-stage integrators. Unfortunately, Table 2 does not suggest a unique candidate for the best performed integrator in this class.

We remark that according to [17], BCSS3 and s-AIA3 around CoLSI give rise to very similar accuracy and performance of HMC for all considered benchmarks. For the German benchmark, the efficiencies of all the three 3-stage integrators, i.e. BCSS3, ME3 and s-AIA3, at CoLSI are almost indistinguishable. Thus, it is not surprising that all those integrators appear in Table 2. However, only s-AIA3 adapts its performance to a choice of a step size, and this property becomes important when randomization of a step size is applied. In this context, s-AIA3 looks like the best pick, but this has to be confirmed yet. In the next Sections, we propose a randomization interval for h_{CoLSI} and use the 3-stage integrators (BCSS3, ME3, s-AIA3) and the standard 1-stage Velocity Verlet for its validation.

4. Randomizing optimal step size

Further, we propose an approach for finding an optimal randomization interval for an integration step size h_{CoLSI} (and for its dimensional counterpart Δt_{CoLSI}) and compare the resulted HMC performance with the best grad/minESS performance in Table 2.

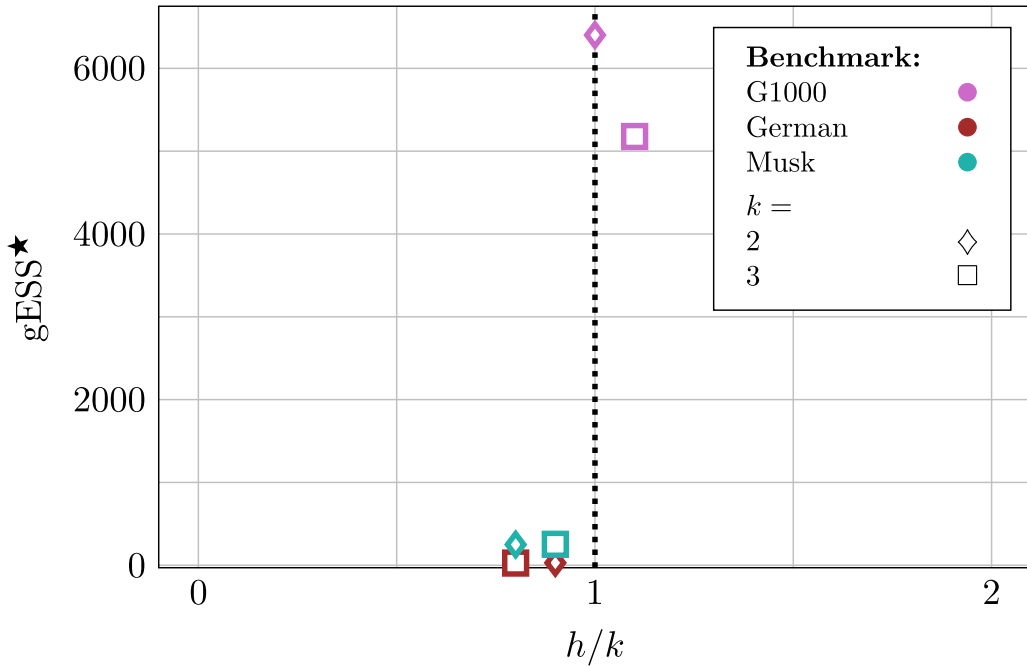


Fig. 1. Best grad/minESS performance (gESS*) obtained for the tested benchmarks – G1000 (mauve), German (dark red) and Musk (turquoise) (Table 2). k is the number of stages of an integrator. For any benchmark, top performance is achieved around the CoLSI, i.e. $h = k$.

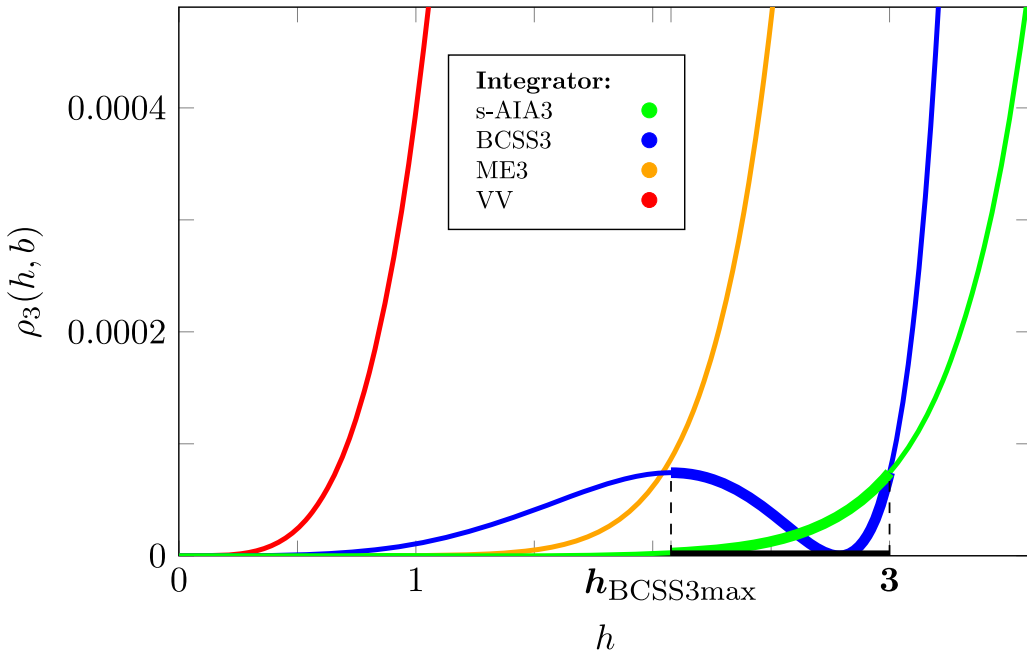


Fig. 2. The upper bound $\rho_{3\text{-stage}}(h, b)$ of the expected energy error for 3-stage integrators.

4.1. Randomization interval

To identify endpoints of a randomization interval for an optimal step size we refer to the upper bound of the expected energy error for 3-stage integrators $\rho_3(h, b)$ derived in [17] and plotted for the range of integrators considered in this study in Fig. 2.

When searching for an optimal randomization interval for h_{CoLSI} the following considerations have to be taken into account. First, the interval should include the step sizes that guarantee the high accuracy of the numerical integration, i.e. $\rho_3(h, b) \ll 1$. Then, one

Table 3
HMC and GHMC parameter settings for the numerical experiments presented in Section 4.2. Integration step sizes for each randomization interval are provided in dimensional units.

Benchmark	Sampler	Integrator	L	Δt	φ
G1000	HMC	s-AIA3			1.0
		BCSS3	$\mathcal{U}\{1, \dots, 2666\}$	$\mathcal{U}(0.036, 0.052)$	
		ME3			
	GHMC	VV	$\mathcal{U}\{1, \dots, 7998\}$	$\mathcal{U}(0.012, 0.017)$	
		s-AIA3			$\mathcal{U}(0.0, 0.5)$
		BCSS3	$\mathcal{U}\{1, \dots, 1332\}$	$\mathcal{U}(0.036, 0.052)$	
German	HMC	ME3			
		VV	$\mathcal{U}\{1, \dots, 49\}$	$\mathcal{U}(0.037, 0.053)$	
		s-AIA3			1.0
	GHMC	BCSS3	$\mathcal{U}\{1, \dots, 16\}$	$\mathcal{U}(0.111, 0.160)$	
		ME3			$\mathcal{U}(0.0, 0.5)$
		VV	$\mathcal{U}\{1, \dots, 24\}$	$\mathcal{U}(0.037, 0.053)$	
Musk	HMC	s-AIA3			1.0
		BCSS3	$\mathcal{U}\{1, \dots, 110\}$	$\mathcal{U}(0.083, 0.110)$	
		ME3			
	GHMC	VV	$\mathcal{U}\{1, \dots, 333\}$	$\mathcal{U}(0.028, 0.040)$	
		s-AIA3			$\mathcal{U}(0.0, 0.5)$
		BCSS3	$\mathcal{U}\{1, \dots, 55\}$	$\mathcal{U}(0.084, 0.121)$	
		VV	$\mathcal{U}\{1, \dots, 166\}$	$\mathcal{U}(0.028, 0.040)$	

should keep interval narrow enough to stay close to an estimated optimal value of a step size and to avoid including too small step sizes, which usually satisfy the aforementioned inequality but lead to correlated samples in HMC. Finally, the approximate analysis proposed in [17] may likely result in the overestimated values of the stability limit and thus Δt_{CoLSI} (see Appendix A in [39]). This implies a reasonable choice for the right endpoint to be h_{CoLSI} .

On the other hand, we recall that BCSS3 was derived to minimize the maximum of $\rho_3(h, b)$ in $0 < h < h_{\text{CoLSI}}$ and thus to achieve the best performance near h_{CoLSI} . The inspection of Fig. 2 reveals that $\rho_3(h, b)$ with $b = b_{\text{BCSS3}}$ exhibits a local maximum h_{BCSS3max} on the left side of h_{CoLSI} . Within the interval $(h_{\text{BCSS3max}}, h_{\text{CoLSI}})$ both s-AIA3 and BCSS3 demonstrate very high accuracy, which deteriorates after h passes the center of the stability interval. Thus, such an interval obeys all requirements on an optimal randomization interval summarized above.

The local maxima of $\rho_{\text{BCSS3}}(h)$, i.e. h_{BCSS3max} , can be obtained by setting $\frac{\partial \rho_3(h, b)}{\partial h} = 0$ and $b = b_{\text{BCSS3}}$ [15] (see Appendix B in [39]) which yields $h_{\text{lower}} \equiv h_{\text{BCSS3max}} \approx 2.0772$.

Therefore, the proposed randomization interval for the nondimensional optimal integration step size is

$$h \sim \mathcal{U}(h_{\text{lower}}, 3) \approx \mathcal{U}(2.0772, 3), \tag{8}$$

where \mathcal{U} denotes the uniform distribution, and the corresponding interval for the dimensional step size is

$$\Delta t \sim \mathcal{U}(\Delta t_{\text{lower}}, \Delta t_{\text{CoLSI}}), \tag{9}$$

where Δt_{lower} and Δt_{CoLSI} are calculated as proposed in [17] and briefly described in [39], Appendix C.

4.2. Validation

To validate the proposed optimal randomized step size and to choose an optimal integrator among the 3-stage integrators, we run HMC simulations with s-AIA3, BCSS3, ME3 and VV at Δt in (9) (in 1-stage units for VV) for all benchmarks introduced in Section 3 and listed in Table 2.

The remaining HMC parameters, namely, a number of integration steps per Monte Carlo iteration, L , and numbers of burn-in and production iterations, are kept the same as in the numerical experiments in [17]. In addition, we conduct GHMC simulations to evaluate the efficacy of the proposed randomization interval with this sampler in comparison with HMC. For the PMU in Eq. (4), we set φ to be uniformly selected from the interval (0.0, 0.5) (a standard recommendation) and use twice shorter trajectories than for HMC. We remark that this choice of GHMC parameters is based on heuristic intuition. The optimal selection of these parameters are investigated in the following sections.

The simulation parameters for HMC and GHMC used in this section are summarized in Table 3.

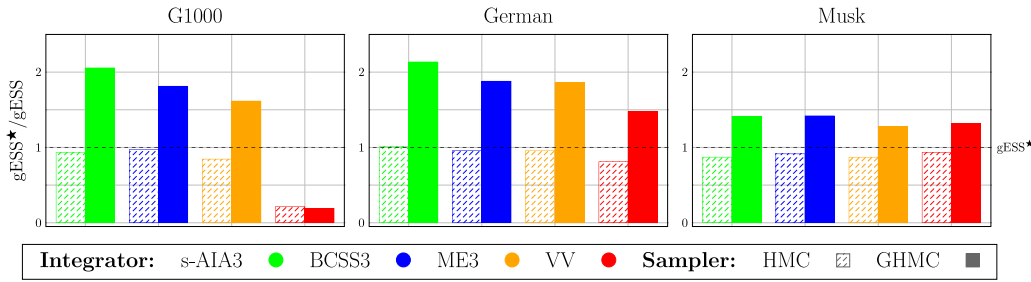


Fig. 3. Comparison of $gESS^*$ performance (Table 2) with $gESS$ observed in HMC (dashed bars) and GHMC (filled bars) with $\Delta t \sim \mathcal{U}(\Delta t_{lower}, \Delta t_{CoLSI})$ for G1000 (left), German (center) and Musk (right) benchmarks using s-AIA3 (green), BCSS3 (blue), ME3 (orange), VV (red) integrators. GHMC combined with s-AIA3 (green filled bars) always outperforms $gESS^*$ in Table 2 and demonstrates best performance in all three benchmarks. HMC (green dashed bars) also shows the best performance with s-AIA3, which is close to $gESS^*$ in Table 2 but not necessarily better.

We emphasize that the comparison between the adaptive s-AIA3 integrator and the other fixed-parameter integrators is fair in terms of computational cost and does not require further normalization with respect to computational time. Indeed, the optimal interval (9) is determined once and shared across all integrators; the average number of gradient evaluations per iteration $\bar{L}k$ (\bar{L} being the mean of the number of integration steps per Monte Carlo iteration) is kept the same for all of them (see Table 3), and the adaptive integration coefficients in s-AIA3 are identified via a pretabulated map [48] (see also [17] for further details), which incurs no additional computational overhead during the simulation.

Fig. 3 compares performance (in terms of $gESS$) of HMC and GHMC using Δt_{CoLSI} and a randomization interval (9) with $gESS^*$ in Table 2 through $gESS^*/gESS$ relations.

First, we recognize that, as expected, for all benchmarks and for both HMC and GHMC, the adaptive s-AIA3 integrator yields either better or similar sampling performance (in terms of $gESS$) compared with other tested integrators. Moreover, for GHMC, this performance is visibly higher (up to 2x) than $gESS$ observed in HMC with the newly proposed settings (see Table 3) and with the previously reported setup (Table 2). We notice that the new settings for HMC with s-AIA3 do not necessarily improve the results in Table 2 for each benchmark, but they always stay close to them. We recall that the $gESS^*$ values in Table 2 for each benchmark were achieved with various integrators and at different step sizes by trial and error, whereas our new settings were generated by the procedure universal for all simulated systems.

Additionally, we perform a sensitivity analysis on the lower endpoint of the randomization interval (8) to validate our choice of h_{lower} . We show that variations of $\pm 5\%$ around h_{lower} produce similar values for $gESS$ and the additional performance metrics proposed in Section 8.1, and thus confirm the robustness of such a choice, despite its numerical nature. The results are summarized in [39], Appendix D.

In summary, the numerical experiments not only justify the proposed optimal choices for a numerical integrator (s-AIA3), a step size ($h_{CoLSI} / \Delta t_{CoLSI}$), and a randomization interval ((8) / (9)), which we will consider for the rest of this study, but also reveal the performance superiority of GHMC over HMC.

We remark that, in contrast to the HMC experiments which relied on the recommended choices of L , the results for GHMC in Fig. 3 were obtained without a proper tuning of L and its additional parameter φ for the PMU. Obviously, refining these two parameters can further improve performance of GHMC.

Our next objective is to search for optimal, or close to optimal choices of the remaining GHMC parameters.

5. Randomization interval for φ

We note that GHMC is not the sole HMC-based method which partially refreshes the momenta. The Modified Hamiltonian Monte Carlo, or MHMC, methods for sampling with modified Hamiltonians also rely on the PMU procedure, though the PMU is Metropolized to secure sampling in a modified ensemble [38]. Both GHMC and MHMC use Hamiltonian dynamics for generating proposals but they are accepted more frequently in MHMC due to the better conservation of modified Hamiltonians by symplectic integrators. On the contrary, partially refreshed momentum is always accepted in GHMC, whereas for MHMC it is accepted according to the modified Metropolis test. Clearly, the MHMC method with very high acceptance rates in the PMU and the GHMC method with very high acceptance rate in the Metropolis test will behave almost indistinguishably.

Here, we aim to derive an optimal φ parameter for a GHMC method with its optimal settings, i.e. when using the s-AIA3 integrator with (8) or (9). With this in mind, we refer to the procedure proposed in [38] which links φ in MHMC with the target acceptance rate in the modified Metropolis test in the PMU (see Eq. (14) in the aforementioned study). Following this idea, we express φ in terms of a step size, parameters of the integrator in use and the target acceptance rate to obtain

$$\varphi = -\ln \bar{\alpha} \frac{1 + 2h^2\lambda}{2Dh^4\lambda^2}. \tag{10}$$

Here, $\bar{\alpha}$ is the expected acceptance rate for updated momentum, D is the dimension of the system, h is the dimensionless counterpart of the integration step size Δt in use, $\lambda = \lambda_k(z)$ is the coefficient depending on k – a number of stages of a numerical integrator – and

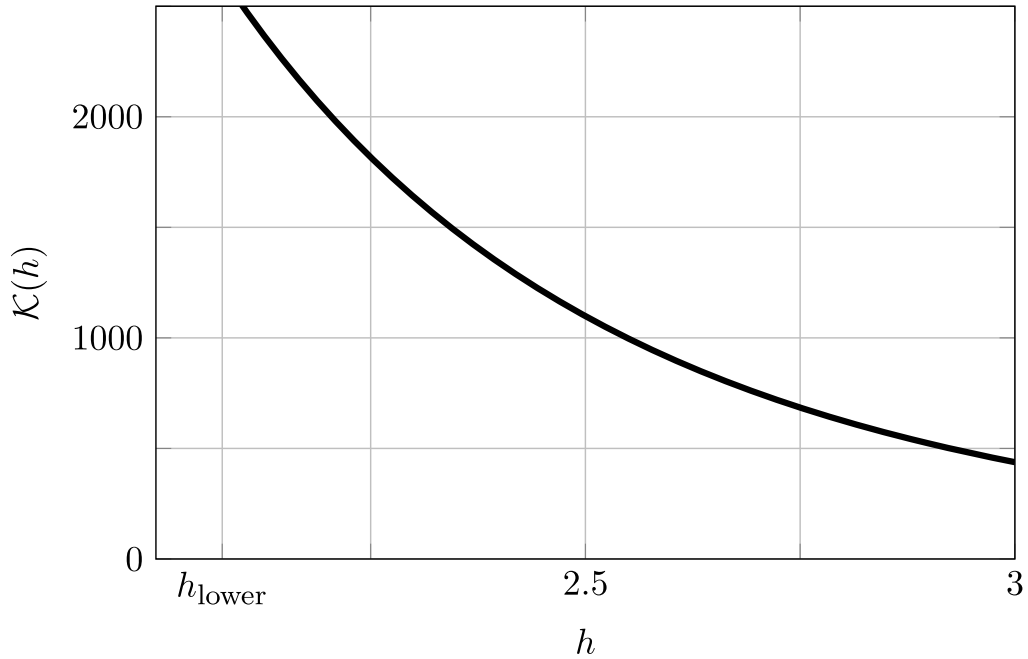


Fig. 4. Plot of $\mathcal{K}(h)$ (Eq. 11) for $h \in (h_{\text{lower}}, h_{\text{CoLSI}})$ (8).

z – the integrator coefficients. In particular [47],

$$\lambda_2(b) = \frac{6b - 1}{24}, \quad \lambda_3(b, a) = \frac{1 - 6a(1 - a)(1 - 2b)}{12}.$$

We remark that in the notations of this study φ corresponds to $\sin^2 \phi$ in [38]. Next, we adapt (10) to the conditions of this work. More precisely, we are interested in optimizing performance of GHMC with s-AIA3 (i.e. $k = 3$) in $(h_{\text{lower}}, h_{\text{CoLSI}})$. We recall that for s-AIA3, the integrator parameters a and b are step size dependent and so is λ_3 .

Given that for GHMC the expected acceptance rates for momentum update are equal to 1, in order to mimic the behaviour of φ (10) in GHMC we set $\bar{\alpha} = 0.999$. Therefore, we obtain

$$\varphi(h) = -\ln 0.999 \frac{\mathcal{K}(h)}{D},$$

where

$$\mathcal{K}(h) = \frac{1 + 2h^2 \lambda_3(h)}{2h^4 \lambda_3(h)^2}. \tag{11}$$

Moreover, from (4), $\varphi(h) \in (0, 1]$, which yields

$$\varphi_{\text{opt}}(h) = \min \left\{ 1, -\ln 0.999 \frac{\mathcal{K}(h)}{D} \right\}. \tag{12}$$

We notice that $\mathcal{K}(h)$ is a monotonically decreasing function of h in the interval $(h_{\text{lower}}, h_{\text{CoLSI}})$ (Fig. 4), and does not depend on a simulated model. Hence, we can choose the bounds of the randomization interval for optimal φ as

$$\varphi_{\text{lower}} = \varphi_{\text{opt}}(h_{\text{CoLSI}}), \quad \varphi_{\text{upper}} = \varphi_{\text{opt}}(h_{\text{lower}}). \tag{13}$$

Thus, an optimal randomization interval for the random noise φ in the PMU (4) reads as

$$\varphi \sim \mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}}). \tag{14}$$

Optionally, one can adapt φ_{opt} (12) in each Monte Carlo iteration for h within the randomization interval (8) and its corresponding parameters of s-AIA3. In this case, no additional randomization of φ is required.

In what follows, we first describe a procedure for an optimal choice of a number of integration steps per Monte Carlo iteration, L , and then compare our proposed scheme (14) combined with the various choices of L with the commonly recommended $\varphi \sim \mathcal{U}(0, 0.5)$ and two additional randomization intervals, $\varphi \sim \mathcal{U}(0, 0.1)$ (close to Metropolized Molecular Dynamics: $\varphi = 0$) and $\varphi \sim \mathcal{U}(0, 0.9)$ (close to HMC: $\varphi = 1$).

6. Optimising Hamiltonian trajectory lengths

A PMU (4) allows for maintaining a favourable direction of sampling, provided (i) the acceptance rate for proposed Hamiltonian trajectories is high and (ii) momentum mixing angle φ at the PMU is small. Under these conditions, choosing the smallest possible L , i.e. $L = 1$ helps to explore in full the area containing a direction once accepted by the Metropolis test, thus mimicking a long but flexible trajectory. So, in contrast to HMC, GHMC may benefit from short Hamiltonian trajectories.

The optimal settings proposed in this study tend to satisfy both conditions, (i) and (ii). Indeed, the randomization interval for optimal step size h_{CoLSI} and the integrator s-AIA3 are chosen to guarantee small expected energy errors (see Fig. 2) and hence, high acceptance rates. Furthermore, as follows from (12), $\varphi \sim 1/D$, where D is the dimension of a simulated system. Thus, the reasonable suggestion for an optimal choice of L might be $L_{\text{opt}} = 1$, provided that the underlying analysis for finding an optimal integrator and a step size is accurate.

However, as was indicated in [17], the accuracy of the approach for estimating h_{CoLSI} may suffer when it is applied to systems with clear anharmonic behaviour. Such behaviour can be caught by inspecting a system-specific value of a fitting factor S_f , computed using the simulation data collected at a burn-in stage of a GHMC simulation as

$$S_f = \begin{cases} S_\omega, & \text{if the full set of } \omega_i, i = 1, \dots, D \text{ is available,} \\ S, & \text{otherwise,} \end{cases} \tag{15}$$

where

$$S_\omega = \max \left(1, \frac{1}{\Delta t_{\text{VV}}} \sqrt[6]{\frac{32 E_{\text{VV}}[\Delta H]}{\sum_{j=1}^D \omega_j^6}} \right), \quad S = \max \left(1, \frac{1}{\tilde{\omega} \Delta t_{\text{VV}}} \sqrt[6]{\frac{32 E_{\text{VV}}[\Delta H]}{D}} \right).$$

Here, $\omega_i, i = 1, \dots, D$ are the angular frequencies of the system ($\tilde{\omega}$ is the highest), Δt_{VV} is the step size used during the burn-in stage performed with the VV integrator, and $E_{\text{VV}}[\Delta H]$ is the expectation of the energy error due to the integration with VV. Further details on fitting factors are provided in [17] and [39], Appendix C.

Its clear deviation from 1 implies the deviation from harmonic behaviour. As a result, for such systems, the expected energy error at the estimated h_{CoLSI} may differ from the one predicted by the analysis of a harmonic oscillator. Moreover, using the expected energy error for 1-stage integrator at $L = 1$ for the analysis of 3-stage integrators in [17] leads to an overestimation of h_{CoLSI} , as discussed in [39], Appendix A. Even though, in the settings proposed in this study, it is not only possible to quantify the error introduced by the suggested analysis but also to compensate for it by choosing parameter L in the appropriate way.

In particular, let $E_{S_f=1}^{L=1}[\Delta H]$ be the expected energy error for 1-stage Velocity Verlet derived at $L = 1$ for systems with harmonic behaviour ($S_f = 1$) and $E_{S_f>1}^L[\Delta H]$ be the expected energy error for 1-stage Velocity Verlet at an arbitrary $L \geq 1$ for anharmonic systems ($S_f > 1$). As shown in [17],

$$E_{S_f=1}^{L=1}[\Delta H] = \frac{h^6}{32}, \quad h = S_f \tilde{\omega} \Delta t.$$

We denote

$$X = \frac{E_{S_f>1}^L[\Delta H]}{E_{S_f=1}^{L=1}[\Delta H]}.$$

By setting $X = 1$ and solving the equation for $L \geq 1$, one can identify values of L that level the difference between the expected errors in a GHMC simulation for harmonic and anharmonic systems. In [39], Appendix E, we follow this idea and obtain such values of L as a function of S_f :

$$L_{\text{opt}} = \begin{cases} 1, & \text{for } 1.0 \leq S_f < 1.5, \\ 2, 5, 7, & \text{otherwise.} \end{cases} \tag{16}$$

Finally, taking the suggestion in [22], we randomize L by selecting the optimal values in (16) with equal probability. The resulting randomization scheme is as follows:

$$L = \begin{cases} 1, & \text{for } 1.0 \leq S_f < 1.5, \\ \mathcal{U}\{2, 5, 7\}, & \text{otherwise.} \end{cases} \tag{17}$$

7. Adaptive Tuning algorithm

Here, we summarize the Adaptive Tuning algorithm (ATune) that, given a simulation system, generates a set of optimal parameters for a GHMC/HMC simulation to give rise to Adaptively Tuned GHMC/HMC (AT-GHMC/HMC).

Similarly to the s-AIA algorithm presented in [17], ATune relies on the simulation data collected during the GHMC/HMC burn-in stage. The latter is run with 1-stage Velocity Verlet using $L = 1$ and a simulation step size Δt_{VV} tuned on the fly, in order to reach a target acceptance rate (AR) for the burn-in (more details in Appendix B in [17]). Finally, the random noise φ for the PMU (4) to run the GHMC burn-in stage is selected according to (13)–(14).

Table 4

List of benchmarks used in this study. \mathcal{W} is the Wishart distribution (more details in [12]), \mathbb{I}_D is the identity matrix in $\mathbb{R}^{D \times D}$, $\mathbf{y} \in \mathbb{R}^K$ is the set of observations.

Benchmark	Model	Data	D (dimension)	K (observations)
G1000			1000	-
G500	$\mathcal{N}(0, \Sigma)$	$\Sigma^{-1} \sim \mathcal{W}(\mathbb{I}_D, D)$ [12]	500	-
G2000			2000	-
German	Bayesian Logistic Regression [52]	[53]	25	1000
Musk			167	476
Banana [50]	$\mathcal{N}(0, \sigma_\theta^2)$	$\sigma_\theta^2 = 1, \mathbf{y} \sim \mathcal{N}(1, 2)$	2	100

On the completion of the GHMC/HMC burn-in stage, the data needed for calculation of the fitting factor S_f (15) of the simulated system are computed and stored:

- AR
- maximum frequency $\tilde{\omega}$
- (optional) frequencies, $\omega_i, i = 1, \dots, D$ and their standard deviation σ .

S_f is critical for the nondimensionalization procedure as well as for finding optimal integration coefficients for s-AIA3 integrators and optimal parameter settings for a GHMC/HMC production simulation, as described in [17] and summarized in [39], Appendix C. If the full set of frequencies $\omega_i, i = 1, \dots, D, \tilde{\omega}$ and σ are available during the burn-in stage (which is optional and requires an extra computational effort due to the calculation of Hessians, see [17]), then $S_f = S_\omega$ (see (15)). Otherwise, a cheaper option of S_f , i.e. $S_f = S$ (15), which only needs $\tilde{\omega}$, is chosen. Then, the parameters of a s-AIA3 integrator are computed according to the procedure introduced in [17] with the use of the pretabulated map $h \rightarrow b_{\text{opt}}$ available in [48].

The GHMC/HMC optimal parameters for the production stage are:

- **Step size Δt**

Δt is picked uniformly randomly from $(\Delta t_{\text{lower}}, \Delta t_{\text{CoLSI}})$, where

$$\Delta t_{\text{lower}} = \frac{h_{\text{lower}}}{\text{CF}}, \quad \Delta t_{\text{CoLSI}} = \frac{3}{\text{CF}},$$

h_{lower} is defined in (8) and

$$\text{CF} = \begin{cases} S_f(\tilde{\omega} - \sigma), & \text{if } \sigma > 1, \\ S_f \tilde{\omega}, & \text{otherwise.} \end{cases} \tag{18}$$

See Appendix C in [39] for details.

- **PMU random noise φ** (for GHMC only)
As at the burn-in stage, φ is randomized following Eqs. (13)–(14).
- **Number of integration steps per Monte Carlo iteration L** (recommended for GHMC but can be used with HMC)
 L is chosen following (17).

Finally, the production stage is run using s-AIA3 integrator [17] and the set of optimal parameters for a chosen number of iterations N_{prod} .

The ATune algorithm is outlined in Fig. 5. The algorithm is implemented (with no computational overheads introduced in a production simulation) in the BCAM in-house software package HaiCS (Hamiltonians in Computational Statistics), designed for statistical sampling of high dimensional and complex distributions and parameter estimation in Bayesian models using MCMC and HMC-based methods.

The package provides analytical gradients for popular statistical models, including all models but one considered in this study. In addition, finite difference approximation of gradients is supported in HaiCS.

A comprehensive overview of the package can be found in [49], whereas applications of HaiCS software are presented in [17,47,50,51].

8. Numerical experiments

The next objective is to numerically justify the efficiency of our new ATune algorithm using a set of standard benchmarks summarized in Table 4.

8.1. Performance metrics

The irreversibility of GHMC may allow it to reach convergence faster, compared to reversible samplers, like HMC. To check this, we choose the performance metric that includes information about the speed of convergence of the Markov chains. The metric is

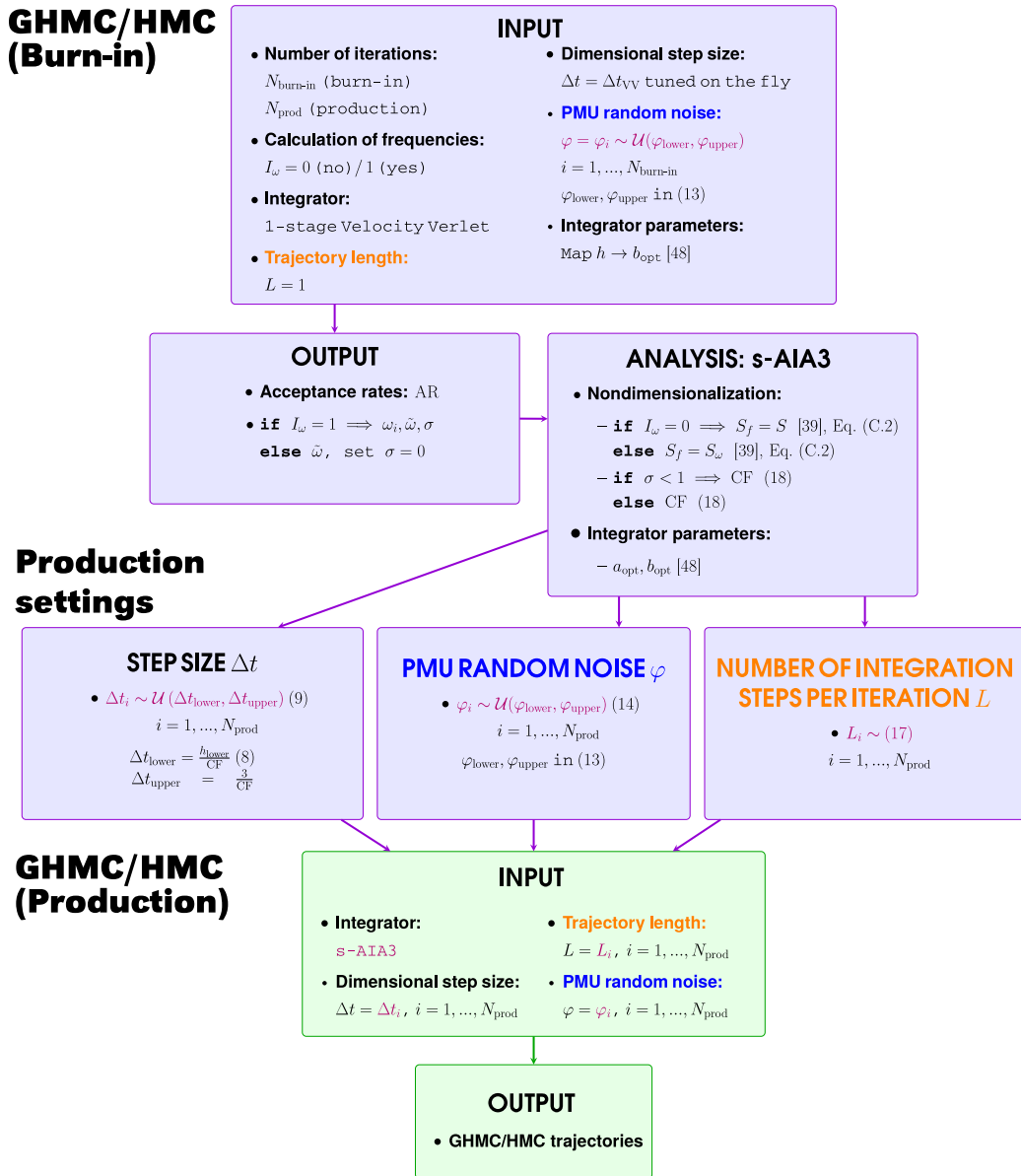


Fig. 5. Schematic representation of the ATune algorithm for generating a set of optimal parameters for a GHMC/HMC simulation. The features specific to AT-GHMC only are highlighted in blue, whereas the features optional for AT-HMC are displayed in orange. Optimal parameters are shown in magenta.

briefly introduced in Section 3. Here we provide more details on the metric and extend it to the alternative approaches for computing effective sample size (ESS).

Following the recommendation in [45], we consider a chain to be converged if $\max \text{PSRF} < 1.01$, where PSRF is the Potential Scale Reduction Factor [54], calculated as described in [55].

Starting from that, let us define the number of samples in the production chains, required for reaching $\max \text{PSRF} < 1.01$, as $N_{1.01}$, and evaluate the ratio grad/ESS between the number of gradient computations and ESS for production chains of the length of $N_{1.01} + 1000$ iterations. Such a metric combines both the information about convergence and sampling efficiency.

The number of gradient evaluations reads as

$$\text{grad} = (N_{1.01} + 1000) \bar{L} k,$$

where \bar{L} is the mean of the number of integration steps per Monte Carlo iteration (it varies depending on a chosen randomization scheme for L) and k is the number of stages of an integrator in use (in our case, $k = 3$). We recall that the number of stages k indicates

Table 5
Optimal randomization intervals for L (17) and φ (14) for each tested benchmark – G1000, G500, G2000, German, Musk, Banana.

Benchmark	L	φ
G1000	1	$\mathcal{U}(0.00044, 0.00264)$
G500	1	$\mathcal{U}(0.00088, 0.00527)$
G2000	1	$\mathcal{U}(0.00022, 0.00132)$
German	1	$\mathcal{U}(0.01752, 0.10545)$
Musk	$\mathcal{U}\{2, 5, 7\}$	$\mathcal{U}(0.00262, 0.01579)$
Banana	$\mathcal{U}\{2, 5, 7\}$	$\mathcal{U}(0.21904, 1.00000)$

the number of times the algorithm performs an evaluation of gradients per step size. Therefore, $\bar{L}k$ represents the theoretical average number of gradient evaluations per Monte Carlo iteration.

In Section 3, we calculate ESS using the `effectiveSize` function of the `coda` package of R [46] and consider the minimum ESS across variates, i.e. `minESS`. We want to include two additional ESS metrics in our list, which now appears like:

- `minESS (coda)`
- `meanESS (coda)` – the average ESS over the sampled variables calculated through the `effectiveSize` function of the `coda` package of R [46]
- `multiESS` – the metric calculated in the `multiESS` function of the `mcmcse` package [56] using the approach proposed in [57].

To summarize, we propose three metrics for assessing convergence and sampling performance, such as `grad/minESS`, `grad/meanESS`, and `grad/multiESS`. For each metric, a lower gradient value (`grad`) indicates better convergence, while a higher ESS suggests improved sampling efficiency. Consequently, the smallest `grad/ESS` ratio represents the best overall performance.

To enable comparison of efficiency between two samplers, a Relative Efficiency Factor (REF) is introduced as:

$$\text{REF}(\text{Sampler}_1, \text{Sampler}_2) = \frac{\mathcal{M}_2}{\mathcal{M}_1}, \tag{19}$$

$$\mathcal{M} = \frac{\text{grad}}{\text{ESS}}, \quad \text{ESS} \in \{\text{minESS}, \text{meanESS}, \text{multiESS}\}.$$

The REF value quantifies the extent to which `Sampler1` outperforms `Sampler2` for a particular $\mathcal{M} = \text{grad/ESS}$ metric, where ESS is picked from the set `{minESS, meanESS, multiESS}`.

8.2. Benchmark tests

We use the metrics described in 8.1 for the full set of simulation benchmarks (Table 4). For the biggest benchmark (G2000), we consider the ESS metrics evaluated over the first $N_{1,01} + 2000$ iterations, in order to take into account enough iterations ($> D = 2000$) for the `multiESS` calculation. We also remark that for Gaussian benchmarks the `grad/ESS` metrics for the simulations with $L = 1$ were calculated over the first $\tilde{N}_{1,01} + 1000$ (2000 for G2000), where $\tilde{N}_{1,01}$ is the number of iterations required to achieve `avgPSRF` < 1.01 . The relaxation of the PSRF threshold helped to avoid extremely long simulations ($N_{\text{prod}} \gg 10^6$) needed for reaching convergence in terms of `max PSRF` < 1.01 . Numerical experiments for all benchmarks considered here, as well as the case studies in Section 9, were performed using 10 chains, except for Banana, which was run with 100 chains.

In order to evaluate the optimal settings proposed in this study, we compare our recommended choice $\varphi \sim \mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$ (14) with the fixed choice of $\varphi = 1$ (which corresponds to standard HMC) and with three randomization intervals $\varphi \sim \mathcal{U}(0, 0.1)$, $\mathcal{U}(0, 0.5)$, $\mathcal{U}(0, 0.9)$. For each choice of φ , in addition to the proposed optimal randomization interval for L (17), we consider five different $\mathcal{U}\{1, L_{\text{upper}}\}$ intervals (four for Banana), where the values of L_{upper} are proportional to the dimension of the simulated system. The optimal randomization intervals for φ and L are summarized in Table 5.

As put forward at the end of Section 4.2, we use `s-AIA3` as a numerical integrator for Hamiltonian dynamics and the step size randomization interval proposed in Section 4.1 (Eq. (9)). With respect to the fitting factor approach (15), all benchmarks tested here use the more accurate \mathcal{S}_ω (see Appendix C in [39] and [17] for further details). Finally, in these and subsequent numerical experiments, eigenvalues are computed via a full dense eigendecomposition using the `dsyev` routine from the LAPACK package [58].

Table 6 presents the performance comparison between GHMC with $\varphi \sim \mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$ and standard HMC ($\varphi = 1$) across the tested benchmarks, with varying values of L_{upper} . For any benchmark and for each metric, the best performance is consistently achieved by GHMC (shown in red). Combining GHMC with our proposed choice of L_{opt} leads to the best GHMC results for the average metrics, i.e. `grad/meanESS` and `grad/multiESS`. Moreover, for non-Gaussian benchmarks (also, with smaller dimensions), `grad/minESS` shows the best results with such a choice of L . For Gaussian benchmarks, the best results for `grad/minESS` are reached for larger values of L_{upper} – they are lower for GHMC (around D) and higher for HMC ($> 2D$).

The results obtained with other tested randomization intervals for φ , $\mathcal{U}(0, 0.1)$, $\mathcal{U}(0, 0.5)$, $\mathcal{U}(0, 0.9)$, support our choice for the optimal randomization interval ($\varphi_{\text{lower}}, \varphi_{\text{upper}}$) and can be examined in [39], Appendix F.

Finally, in Fig. 6, we provide performance comparison in terms of the REF (19) between GHMC with our optimal settings, i.e. AT-GHMC, and HMC with its best settings found heuristically (see Table 6) using all proposed ESS metrics and all tested benchmarks.

Table 6

Performance comparison between GHMC with optimal randomization scheme $\varphi \in \mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$ (14) and standard HMC, in terms of grad/ESS metrics (ESS $\in \{\text{minESS, meanESS, multiESS}\}$) for the range of randomization schemes for L in all the tested benchmarks. The overall top values for each metric in each benchmark are highlighted in red. The best values for the sampler, but not overall the best for each metric, are shown in bold. The results achieved with the most successful simulation settings for the most successful sampler are shaded in light blue, whereas the results obtained with the most successful simulation settings for less successful sampler are highlighted in light grey.

Benchmark	φ	ESS	$L = 1$	L_{opt}	$\mathcal{U}\{1, \frac{D}{3}\}$	$\mathcal{U}\{1, \frac{2}{3}D\}$	$\mathcal{U}\{1, \frac{4}{3}D\}$	$\mathcal{U}\{1, \frac{8}{3}D\}$
G1000	$\mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$	minESS	39373.1	2387.2	1491.1	2509.4	5021.0	
		meanESS	0.5	532.1	1071.4	2142.4	4285.3	
		multiESS	0.4	299.8	402.5	895.4	3069.2	
	1 (HMC)	minESS	132650.2	46712.1	21620.1	11740.0	5554.2	
		meanESS	2.5	533.6	1054.3	2093.2	4198.1	
		multiESS	3.7	301.3	1001.0	2000.0	4001.0	
G500	$\mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$	minESS	31792.2	1553.4	919.8	1090.2	2339.1	
		meanESS	0.3	260.0	520.7	1043.2	2093.1	
		multiESS	0.3	170.6	320.8	384.0	1422.3	
	1 (HMC)	minESS	116471.2	42111.1	21821.2	10094.4	4790.2	
		meanESS	2.4	261.3	515.5	1035.3	2046.0	
		multiESS	3.7	107.8	321.6	563.3	1325.2	
G2000	$\mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$	minESS	63451.2	3305.0	2515.3	5036.4	9195.1	
		meanESS	0.7	1104.1	2207.4	4417.2	8853.4	
		multiESS	0.5	428.5	860.1	1723.0	7096.3	
	1 (HMC)	minESS	174435.1	47859.0	22770.3	10870.1	9582.1	
		meanESS	2.8	1099.1	2178.4	4348.1	8671.2	
		multiESS	3.7	677.3	2000.0	4001.0	8000.0	
German	$\mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$	minESS	1.7	13.9	25.9	53.4	113.6	
		meanESS	0.3	10.2	23.8	49.1	101.0	
		multiESS	0.1	10.6	25.1	49.5	100.5	
	1 (HMC)	minESS	24.9	13.3	26.2	53.1	112.6	
		meanESS	5.9	10.6	24.5	49.2	102.9	
		multiESS	2.3	10.9	25.4	49.4	100.8	
Musk	$\mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$	minESS	216.8	143.7	229.4	389.7	601.3	1458.3
		meanESS	145.4	52.1	68.4	165.0	464.5	1031.2
		multiESS	19.8	12.8	36.0	160.4	342.0	774.5
	1 (HMC)	minESS	6026.1	1156.4	187.7	290.3	525.7	1215.2
		meanESS	3068.0	705.6	143.5	176.3	454.6	1047.4
		multiESS	375.8	73.0	50.6	159.5	279.5	506.9
Banana	$\mathcal{U}(\varphi_{\text{lower}}, \varphi_{\text{upper}})$	minESS	$L = 1$	$U\{1, D\}$	L_{opt}	$U\{1, 5D\}$	$U\{1, 6D\}$	
		meanESS	283.9	245.2	160.2	168.0	184.1	
		multiESS	185.9	164.6	149.0	167.3	176.9	
	1 (HMC)	minESS	192.2	176.3	166.1	182.8	183.0	
		meanESS	587.7	426.7	229.8	210.0	206.9	
		multiESS	392.6	275.0	191.9	177.9	199.6	
			421.4	292.3	214.9	198.7	218.7	

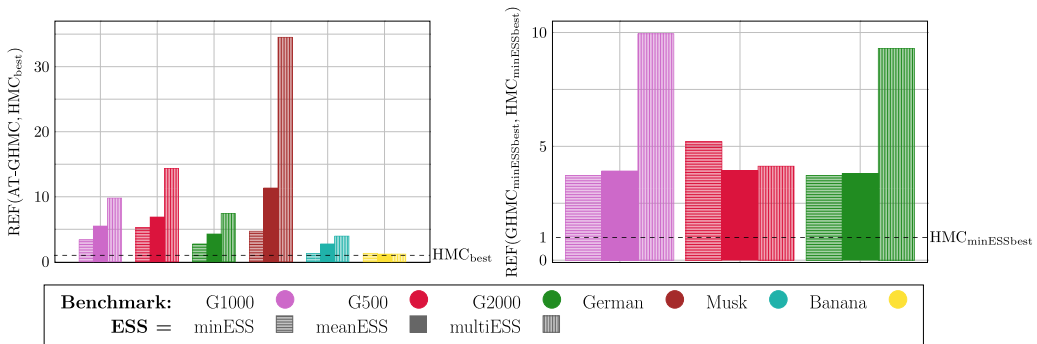


Fig. 6. On the left: Relative efficiency (REF (19)) of GHMC with the optimal parameter settings, AT-GHMC (light blue boxes in Table 6) with respect to HMC with its top parameter setting, HMC_{best} (light grey boxes in Table 6) among all the tested benchmarks. All the benchmarks show the superiority (up to 34x) of AT-GHMC over HMC_{best} with each performance metric. On the right: Relative efficiency (REF (19)) of GHMC with respect to HMC with their best settings for minESS (see Table 6) for the Gaussian benchmarks – G1000, G500, G2000. All the benchmarks show the superiority (up to 10x) of $GHMC_{\text{minESSbest}}$ over $HMC_{\text{minESSbest}}$ with each performance metric.

Table 7
Parameter settings and AR for the contour plots in Fig. 7.

Sampler	Δt	L	φ	AR
AT-GHMC	$\mathcal{U}(0.152, 0.219)$	$\mathcal{U}(2, 8)$	$\mathcal{U}(0.21904, 1.00000)$	0.92
HMC tuned	0.111	7	1.00000	0.69
HMC	0.050	2	1.00000	0.91
	0.500			0.02

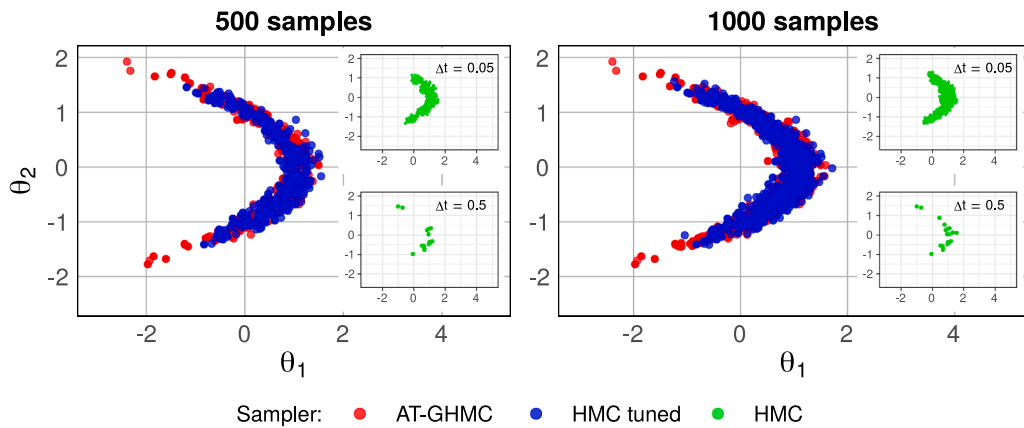


Fig. 7. First 50 and 100 samples from 10 independent chains obtained with AT-GHMC (red), tuned HMC (blue), and standard HMC (green) with two choices of Δt .

For the Gaussian benchmarks, we also present the comparison at the L_{upper} values that optimize the minESS metric in HMC and GHMC (see Table 6). Both plots in Fig. 6 confirm the improvements (up to 34x and 10x, respectively) gained with our proposed GHMC settings across all metrics and benchmarks. They also demonstrate that greater improvements are achieved with the average metrics, i.e. meanESS and multiESS.

Last of all, we investigate the ability of AT-GHMC to explore the phase space in comparison with both tuned and standard HMC. We consider the Banana benchmark and compare AT-GHMC against the manually tuned HMC of [59], as well as standard HMC with $L = D$ and two choices of the step size: $\Delta t = 0.05$, comparable with the value identified by AT-GHMC, and $\Delta t = D^{-1}$. The parameter settings and AR are reported in Table 7.

Fig. 7 illustrates the first 50 and 100 samples from 10 independent chains for each sampler. The results clearly show that AT-GHMC explores a broader region of the phase space than tuned HMC. Moreover, both AT-GHMC and tuned HMC substantially outperform untuned HMC, highlighting the critical role of hyperparameter selection in achieving efficient sampling. Finally, we emphasize that the tuning strategy employed in [59] relies on a manual, trial-and-error procedure, whereas AT-GHMC leverages the fully automated ATune approach.

9. Applications – Case studies

Next, we apply ATune to three case studies:

- Patient resistance to endocrine therapy in breast cancer
- Cell-cell adhesion dynamics
- Influenza A (H1N1) epidemics outbreak.

9.1. Patient resistance to endocrine therapy in breast cancer

Estrogen-positive (ER+) breast cancer is the most common subtype of breast cancer, accounting for over 70% of all breast tumours diagnosed. Hormone therapies such as tamoxifen are the gold-standard treatment. However, a significant amount of patients develops resistance to the treatment, which constitutes a serious clinical problem. The development of this resistance is not yet well understood, and plenty of clinical and laboratory research aims to find which genes may be relevant to the problem.

For one such clinical study presented in [60], ER+ breast cancer patients treated with tamoxifen for 5 years were tested for the presence of a specific gene marker, SOX2. The objective of the study was to analyse the connection between SOX2 and the emergence of resistance to tamoxifen treatment. An overexpression of this gene was previously identified in bioinformatic analyses of resistant breast cancer cells as a key contributor for resistance, but clinical evidence was presented in [60] for the first time.

The dataset used in our work stems from that clinical trial. It contains numerical data for SOX2 values and 7 other gene-related covariates taken from 75 tumours samples. The recorded response variable was the event of relapse after 5 years. Notably, the SOX2

Table 8

Simulation parameters for the numerical experiments on the breast cancer dataset with the BLR model combined with three different priors – $\mathcal{N}(0, 5)$, $\mathcal{N}(0, 2.5)$, $\mathcal{N}(0, 1)$. The optimal settings for AT-GHMC and AT-HMC, as well as the optimal Δt randomization interval for $HMC_{L_{NUTS}}$ are found using the procedure described in Section 7. AT-GHMC, AT-HMC, and $HMC_{L_{NUTS}}$ employ s-AIA3 integrator while NUTS uses standard Verlet. The mean values of L and Δt used by NUTS optimal routine are denoted as \bar{L} and $\bar{\Delta t}$, respectively.

Prior	Sampler	$N_{1,01}$	L	Δt	φ
$\mathcal{N}(0, 5)$	AT-GHMC	300	1	$\mathcal{U}\{0.889, 1.284\}$	$\mathcal{U}\{0.055, 0.330\}$
	AT-HMC	200	1	$\mathcal{U}\{0.890, 1.285\}$	-
	$HMC_{L_{NUTS}}$	100	$\mathcal{U}\{1, 6\}$	-	-
	NUTS	100	$\bar{L} = 15$	$\bar{\Delta t} = 0.247$	-
$\mathcal{N}(0, 2.5)$	AT-GHMC	100	1	$\mathcal{U}\{0.725, 1.047\}$	$\mathcal{U}\{0.055, 0.330\}$
	AT-HMC	100	1	-	-
	$HMC_{L_{NUTS}}$	200	$\mathcal{U}\{1, 8\}$	$\mathcal{U}\{0.723, 1.045\}$	-
	NUTS	100	$\bar{L} = 12$	$\bar{\Delta t} = 0.354$	-
$\mathcal{N}(0, 1)$	AT-GHMC	100	1	$\mathcal{U}\{0.545, 0.787\}$	$\mathcal{U}\{0.055, 0.330\}$
	AT-HMC	200	1	-	-
	$HMC_{L_{NUTS}}$	200	$\mathcal{U}\{1, 11\}$	$\mathcal{U}\{0.545, 0.788\}$	-
	NUTS	100	$\bar{L} = 8$	$\bar{\Delta t} = 0.474$	-

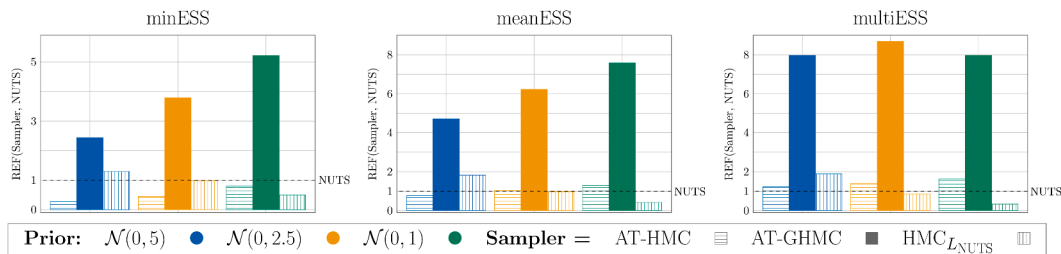


Fig. 8. Relative efficiency REF (19) for minESS (left), meanESS (center), and multiESS (right) of AT-GHMC, AT-HMC, and $HMC_{L_{NUTS}}$ with respect to Stan NUTS obtained in the numerical experiments run with the settings presented in Table 8. The performance of AT-GHMC (filled bars), AT-HMC (horizontal bars), and $HMC_{L_{NUTS}}$ (vertical bars) are compared with NUTS for the three examined priors: $\mathcal{N}(0, 5)$ (blue), $\mathcal{N}(0, 2.5)$ (yellow), $\mathcal{N}(0, 1)$ (green). AT-GHMC significantly outperforms NUTS for any metric and prior. AT-HMC demonstrates good results for average metrics (meanESS, multiESS) with all priors but it is not convincing for minESS. Using L_{NUTS} instead of L_{opt} (17) in HMC leads to performance degradation in HMC simulations with $\mathcal{N}(0, 1)$, whereas for $\mathcal{N}(0, 5)$ the results are visibly better than with NUTS HMC and with AT-HMC for all metrics.

variable perfectly separates the dataset for this response variable, as all cases with SOX2 values below 3 respond well to the treatment and all those above this value present a resistant response. The appearance of separation implies that maximum likelihood estimates do not exist for some binary classification models such as logistic regression [61] in the classical frequentist interpretation. The perfect separation makes the Bayesian framework natural to model such a dataset, as it can provide estimates for the parameters involved without the need to use regularization techniques.

The model of choice is a Bayesian Logistic Regression, where the dataset has 75 observations and dimension $D = 8$ (including intercept parameter). We test the effect of three priors, informative $\mathcal{N}(0, 1)$, weakly informative $\mathcal{N}(0, 2.5)$ (according to [62]) and low informative $\mathcal{N}(0, 5)$.

We compare the performance of AT-GHMC proposed in this study against the No-U-Turn-Sampler (NUTS) [12], the optimized HMC sampler implemented in Stan [63] using its R interface rstan [64]. NUTS in Stan uses the leapfrog/Verlet integrator [13] and the original optimization routine to determine both a step size, Δt , and a number of Hamiltonian dynamics integration steps per iterations, L . Additionally, we perform numerical experiments with HMC using L_{opt} from Eq. (17) (AT-HMC) and setting $L = L_{NUTS}$ ($HMC_{L_{NUTS}}$). The latter is determined such that the integration leg $\Delta t L$ matches that found by NUTS. AT-GHMC, AT-HMC and $HMC_{L_{NUTS}}$ use s-AIA3 and the more accurate S_{ω} fitting factor approach (15) (see Appendix C in [39] for further details).

All samplers are run with the same number of burn-in iterations, $N_{burn-in}$, to ensure a fair comparison. We assess performance using grad/ESS metrics introduced in Section 8.1. Small values of $N_{1,01}$ in NUTS are expected, given its optimization routine aims to bring PSRF close to 1. The simulation settings are summarized in Table 8.

Fig. 8 compares the sampling performance of AT-GHMC, AT-HMC, and $HMC_{L_{NUTS}}$ with NUTS using REF metrics (19) for ESS \in {minESS, meanESS, multiESS}. AT-GHMC consistently outperforms NUTS across all metrics and priors, achieving more than 8 times improvement (e.g., multiESS for the $\mathcal{N}(0, 2.5)$ prior).

We emphasize that the ATune algorithm does not offer an optimal choice of L for HMC. The values of L_{opt} (17) used in AT-HMC are the recommended values for GHMC. However, the strong meanESS and multiESS performance of AT-HMC in all experiments confirms

their applicability in HMC. On the other hand, the relative efficiency of $HMC_{L_{NUTS}}$ with respect to NUTS shows its strong dependence on a chosen prior. Specifically, $HMC_{L_{NUTS}}$ demonstrates visible improvements over NUTS with the low informative prior $\mathcal{N}(0, 5)$; the comparable performance with NUTS if the chosen prior is $\mathcal{N}(0, 2.5)$, and the poor sampling efficiency with the highly informative prior $\mathcal{N}(0, 1)$. For $\mathcal{N}(0, 5)$, the improved performance can be attributed to the higher accuracy and longer optimal step size of the 3-stage integrator (s-AIA3) employed in $HMC_{L_{NUTS}}$ compared to those achievable with the standard 1-stage Verlet integrator used in NUTS. On the contrary, the poor performance of $HMC_{L_{NUTS}}$ with $\mathcal{N}(0, 1)$ is likely caused by the underestimation of Δt_{ColSI} hinted by the comparison of its value with $\bar{\Delta t}$ (see Table 8). Indeed, the values are close, which is not what one expects when comparing optimal step sizes for 1- and 3- stage integrators. It is also not the case for $\mathcal{N}(0, 5)$ and $(0, 2.5)$, where optimal step sizes are longer than the NUTS optimal step sizes (less pronounced for $\mathcal{N}(0, 2.5)$ though), and where AT-GHMC and AT-HMC have never been outperformed by NUTS. These observations are encouraging. They mean that even in the worst scenario, when the estimated optimal parameters are less accurate, they still guarantee the good sampling performance (see AT-GHMC and AT-HMC for $\mathcal{N}(0, 1)$) and are not inferior to the optimal parameters offered by the state-of-the-art NUTS approach. Moreover, the sampling advantages of GHMC over HMC, enhanced by the stochastic procedure for finding its optimal settings, disregard the inaccuracy in the estimation of an optimal step size and, as a result, AT-GHMC demonstrates a significant improvement over NUTS, AT-HMC and $HMC_{L_{NUTS}}$ as happened in the case of $\mathcal{N}(0, 1)$. The fact that the performance of AT-HMC in this case is comparable with, or better than the NUTS performance also confirms the robustness of the proposed methodology.

9.2. Parameter inference on a cell-cell adhesion model

Many phenomena on the cellular level can be explained by introducing differential cell-cell adhesion [65–68]. This approach has been used, e.g., to describe pattern formation for zebrafish lateral lines, cancer invasion modeling and cell sorting in early fly brain and early mice brain development (see [69] and references therein). When modeling such biological systems using PDEs, the adhesion can be represented by means of a non-local interaction potential [65,67,68,70]. To enhance predictive power of such models, a careful calibration of the model parameters according to experimental lab data is required. This task can become increasingly difficult due to sparsity of available data or increasing complexity of the model (such as, for example, by the addition of an interaction term). The application of state-of-the-art methodologies for performing parameter estimation on PDE models is receiving lots of attention in the last years [71,72], and, in this context, our goal is to investigate the feasibility of using the methodology introduced in this work for the task.

To achieve this goal, we propose a benchmark PDE model:

$$\frac{\partial u}{\partial t} = \nabla \cdot [u \nabla (\Pi'(u) + (W * u))], \tag{20}$$

which describes the time evolution of the density of cells $u(\mathbf{x}, t) > 0$, $\mathbf{x} \in \mathbb{R}^d$, $t > 0$ ($d = 1, 2, 3$ is the spatial dimension), as governed by a series of potentials. In (20), $\Pi(u)$ is a density of the internal energy and $W(\mathbf{x})$ is an interaction potential (also called an interaction kernel) describing cell adhesion. The convolution $(W * u)$ gives rise to non-local interactions. Depending on the explicit form of the potentials, they can describe a wide range of biological phenomena such as the formation of ligands through long filopodia, modeled by nonlocal attractive potentials, and volume constraints, modeled by strong repulsive nonlocal potentials or nonlinear diffusive terms. We refer to [73] for more information about the applications of these aggregation-diffusion models in the sciences. We consider the following explicit forms for $\Pi'(u)$ and $W(\mathbf{x})$:

$$\begin{aligned} \Pi'(u) &= \nu u^{m-1} \\ W(\mathbf{x}) &= -a e^{-|\mathbf{x}|^2/2r_W} / \sqrt{2\pi r_W}, \end{aligned} \tag{21}$$

where $\Pi'(u)$ gives rise to a diffusion term, governed by the diffusion exponent m . In the limit $m \rightarrow 1$, one recovers linear diffusion, whereas if $m > 1$ the term represents porous medium and leads to finite speed propagation fronts. The diffusion coefficient ν defines the cell mobility in the medium. On the other hand, the explicit form of $W(\mathbf{x})$ leads to symmetric attraction between cells due to ligands formation on the cell’s membrane. The strength of the attraction is given by the parameter $a > 0$, whereas $r_W > 0$ controls the range of the interaction.

Our purpose is to perform Bayesian inference of the parameters of this model using three different MCMC methods, and to evaluate the efficiency of each tested sampler for this class of models. Parameter inference in a Bayesian framework requires sampling from the posterior distribution of parameters given by $\pi(\theta|\mathbf{y}) \propto L(\mathbf{y}|\theta)p(\theta)$, where \mathbf{y} are the experimental data and θ is the vector of model parameters. For the proposed model, the data consists of a collection of measurements of the cell density at different spatial points and different times $\mathbf{y} = \{u_{\mathbf{y}}(\mathbf{x}_i, t_j)\}_{i,j}$. In cell modeling, it is usual to assume that the experimental data stems from the proposed model with additive Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma)$, i.e. $u_{\mathbf{y}}(\mathbf{x}_i, t_j) = u(\mathbf{x}_i, t_j) + \varepsilon$. Under this assumption, the likelihood takes the form:

$$L(\mathbf{y}|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\sum_j \left(\frac{u_{\mathbf{y}}(\mathbf{x}, t_j) - u(\mathbf{x}, t_j)}{\sigma}\right)^2\right). \tag{22}$$

In this work, we use synthetic data generated from model (20) in a one dimensional case ($d = 1$). First, we numerically solve the model in the interval $t \in [0, 5]$ by using the finite volume scheme proposed in [74] with a spatial meshing of 50 cells and a time step of $\delta t = 0.1$, resulting in a total of 51 time points. We denote the numerical solution at point x_i and time t_j as $\tilde{u}_s(x_i, t_j)$. We then generate the synthetic data by adding Gaussian white noise $\varepsilon_{ij} \sim \mathcal{N}(0, \sigma)$ to each cell and time point $\tilde{u}_{\mathbf{y}}(x_i, t_j) = \tilde{u}_s(x_i, t_j) + \varepsilon_{ij}$, where σ is the noise parameter.

Table 9

Simulation parameters for the numerical experiments on the cell-cell adhesion model. The optimal settings for AT-GHMC and AT-HMC are found using the procedure described in Section 7. For RW applied to the 4-parameter model, the convergence criterion used to determine $N_{1.01}$ is relaxed to $\max\text{PSRF} < 1.1$.

Model	Sampler	$N_{1.01}$	L	Δt	φ
3-parameter	AT-GHMC	46,300	1	$\mathcal{U}(0.0024, 0.0035)$	$\mathcal{U}(0.15, 0.88)$
	AT-HMC	288,900	-	-	-
	RW	1,370,500	-	-	-
4-parameter	AT-GHMC	47,000	1	$\mathcal{U}(0.0025, 0.0035)$	$\mathcal{U}(0.11, 0.66)$
	AT-HMC	471,400	-	-	-
	RW	718,000	-	-	-

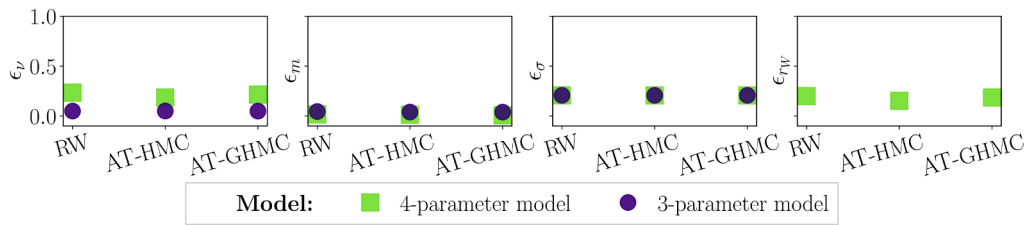


Fig. 9. Relative parameter of the diffusion coefficient (left), diffusion exponent (second to left), noise parameter (second to right) and range of interaction (right) for the 3-parameter model (purple circles) and 4-parameter model (green squares). All samplers are able to recover the true parameter value with low relative error, in particular AT-HMC and AT-GHMC show improved precision over RW.

We consider two different variations of the model for the numerical experiments. In the first one (3-parameter model) the parameters sampled are the diffusion coefficient ν , the diffusion exponent m from the internal energy potential in (21), and the model independent noise parameter σ . The remaining parameters in (21) are fixed to $a = 2$ and $r_W = 0.5$. In the second variation (4-parameter model), we extend the previous model to also consider r_W as a sampled parameter, leaving $a = 2$ as the only fixed parameter in (21). The chosen priors are $\nu \sim \text{Exp}(1)$, $m \sim \Gamma(10, 3)$, $\sigma \sim \text{Exp}(0.1)$, and $r_W \sim \text{Exp}(1)$.

Three sampling methodologies are examined: Random Walk Metropolis-Hastings (RW) (the most commonly used method for Bayesian inference), AT-HMC and AT-GHMC. In this case, we adopt the cheaper S fitting factor approach (15) (see Appendix C in [39] for details) to avoid the additional computational cost associated with Hessian evaluations in a PDE model. Gradients are estimated using finite differences.

We evaluate performance using the metrics introduced in Section 8.1. For this study, the metrics are computed over the first $N_{1.01} + 5000$ iterations, where $N_{1.01}$ is the number of iterations needed for convergence. We note that the slower-converging RW requires a large number of iterations to meet the strict threshold $\max\text{PSRF} < 1.01$ ($N_{1.01} > 10^6$). Therefore, for this sampler, we relax a convergence threshold to 1.1, as suggested in [54]. Since RW does not involve gradient evaluations (which constitute the bulk of the computational cost in HMC/GHMC), we normalize ESS metrics with respect to the simulation time T . Simulation settings are summarized in Table 9.

Fig. 9 shows the relative error for each parameter and both models, defined as $\epsilon_\theta = \frac{|\theta_{\text{true}} - \theta|}{\theta_{\text{true}}}$, where θ_{true} is the true parameter value used for generating the synthetic data and θ is the mean parameter value obtained from the RW, AT-HMC and AT-GHMC simulations considering the first 5000 iterations after convergence. The three samplers are able to recover the true parameter values with high precision, as evidenced by the low relative errors, meaning all parameter can be confidently identified with small variance. In particular, AT-GHMC and AT-HMC exhibit higher precision than RW. Nevertheless, the number of iterations needed to reach the stationary distribution are vastly different across samplers: AT-GHMC needs $\sim 10^4$ samples to converge for both models, whilst AT-HMC requires $\sim 10^5$ and RW $\sim 10^6$ (even with the relaxed convergence threshold), confirming that AT-GHMC has a much faster convergence rate.

In Fig. 10, we present the comparison of the REF metric (19) for the three samplers ($\text{Sampler}_1 = \{\text{AT-GHMC}, \text{AT-HMC}\}$, $\text{Sampler}_2 = \text{RW}$) for both models, in terms of ESS/T, $\text{ESS} \in \{\text{minESS}, \text{meanESS}, \text{multiESS}\}$. AT-GHMC exhibits high efficiency across all metrics and models, outperforming the popular RW and AT-HMC by up to an order of magnitude in both models. Additionally, AT-GHMC shows visibly higher relative performance in the 4-parameter model than in the 3-parameter model for two out of three considered metrics. This may indicate the potential of the method for high dimensional problems. These results suggest that AT-GHMC provides a suitable framework for PDE calibration. On the other hand, AT-HMC displays worse performance than RW in two out of three metrics when increasing the dimension of the model. Most likely, an optimal choice of L for GHMC (17) does not necessarily translate to optimal performance for HMC, especially for higher dimensions.

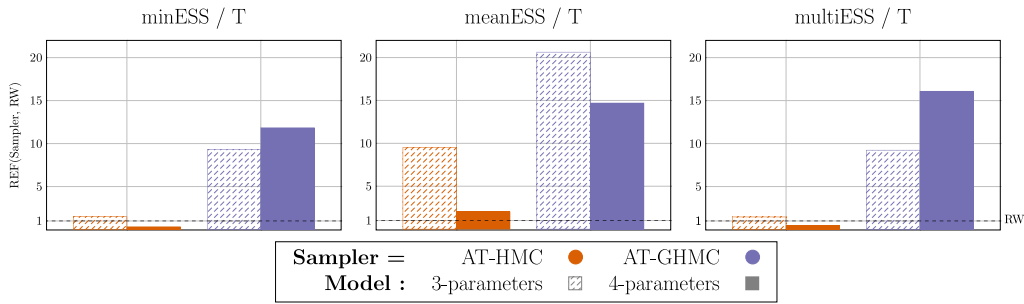


Fig. 10. Relative efficiency REF for minESS/T (left), meanESS/T (center), and multiESS/T (right) (T is a computational time) of AT-GHMC and AT-HMC with respect to RW. For the 3-parameter model, AT-GHMC (purple) and AT-HMC (orange) show improved performance over RW (dotted line) for all metrics. For the 4-parameter model, RW performs better than AT-HMC for two out of three metrics. AT-GHMC demonstrates the best performance by a wide margin, outperforming RW by a factor of up to 20 in the 3-parameter model and up to 15 times in the 4-parameter model.

Table 10

Simulation parameters for the numerical experiments on the Influenza A (H1N1) epidemics dataset with the SIR model. The optimal settings for AT-GHMC is found using the algorithm described in Section 7. AT-GHMC employs s-AIA3 integrator while NUTS uses standard Verlet. The mean values of L and Δt used by NUTS optimal routine are denoted as \bar{L} and $\bar{\Delta t}$, respectively.

Sampler	$N_{1,01}$	L	Δt	ϕ
AT-GHMC	2250	1	$\mathcal{U}(0.073, 0.105)$	$\mathcal{U}(0.088, 0.527)$
NUTS	140	$\bar{L} = 5$	$\bar{\Delta t} = 0.607$	-

9.3. Influenza A (H1N1) epidemics outbreak

Mechanistic disease transmission models are mathematical frameworks used to describe how infectious diseases spread through populations. These models commonly employ compartmental structures that classify individuals into distinct states (such as susceptible, infected, recovered), and define transitions between these states over time using systems of ODEs. Classical disease transmission models are often deterministic, meaning that the system’s evolution is fully determined by its parameters and initial conditions, always yielding the same outcome for a given setup. While deterministic models offer valuable insights into average system behavior, they fail to capture the inherent variability and stochasticity of real-world disease transmission. To address this limitation, mechanistic models can be enriched with statistical components that incorporate observational data to infer the underlying parameters. This is achieved by overlaying a probabilistic (or generative) model onto the deterministic structure. Bayesian inference, in particular, has proven highly effective for this purpose (refer to [51,75] and references therein).

As an illustrative example, we consider an outbreak of influenza A (H1N1) in 1978 at a British boarding school involving 763 students. The dataset, available in the R package `outbreaks` [76], records the daily number of students confined to bed over a 14-day period. Following [75], we model the outbreak dynamics using a classical SIR (Susceptible-Infected-Recovered) model, coupled with a negative binomial observation model to account for overdispersion in the reported case counts. The model includes the transmission rate, β , and the recovery rate, γ , which govern the transitions from susceptible to infected and from infected to recovered, respectively. For the prior distributions, we assume $\beta \sim \mathcal{N}^0(1, 1)$ and $\gamma \sim \mathcal{N}^0(0.4, 0.5)$, where \mathcal{N}^a denotes a normal distribution truncated at a . The prior specification is completed by assigning an exponential prior to the inverse of the dispersion parameter, ϕ , of the negative binomial distribution, i.e. $1/\phi \sim \text{Exp}(5)$. The initial conditions are set as $I_0 = 1$, $R_0 = 0$, and thus $S_0 = 762$.

As in Section 9.1, we compare the performance of two samplers: AT-GHMC and NUTS [12], as implemented in the R package `rstan` [64]. We note that, in both cases, the (approximate) solution to the ODE system is obtained using the CVODES component of the SUNDIALS suite [77]. Specifically, we use the ODE solver that relies on the backward differentiation formula (BDF) method. Similarly to the breast cancer study, both samplers are run with the same number of burn-in iterations, and their performance is assessed in terms of the grad/ESS metrics calculated over the first $N_{1,01} + 1000$ iterations after the burn-in (see Section 8.1).

The simulation settings for the numerical experiments are summarized in Table 10. We emphasize that, as with the cell adhesion model in Section 9.2, we bypass Hessian computations and the resulting computational overhead by adopting the cheaper S fitting factor approach (15) (see Appendix C in [39] for details). Fig. 11 shows the relative efficiency REF (19) of AT-GHMC over NUTS, evaluated in terms of grad/ESS, $\text{ESS} \in \{\text{minESS, meanESS, multiESS}\}$. In addition, to account for the extra computational cost incurred solely by NUTS due to its optimization process during the burn-in phase, we multiply the grad/ESS metrics for NUTS by a factor quantifying such overheads, δT_{NUTS} . This factor is carefully evaluated using the `rstan` package.

AT-GHMC with our proposed settings outperforms NUTS in terms of both meanESS and multiESS, with improvements increasing from a factor of 2x, when no NUTS computational overheads are considered, to 5x, otherwise. In contrast, NUTS achieves better sampling with the most restrictive metric, minESS, which is likely enabled by the use of a tuned mass matrix. This trend aligns with

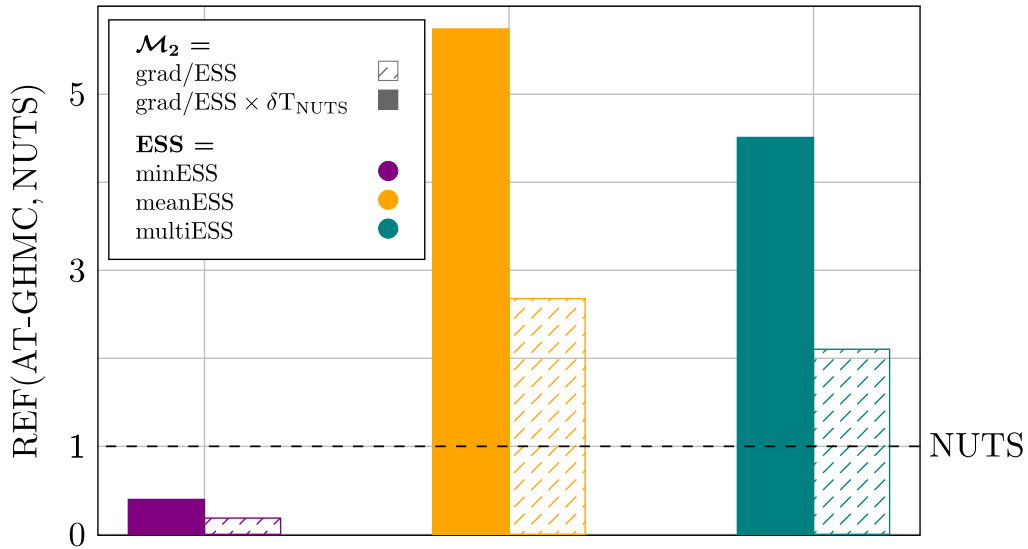


Fig. 11. Relative efficiency (REF) (19) in terms of grad/minESS (violet), grad/meanESS (yellow), and grad/multiESS (green) of AT-GHMC with respect to Stan NUTS. Filled bars correspond to the performance metrics that take into account the overheads introduced by NUTS at the optimization stage (quantified by δT_{NUTS}). AT-GHMC demonstrates superior performance with meanESS and multiESS, while NUTS outperforms AT-GHMC for minESS. When accounting for computational time differences, AT-GHMC shows around 5x improvement over NUTS for both meanESS and multiESS.

the findings from the breast cancer study (Fig. 8), where AT-GHMC shows more significant improvements over NUTS for average ESS metrics than for minESS. This suggests that tuning a position-dependent mass matrix during the NUTS burn-in stage may have a pronounced effect on the efficiency of an HMC algorithm in sampling complex distributions. Nevertheless, the computational time analysis shows that, in this case study, NUTS requires approximately twice the computational time of a tuned GHMC to achieve on average comparable sampling performance, as highlighted in Fig. 11.

10. Conclusion

The challenge of tuning appropriate hyperparameters to enhance HMC sampling performance has been extensively addressed by the statistical community. In contrast, GHMC, despite its promising potential to improve sampling efficiency due to the inherent irreversibility of its Markov chains, has not garnered comparable attention.

Here we present a computationally inexpensive adaptive tuning approach ATune which generates optimal parameter settings on the fly for HMC and GHMC simulations, bringing forth adaptively tuned AT-HMC and AT-GHMC methods. Given a simulated system, the new methodology delivers a set of system-specific optimized HMC/GHMC hyperparameters, without introducing any additional computational overhead during the production simulation stage. Owing to the flexibility of the HMC framework, AT-HMC and AT-GHMC are applicable to a broad spectrum of real-world Bayesian inference problems. Nevertheless, for non-differentiable posteriors or models with inaccessible or highly complex internal structure, gradient-free sampling approaches remain necessary, preventing such models from benefiting from the enhanced sampling efficiency offered by HMC and GHMC and their top-tier variants AT-HMC and AT-GHMC.

To determine an optimal integration step size, we build on the insights from the s-AIA adaptive integration approach introduced in [17]. We propose a system-specific randomization interval that minimizes energy error near the center of an estimated stability interval – the region typically associated with the most efficient performance [15,37]. Additionally, we confirm that the s-AIA3 integrator consistently outperforms fixed-parameter integrators when combined with both HMC and GHMC samplers. We notice that even following the general recommendations for GHMC settings leads to its superior performance over manually tuned HMC. These findings motivate a deeper exploration of the proper tuning of φ and L parameters for GHMC.

Leveraging the methodology proposed in [38], we develop an automated procedure for finding an optimal value for the random noise parameter in PMU (4), based solely on the size of the simulated system (assuming the simulation step size and numerical integrator are chosen as discussed). When applied to high-dimensional systems, this approach leads to smaller modifications of momentum, which, combined with the high accuracy provided by s-AIA3 and an optimal choice of Δt , facilitates the use of shorter trajectories, thus enhancing sampling efficiency in GHMC [31]. We note that, unlike the optimization procedures proposed for other hyperparameters, the selection of the optimal random noise φ can be performed prior a burn-in stage, relying only on the known dimensionality of the system. As a result, we propose employing the optimal φ from the very start of a GHMC simulation, rather than limiting it to the production stage only.

The introduced optimal hyperparameter settings exhibit superiority over other combinations of hyperparameters, leading to significant GHMC performance improvement across a set of popular benchmarks. Additionally, the performance gains observed when

comparing our results to state-of-the-art samplers in three distinct real-world case studies indicate the broad applicability and effectiveness of AT-GHMC for diverse probabilistic models. It is worth noting that, although the optimal choice of L is specifically tailored for GHMC, L_{opt} also enhances sampling efficiency when applied to HMC in some cases, especially of small dimensions.

The proposed ATune algorithm can be easily implemented in any software package for statistical modeling using Bayesian inference with Hamiltonian Monte Carlo to offer the advantages of AT-GHMC over conventional HMC.

CRedit authorship contribution statement

Elena Akhmatskaya: Writing – original draft, Visualization, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization; **Lorenzo Nagar:** Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation; **Jose Antonio Carrillo:** Writing – review & editing, Validation, Methodology, Funding acquisition, Data curation; **Leonardo Gavira Balmaciz:** Writing – original draft, Visualization, Validation, Software, Data curation; **Hristo Inouzhe:** Writing – review & editing, Validation, Software, Data curation; **Martín Parga Pazos:** Writing – original draft, Validation, Software, Data curation; **María Xosé Rodríguez Álvarez:** Writing – original draft, Validation, Data curation.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge Caetano Souto Maior for his valuable contributions at the early stages of this study, particularly to the implementation of the cell-cell adhesion PDE model in HaiCS. We thank Maria dM Vivanco for generously sharing her knowledge on the resistance to endocrine therapy in breast cancer and for the fruitful discussions.

This research was supported by MICIU/AEI/10.13039/501100011033 and by ERDF A way for Europe under Grant PID2022-136585NB-C22; by the Basque Government through ELKARTEK Programme under Grants KK-2024/00062; through TT24 under 6/12/TT/2024/00003, and through the BERC 2022–2025 program. We acknowledge the financial support by the Ministry of Science and Innovation through BCAM Severo Ochoa accreditation CEX2021-001142-S/MICIU/AEI/10.13039/501100011033 and PRE2022-104791 (L. G-B), and PLAN COMPLEMENTARIO MATERIALES AVANZADOS 2022–2025, PROYECTO 1101288. The authors acknowledge the financial support received from the grant BCAM-IKUR HPC&IA 2025–2026, HPCAI4, funded by the Basque Government by the IKUR Strategy and by the European Union NextGenerationEU/PRTR. J.A. C. was supported by the Advanced Grant Nonlocal-CPD (Nonlocal PDEs for Complex Particle Dynamics: Phase Transitions, Patterns and Synchronization) of the European Research Council Executive Agency (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 883363). M.X. R-A was partially funded by the Spanish State Research Agency through the Ramón y Cajal grant RYC2019-027534-I. This work has been possible thanks to the support of the computing infrastructure of the BCAM in-house cluster Hipatia, i2BASQUE academic network, Barcelona Supercomputing Center (RES, QHS-2025-1-0027), DIPC Computer Center and the technical and human support provided by IZO-SGI SGiker of UPV/EHU.

References

- [1] S. Duane, A.D. Kennedy, B.J. Pendleton, et al., Hybrid Monte Carlo, *Phys. Lett. B* 195 (2) (1987) 216–222. [https://doi.org/10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X)
- [2] P. Sountsov, M.D. Hoffman, Focusing on Difficult Directions for Learning HMC Trajectory Lengths, (2021). <https://doi.org/10.48550/arXiv.2110.11576>
- [3] S. Apers, S. Gribbling, D. Szilágyi, Hamiltonian Monte Carlo for efficient Gaussian sampling: long and random steps, (2022). <https://doi.org/10.48550/arXiv.2209.12771>
- [4] Y. Chen, K. Gattmiry, When does Metropolized Hamiltonian Monte Carlo provably outperform Metropolis-adjusted Langevin algorithm?, (2023). <https://doi.org/10.48550/arXiv.2304.04724>
- [5] T. Marwala, W.T. Mongwe, R. Mbuva, *Hamiltonian Monte Carlo Methods in Machine Learning*, Academic Press, London, London, 2023. <https://doi.org/10.1016/C2021-0-02845-5>
- [6] N. Bou-Rabee, B. Carpenter, M. Marsden, GIST: Gibbs self-tuning for locally adaptive Hamiltonian Monte Carlo, *arXiv:2404.15253* (2024). <https://doi.org/10.48550/arXiv.2404.15253>
- [7] N. Bou-Rabee, B. Carpenter, T.S. Kleppen, et al., Incorporating Local Step-Size Adaptivity into the No-U-Turn Sampler using Gibbs Self Tuning, (2024). <https://doi.org/10.48550/arXiv.2408.08259>
- [8] C. Modi, ATLAS: Adapting Trajectory Lengths and Step-Size for Hamiltonian Monte Carlo, (2024). <https://doi.org/10.48550/arXiv.2410.21587>
- [9] S. Biron-Lattes, M. and Surjanovic, N. and Syed, et al., AutoMALA: Locally adaptive Metropolis-adjusted Langevin algorithm, in: *Proc. 27th Int. Conf. Artif. Intell. Stat., 238 of Proc. Mach. Learn. Res.*, 2024, pp. 4600–4608. <https://proceedings.mlr.press/v238/biron-lattes24a.html>
- [10] A. Beskos, N. Pillai, G. Roberts, et al., Optimal tuning of the hybrid Monte Carlo algorithm, *Bernoulli* 19 (5A) (2013) 1501–1534. <http://www.jstor.org/stable/42919328>
- [11] M. Betancourt, S. Byrne, M. Girolami, Optimizing The Integrator Step Size for Hamiltonian Monte Carlo, (2014). <https://doi.org/10.48550/arXiv.1411.6669>
- [12] M.D. Hoffman, A. Gelman, The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, *J. Mach. Learn. Res.* 15 (1) (2014) 1593–1623. <https://www.jmlr.org/papers/volume15/hoffman14a/hoffman14a.pdf>

- [13] L. Verlet, Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules, *Phys. Rev.* 159 (1967) 98–103. <https://doi.org/10.1103/PhysRev.159.98>
- [14] T. Takaishi, P. de Forcrand, Testing and tuning symplectic integrators for the hybrid Monte Carlo algorithm in lattice QCD, *Phys. Rev. E* 73 (2006) 036706. <https://doi.org/10.1103/PhysRevE.73.036706>
- [15] S. Blanes, F. Casas, J.M. Sanz-Serna, Numerical integrators for the Hybrid Monte Carlo method, *SIAM J. Sci. Comput.* 36 (4) (2014) A1556–A1580. <https://doi.org/10.1137/130932740>
- [16] M. Fernández-Pendás, E. Akhmatskaya, J.M. Sanz-Serna, Adaptive multi-stage integrators for optimal energy conservation in molecular simulations, *J. Comput. Phys.* 327 (2016) 434–449. <https://doi.org/10.1016/j.jcp.2016.09.035>
- [17] L. Nagar, M. Fernández-Pendás, J.M. Sanz-Serna, et al., Adaptive multi-stage integration schemes for Hamiltonian Monte Carlo, *J. Comput. Phys.* 502 (2024) 112800. <https://doi.org/10.1016/j.jcp.2024.112800>
- [18] C. Tamborrino, F. Diele, C. Marangi, et al., Adaptive parameters tuning based on energy-preserving splitting integration for Hamiltonian Monte Carlo Method, *Commun. Nonlinear Sci. Numer. Simul.* 137 (2024) 108168. <https://doi.org/10.1016/j.cnsns.2024.108168>
- [19] L. Nagar, Optimising performance of Hamiltonian Monte Carlo (HMC) in molecular simulation and computational statistics, Ph.D. thesis, UPV/EHU, Bilbao (Spain), 2025. <https://doi.org/20.500.11824/2005>
- [20] N. Bou-Rabee, J.M. Sanz-Serna, Geometric integrators and the Hamiltonian Monte Carlo method, *Acta Numer.* 27 (2018) 113–206. <https://doi.org/10.1017/S0962492917000101>
- [21] S. Blanes, F. Casas, A. Murua, Splitting Methods for differential equations, *Acta Numer.* 33 (2024) 1–161. <https://doi.org/10.1017/S0962492923000077>
- [22] N. Bou-Rabee, J.M. Sanz-Serna, Randomized Hamiltonian Monte Carlo, *Ann. Appl. Probab.* 27 (2017) 2159–2194. <https://doi.org/10.1214/16-AAP1255>
- [23] M. Girolami, B. Calderhead, Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods, *J. R. Stat. Soc. Ser. B Stat. Methodol.* 73 (2011) 123–214. <https://doi.org/10.1111/j.1467-9868.2010.00765.x>
- [24] C.C. Margossian, A. Gelman, For how many iterations should we run Markov chain Monte Carlo?, (2023). <https://doi.org/10.48550/arXiv.2311.02726>
- [25] A.D. Kennedy, B. Pendleton, Cost of the generalised hybrid Monte Carlo algorithm for free field theory, *Nucl. Phys. B* 607 (3) (2001) 456–510. [https://doi.org/10.1016/S0550-3213\(01\)00129-8](https://doi.org/10.1016/S0550-3213(01)00129-8)
- [26] A.M. Horowitz, A generalized guided Monte Carlo algorithm, *Phys. Lett. B* 268 (2) (1991) 247–252. [https://doi.org/10.1016/0370-2693\(91\)90812-5](https://doi.org/10.1016/0370-2693(91)90812-5)
- [27] M. Ottobre, Markov Chain Monte Carlo and Irreversibility, *Rep. Math. Phys.* 77 (2016) 267–292. [https://doi.org/10.1016/S0034-4877\(16\)30031-3](https://doi.org/10.1016/S0034-4877(16)30031-3)
- [28] R.M. Neal, Improving Asymptotic Variance of MCMC Estimators: Non-reversible Chains are Better, (2004). <https://doi.org/10.48550/arXiv.math/0407281>
- [29] Y. Sun, J. Schmidhuber, F. Gomez, Improving the Asymptotic Performance of Markov Chain Monte-Carlo by Inserting Vortices, in: *Adv. Neural Inf. Process. Syst.*, 23, Curran Associates, Inc., 2010, pp. 2235–2243. https://proceedings.neurips.cc/paper_files/paper/2010/file/819f46e52c25763a55cc642422644317-Paper.pdf
- [30] Z. Song, Z. Tan, On Irreversible Metropolis Sampling Related to Langevin Dynamics, *SIAM J. Sci. Comput.* 44 (2022) A2089–A2120. <https://doi.org/10.1137/21M1423701>
- [31] Y. Fang, J.M. Sanz-Serna, R.D. Skeel, Compressible generalized hybrid Monte Carlo, *J. Chem. Phys.* 140 (2014) 174108. <https://doi.org/10.1063/1.4874000>
- [32] N. Gouraud, P.L. Bris, A. Majka, et al., HMC and underdamped Langevin united in the unadjusted convex smooth case, (2022). <https://doi.org/10.48550/arXiv.2202.00977>
- [33] L. Riou-Durand, J. Vogrinc, Metropolis Adjusted Langevin Trajectories: a robust alternative to Hamiltonian Monte Carlo, (2022). <https://doi.org/10.48550/arXiv.2202.13230>
- [34] M.D. Hoffman, P. Sountsov, Tuning-Free Generalized Hamiltonian Monte Carlo, in: *Proc. 25th Int. Conf. Artif. Intell. Stat.*, 151 of *Proc. Mach. Learn. Res.*, 2022, pp. 7799–7813. <https://proceedings.mlr.press/v151/hoffman22a.html>
- [35] J. Riou-Durand, L. and Sountsov, P. and Vogrinc, et al., Adaptive Tuning for Metropolis Adjusted Langevin Trajectories, in: *Proc. 26th Int. Conf. Artif. Intell. Stat.*, 206 of *Proc. Mach. Learn. Res.*, 2023, pp. 8102–8116. <https://proceedings.mlr.press/v206/riou-durand23a.html>
- [36] R.M. Neal, MCMC Using Hamiltonian Dynamics, in: S. Brooks, A. Gelman, G.L. Jones, X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo*, Chapman and Hall/CRC, 2011, pp. 113–162. <https://doi.org/10.1201/b10905-6>
- [37] A.K. Mazur, Common Molecular Dynamics Algorithms Revisited: Accuracy and Optimal Time Steps of Störmer-Leapfrog Integrators, *J. Comput. Phys.* 136 (2) (1997) 354–365. <https://doi.org/10.1006/jcph.1997.5740>
- [38] E. Akhmatskaya, M. Fernández-Pendás, T. Radivojević, et al., Adaptive Splitting Integrators for Enhancing Sampling Efficiency of Modified Hamiltonian Monte Carlo Methods in Molecular Simulation, *Langmuir* 33 (42) (2017) 11530–11542. <https://doi.org/10.1021/acs.langmuir.7b01372>
- [39] E. Akhmatskaya, L. Nagar, J.A. Carrillo, et al., Adaptive tuning of Hamiltonian Monte Carlo methods, (2025). <https://doi.org/10.48550/arXiv.2506.04082>
- [40] J.M. Sanz-Serna, M.P. Calvo, *Numerical Hamiltonian Problems*, Chapman and Hall, London, London, 1994. <https://doi.org/10.1137/1037075>
- [41] C.M. Campos, J.M. Sanz-Serna, Palindromic 3-stage splitting integrators, a roadmap, *J. Comput. Phys.* 346 (2017) 340–355. <https://doi.org/10.1016/j.jcp.2017.06.006>
- [42] R.I. McLachlan, On the Numerical Integration of Ordinary Differential Equations by Symmetric Composition Methods, *SIAM J. Sci. Comput.* 16 (1) (1995) 151–168. <https://doi.org/10.1137/0916010>
- [43] C. Predescu, R.A. Lippert, M.P. Eastwood, et al., Computationally efficient molecular dynamics integrators with improved sampling accuracy, *Mol. Phys.* 110 (9–10) (2012) 967–983. <https://doi.org/10.1080/00268976.2012.681311>
- [44] C.J. Geyer, Practical Markov Chain Monte Carlo, *Stat. Sci.* 7 (4) (1992) 473–483. <https://doi.org/10.1214/ss/1177011137>
- [45] D. Vehtari, A. and Gelman, A. and Simpson, et al., Rank-Normalization, Folding, and Localization: An Improved \hat{R} for Assessing Convergence of MCMC (with Discussion), *Bayesian Anal.* 16 (2) (2021) 667–718. <https://doi.org/10.1214/20-BA1221>
- [46] K. Plummer, M. and Best, N. and Cowles, et al., CODA: convergence diagnosis and output analysis for MCMC, *R News* 6 (1) (2006) 7–11. <http://oro.open.ac.uk/22547/>
- [47] T. Radivojević, M. Fernández-Pendás, J.M. Sanz-Serna, et al., Multi-stage splitting integrators for sampling with modified Hamiltonian Monte Carlo methods, *J. Comput. Phys.* 373 (2018) 900–916. <https://doi.org/10.1016/j.jcp.2018.07.023>
- [48] L. Nagar, M. Fernández-Pendás, J.M. Sanz-Serna, et al., Finding the optimal integration coefficient for a palindromic multi-stage splitting integrator in HMC applications to Bayesian inference [dataset], 2023, (Mendeley Data, v1). <https://doi.org/10.17632/5mmh4wccdd6.1>
- [49] T. Radivojević, Enhancing Sampling in Computational Statistics Using Modified Hamiltonians, Ph.D. thesis, UPV/EHU, Bilbao (Spain), 2016. <https://doi.org/20.500.11824/323>
- [50] T. Radivojević, E. Akhmatskaya, Modified Hamiltonian Monte Carlo for Bayesian inference, *Stat. Comput.* 30 (2) (2020) 377–404. <https://doi.org/10.1007/s11222-019-09885-x>
- [51] H. Inouzhe, M.X. Rodríguez-Álvarez, L. Nagar, et al., Dynamic SIR/SEIR-like models comprising a time-dependent transmission rate: Hamiltonian Monte Carlo approach with applications to COVID-19, (2023). <https://doi.org/10.48550/arXiv.2301.06385>
- [52] J.S. Liu, *Monte Carlo strategies in scientific computing*, 10, Springer, New York, New York, 2001. <https://doi.org/10.1007/978-0-387-76371-2>
- [53] M. Lichman, UCI Machine Learning Repository [dataset], 2013, <https://doi.org/10.24432/C5NC77>
- [54] A. Gelman, D.B. Rubin, Inference from Iterative Simulation Using Multiple Sequences, *Stat. Sci.* 7 (4) (1992) 457–472. <https://doi.org/10.1214/ss/1177011136>
- [55] S.P. Brooks, A. Gelman, General Methods for Monitoring Convergence of Iterative Simulations, *J. Comput. Graph. Stat.* 7 (4) (1998) 434–455. <https://doi.org/10.1080/10618600.1998.10474787>
- [56] J.M. Flegal, J. Hughes, D. Vats, et al., *mcmcse: Monte Carlo Standard Errors for MCMC*, Riverside, CA, and Kanpur, India, 2025. R package version 1.5-1.
- [57] D. Vats, J.M. Flegal, G.L. Jones, Multivariate output analysis for Markov chain Monte Carlo, *Biometrika* 106 (2019) 321–337. <https://doi.org/10.1093/biomet/asz002>
- [58] Netlib, LAPACK: Linear Algebra PACKage, 2023, <https://www.netlib.org/lapack/>. Accessed: 95 December 2025.

- [59] S. Lan, V. Stathopoulos, B. Shahbaba, et al., Markov Chain Monte Carlo from Lagrangian Dynamics, *J. Comput. Graph. Stat.* 24 (2015) 357–378. <https://doi.org/10.1080/10618600.2014.902764>
- [60] O. Piva, M. and Domenici, G. and Iriondo, et al., Sox2 promotes tamoxifen resistance in breast cancer cells, *EMBO Mol. Med.* 6 (1) (2014) 66–79. <https://doi.org/10.1002/emmm.201303411>
- [61] M.A. Mansournia, A. Geroldinger, S. Greenland, et al., Separation in Logistic Regression: Causes, Consequences, and Control, *Am. J. Epidemiol.* 187 (2018) 864–870. <https://doi.org/10.1093/aje/kwx299>
- [62] A. Gelman, A. Jakulin, M.G. Pittau, et al., A weakly informative default prior distribution for logistic and other regression models, *Ann. Appl. Stat.* 2 (2008) 1360–1383. <https://doi.org/10.1214/08-AOAS191>
- [63] Stan Development Team, *Stan Modeling Language Users Guide and Reference Manual*, Stan Development Team, 2024. Version 2.33, <https://mc-stan.org>.
- [64] S.D. Team, RStan: the R interface to Stan, 2024. R package v2.33 [software], <https://mc-stan.org/>.
- [65] N.J. Armstrong, K.J. Painter, J.A. Sherratt, A continuum approach to modelling cell-cell adhesion, *J. Theor. Biol.* 243 (1) (2006) 98–113. <https://doi.org/10.1016/j.jtbi.2006.05.030>
- [66] H. Murakawa, H. Togashi, Continuous models for cell-cell adhesion, *J. Theor. Biol.* 374 (2015) 1–12. <https://doi.org/10.1016/j.jtbi.2015.03.002>
- [67] J.A. Carrillo, Y. Huang, M. Schmidtchen, Zoology of a Nonlocal Cross-Diffusion Model for Two Species, *SIAM J. Appl. Math.* 78 (2) (2018) 1078–1104. <https://doi.org/10.1137/17M1128782>
- [68] J.A. Carrillo, H. Murakawa, M. Sato, et al., A population dynamics model of cell-cell adhesion incorporating population pressure and density saturation, *J. Theor. Biol.* 474 (2019) 14–24. <https://doi.org/10.1016/j.jtbi.2019.04.023>
- [69] J.A. Carrillo, H. Murakawa, M. Sato, et al., A new paradigm considering multicellular adhesion, repulsion and attraction represent diverse cellular tile patterns, *PLoS Comput. Biol.* 21 (2025) e1011909. <https://doi.org/10.1371/journal.pcbi.1011909>
- [70] A. Gerisch, M.A.J. Chaplain, Mathematical modelling of cancer cell invasion of tissue: Local and non-local models and the effect of adhesion, *J. Theor. Biol.* 250 (4) (2008) 684–704. <https://doi.org/10.1016/j.jtbi.2007.10.026>
- [71] M.J. Simpson, R.E. Baker, S.T. Vittadello, et al., Practical parameter identifiability for spatio-temporal models of cell invasion, *J. R. Soc. Interface* 17 (2020). <https://doi.org/10.1098/rsif.2020.0055>
- [72] C. Falcó, D.J. Cohen, J.A. Carrillo, et al., Quantifying tissue growth, shape and collision via continuum models and Bayesian inference, *J. R. Soc. Interface* 20 (2023). <https://doi.org/10.1098/rsif.2023.0184>
- [73] R. Bailo, J.A. Carillo, D. Gómez-Castro, Aggregation-diffusion Equations for Collective Behavior in the Sciences, 2024, <https://www.siam.org/publications/siam-news/articles/aggregation-diffusion-equations-for-collective-behavior-in-the-sciences/>. Accessed: 19 December 2025.
- [74] J.A. Carrillo, A. Chertock, Y. Huang, A Finite-Volume Method for Nonlinear Nonlocal Equations with a Gradient Flow Structure, *Commun. Math. Phys.* 17 (1) (2015) 233–258. <https://doi.org/10.4208/cicp.160214.010814a>
- [75] L. Grinsztajn, E. Semenova, C.C. Margossian, et al., Bayesian workflow for disease transmission modeling in Stan, *Stat. Med.* 40 (27) (2021) 6209–6234. <https://doi.org/10.1002/sim.9164>
- [76] T. Jombart, S. Frost, P. Nouvellet, et al., outbreaks: A Collection of Disease Outbreak Data, 2020. R package version 1.9.0, <https://CRAN.R-project.org/package=outbreaks>.
- [77] C.J. Hindmarsh, A. C. and Serban, R. and Balos, et al., User Documentation for CVODE v5. 7.0 (sundials v5. 7.0), 2021, https://computing.llnl.gov/sites/default/files/cvs_guide-5.7.0.pdf. Accessed: 19 December 2025.