Journal of Statistical Physics (2019) 174:692–714
https://doi.org/10.1007/s10955-018-2195-6

Simulating Coulomb and Log-Gases with Hybrid Monte Carlo Algorithms

Djalil Chafaï[1] · Grégoire Ferré[2]

Received: 17 August 2018 / Accepted: 16 November 2018 / Published online: 24 November 2018

Abstract

Coulomb and log-gases are exchangeable singular Boltzmann–Gibbs measures appearing in mathematical physics at many places, in particular in random matrix theory. We explore experimentally an efficient numerical method for simulating such gases. It is an instance of the Hybrid or Hamiltonian Monte Carlo algorithm, in other words a Metropolis–Hastings algorithm with proposals produced by a kinetic or underdamped Langevin dynamics. This algorithm has excellent numerical behavior despite the singular interaction, in particular when the number of particles gets large. It is more efficient than the well known overdamped version previously used for such problems, and allows new numerical explorations. It suggests for instance to conjecture a universality of the Gumbel fluctuation at the edge of beta Ginibre ensembles for all beta.

Keywords Numerical simulation · Random number generator · Singular Stochastic differential equation · Coulomb gas · Monte Carlo adjusted Langevin · Hybrid Monte Carlo · Markov chain Monte Carlo · Langevin dynamics · Kinetic equation

Mathematics Subject Classification 65C05 (Primary) · 82C22 · 60G57

We explore the numerical simulation of Coulomb gases and log-gases by mean of Hybrid or Hamiltonian Monte Carlo algorithms (HMC) [19,36]. Such algorithms consist basically in using discretized kinetic (underdamped) Langevin dynamics to produce proposals for Metropolis–Hastings algorithms. This can be viewed as a way to add momentum to a Monte Carlo interacting particle system. The basic outcome of this exploratory work is that HMC algorithms have remarkably good numerical behavior for such gases despite the singularity

✉ Djalil Chafaï
djalil@chafai.net
http://djalil.chafai.net/

Grégoire Ferré
gregoire.ferre@enpc.fr
https://team.inria.fr/matherials/team-members/gregoire-ferre/

[1] Université Paris-Dauphine, PSL, CNRS, CEREMADE, 75016 Paris, France

[2] Université Paris-Est, CERMICS (ENPC), INRIA, 77455 Marne-la-Vallée, France

of the interactions. Such algorithms scale well with the dimension of the system, see [4,8]. They are therefore more efficient than the tamed overdamped version already explored in the literature for instance in [55]. In this paper, we benchmark the capability of the algorithm to reproduce known results efficiently, and we make it ready to explore new conjectures.

Another advantage of this approach is that it could be adapted to take into account a submanifold constraint [51]. For instance, this could be used for simulating random matrices with prescribed trace or determinant, which is difficult to achieve by direct sampling of matrices.

For the sake of completeness, we should mention that there are remarkable alternative simulation algorithms which are not based on a diffusion process, such as the ones based on piecewise deterministic Markov processes (PDMP), see for instance [41] and [72].

1 Boltzmann–Gibbs Measures

We are interested in interacting particle systems subject to an external field and experiencing singular pair interactions. In order to encompass Coulomb gases as well as log-gases from random theory, we introduce a vector subspace S of dimension d of $\mathbb{R}^n$, with $n \geq 2$ and $n \geq d \geq 1$. The particles belong to S, and $\mathbb{R}^n$ is understood as a physical ambient space. We equip S with the trace of the Lebesgue measure of $\mathbb{R}^n$, denoted by $\mathrm{d}x$. The external field and the pair interaction are respectively denoted by $V : S \mapsto \mathbb{R}$ and $W : S \mapsto (-\infty, +\infty]$, and belong to $\mathcal{C}^2$ functions, with $W(x) < \infty$ for all $x \neq 0$. For any $N \geq 2$, we consider the probability measure P_N on $S^N = S \times \cdots \times S$ defined by

$$P_N(\mathrm{d}x) = \frac{\mathrm{e}^{-\beta_N H_N(x_1,\ldots,x_N)}}{Z_N} \mathrm{d}x_1, \ldots, \mathrm{d}x_N, \tag{1.1}$$

where $\beta_N > 0$ is a parameter,

$$Z_N = \int_{S^N} \mathrm{e}^{-\beta_N H_N(x_1,\ldots,x_N)} \mathrm{d}x_1, \ldots, \mathrm{d}x_N$$

is the normalizing factor, and

$$H_N(x_1, \ldots, x_N) = \frac{1}{N} \sum_{i=1}^{N} V(x_i) + \frac{1}{2N^2} \sum_{i \neq j} W(x_i - x_j)$$

is usually called energy or Hamiltonian of the system. We assume that β_N, V, and W are chosen in such a way that $Z_N < \infty$ for any N. The law P_N is invariant by permutation of the coordinates $x_1, \ldots, x_N$ (exchangeable), and H_N depends only on the empirical measure

$$\mu_N = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_i}.$$

Therefore P_N is also the law of a random empirical measure encoding a cloud of indistinguishable particles $x_1, \ldots, x_N$. We emphasize that the particles live on the space $S^N = S \times \cdots \times S$ of dimension dN. The parameter n serves as the physical dimension of the ambient space, for the Coulomb gas setting described next.

For any $m \geq 1$ and $x \in \mathbb{R}^m$, we denote by $|x| = \sqrt{x_1^2 + \cdots + x_m^2}$ the Euclidean norm of x. This matches the absolute value when $m = 1$ and the modulus when $m = 2$, $\mathbb{R}^2 \equiv \mathbb{C}$.

1.1 Coulomb Gases

The notion of Coulomb gas is based on elementary electrostatics. Here the vector subspace S is interpreted as a conductor. It corresponds to taking $W = g$ where g is the Coulomb kernel or Green function in the physical space $\mathbb{R}^n$. More precisely, recall that the Green function g in $\mathbb{R}^n$, $n \geq 2$, is defined for all $x \in \mathbb{R}^n$, $x \neq 0$, by

$$g(x) = \begin{cases} \log \frac{1}{|x|} & \text{if } n = 2, \\ \frac{1}{|x|^{n-2}} & \text{if } n \geq 3. \end{cases}$$

This function is the fundamental solution of the Poisson equation, namely, denoting by Δ the Laplace operator in $\mathbb{R}^n$ and by δ_0 the Dirac mass at 0, we have, in the sense of distributions,

$$-\Delta g = c\delta_0, \quad \text{with} \quad c = \begin{cases} 2\pi & \text{if } n = 2, \\ (n-2)|\mathbb{S}^{n-1}| = \frac{n(n-2)\pi^{n/2}}{\Gamma(1+n/2)} & \text{if } n \geq 3. \end{cases}$$

The physical interpretation in terms of electrostatics is as follows: $H_N(x_1, \ldots, x_N)$ is the electrostatic energy of a configuration of N electrons in $\mathbb{R}^n$ lying on S at positions $x_1, \ldots, x_N$, in an external field given by the potential V. The Green function or Coulomb kernel g expresses the Coulomb repulsion which is a two body singular interaction. The probability measure P_N can be seen as a Boltzmann–Gibbs measure, β_N playing the role of an inverse temperature. The probability measure P_N is known as a Coulomb gas or as a one-component plasma, see for instance [68] and references therein.

1.2 Log-Gases

A log-gas corresponds to choosing $d = n$ and a logarithmic interaction W whatever the value of n is, namely

$$W(x) = \log \frac{1}{|x|} = -\frac{1}{2} \log(x_1^2 + \cdots + x_d^2), \quad x \in S.$$

Coulomb gases and log-gases coincide when $d = n = 2$. In dimension $d = n \geq 3$, log-gases are natural and classical objects of approximation theory and can be seen as limiting Riesz potentials, namely $\lim_{\alpha \to 0} \frac{1}{\alpha}(|x|^{-\alpha} - 1)$, see for instance [68–70].

1.3 Static Energy and Equilibrium Measures

Under natural assumptions over V and W, typically when $\beta_N \gg N$ and V beats W at infinity, it is well known, see for instance [14,67] and references therein, that P_N almost surely, the empirical measure

$$\mu_N = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_i}$$

tends as $N \to \infty$ to a non random probability measure, the equilibrium measure

$$\mu_* = \arg\inf \mathcal{E},$$

Table 1 Examples of equilibrium measures for Coulomb gases, see [14,65]

d	S	n	V	μ_*	Nickname
1	$\mathbb{R}$	2	$\infty\mathbf{1}_{\text{interval}^c}$	Arcsine	
1	$\mathbb{R}$	2	x^2	Semicircle	GUE
2	$\mathbb{R}^2$	2	$\|x\|^2$	Uniform on a disc	Ginibre
$d \geq 3$	$\mathbb{R}^d$	d	$\|x\|^2$	Uniform on a ball	
$d \geq 3$	$\mathbb{R}^d$	d	Radial	Radial in a ring	

the unique minimizer of the strictly convex lower semi-continuous "energy" $\mathcal{E}$ defined by

$$\mu \mapsto \mathcal{E}(\mu) = \int V \mathrm{d}\mu + \iint W(x-y)\mu(\mathrm{d}x)\mu(\mathrm{d}y).$$

When $W = g$ is the Coulomb kernel, the quantity $\mathcal{E}(\mu)$ is the electrostatic energy of the distribution of charges μ, formed by the sum of the electrostatic potential coming from the external electric field V with the Coulomb self repulsion by mean of the Coulomb kernel g. Note that $\mathcal{E}(\mu) = \infty$ if μ has a Dirac mass due to the singularity of g. An Euler–Lagrange variational analysis reveals that when $S = \mathbb{R}^d$ and V is smooth, convex, and grows faster than g at infinity then the equilibrium probability measure μ_* is compactly supported and has density proportional to ΔV, see [14] and references therein. Table 1 gives examples of equilibrium measures in this Coulomb setting. We refer to [33,44,65,67,68] for old and new potential theory from this analytic point of view. Moreover, quite a few equilibrium measures are known for log-gases beyond Coulomb gases, see for instance [16].

Actually it can be shown that essentially if $\beta_N \gg N$ and V beats g at infinity then under $(P_N)_N$ the sequence of random empirical measures $(\mu_N)_N$ satisfies a large deviation principle with speed β_N and good rate function $\mathcal{E}$, see [3,14,30]. Concentration of measure inequalities are also available, see [12] and references therein.

1.4 Two Remarkable Gases from Random Matrix Theory

Let us give a couple of famous gases from random matrix theory that will serve as benchmark for our algorithm. They correspond to $n = 2$ because the Lebesgue measure on a matrix translates via the Jacobian of the change of variable to a Vandermonde determinant on the eigenvalues, giving rise to the two-dimensional Coulomb kernel inside the exponential via the identity

$$\prod_{i<j} |x_i - x_j| = \exp\left(\sum_{i<j} \log|x_i - x_j|\right).$$

Hence the name "log-gases". A good reference on this subject is [28] and we refer to [21,24,28,29,39] for more examples of Coulomb gases related to random matrix models. Coulomb gases remain interesting in any dimension n beyond random matrices, see [67,68].

Beta-Hermite model This model corresponds to

$$d = 1,\ n = 2,\ S = \mathbb{R},\ V(x) = \frac{x^2}{2\beta},\ W(x) = -\log|\cdot|,\ \beta_N = N^2\beta,\ \beta \in (0,\infty).$$

This means that the particles evolve on the line $\mathbb{R}$ with Coulomb interactions given by the Coulomb kernel in $\mathbb{R}^2$. For $\beta = 2$, it becomes the famous Gaussian Unitary Ensemble (GUE), which is the distribution of the eigenvalues of random $N \times N$ Hermitian matrices distributed according to the Gaussian probability measure with density proportional to $H \mapsto \mathrm{e}^{-N\mathrm{Tr}(H^2)}$. Beyond the case $\beta = 2$, the cases $\beta = 1$ and $\beta = 4$ correspond respectively to Gaussian random matrices with real and quaternionic entries. Following [21], for all $\beta \in (0, \infty)$, the measure P_N is also the distribution of the eigenvalues of special random $N \times N$ Hermitian tridiagonal matrices with independent but non identically distributed entries. Back to the case $\beta = 2$, the law P_N writes

$$(x_1, \ldots, x_N) \in \mathbb{R}^N \mapsto \mathrm{e}^{-\frac{N}{2}\sum_{i=1}^N x_i^2} \prod_{i<j} (x_i - x_j)^2. \tag{1.2}$$

In this case, the Coulomb gas P_N has a determinantal structure, making it integrable or exactly solvable for any $N \geq 2$, see [28,57]. This provides in particular a formula for the density of the mean empirical spectral distribution $\mathbb{E}\mu_N$ under P_N, namely

$$x \in \mathbb{R} \mapsto \frac{\mathrm{e}^{-\frac{N}{2}x^2}}{\sqrt{2\pi N}} \sum_{\ell=0}^{N-1} H_\ell^2(\sqrt{N}x), \tag{1.3}$$

where $(H_\ell)_{\ell \geq 0}$ are the Hermite polynomials which are the orthonormal polynomials for the standard Gaussian distribution $\mathcal{N}(0, 1)$. The equilibrium measure μ_* in this case is the Wigner semicircle distribution with the following density with respect to the Lebesgue measure:

$$x \in \mathbb{R} \mapsto \frac{\sqrt{4 - x^2}}{2\pi} \mathbf{1}_{x \in [-2,2]}. \tag{1.4}$$

A plot of μ_* and $\mathbb{E}\mu_N$ is provided in Fig. 1, together with our simulations. We refer to [46] for a direct proof of convergence of (1.3)–(1.4) as $N \to \infty$. Beyond the case $\beta = 2$, the equilibrium measure μ_* is still a Wigner semicircle distribution, scaled by β, supported by the interval $[-\beta, \beta]$, but up to our knowledge we do not have a formula for the mean empirical spectral distribution $\mathbb{E}\mu_N$, except when β is an even integer, see [21].

Beta-Ginibre model This model corresponds to

$$d = 2,\ n = 2,\ S = \mathbb{R}^2,\ V(x) = \frac{|x|^2}{\beta},\ W(x) = -\log|x|,\ \beta_N = N^2\beta,\ \beta \in (0, \infty).$$

In this case, the particles move in $\mathbb{R}^2$ with a Coulomb repulsion of dimension 2—it is therefore a Coulomb gas. As for the GUE, the law P_N can be written as

$$(x_1, \ldots, x_N) \in (\mathbb{R}^2)^N \mapsto \mathrm{e}^{-N\sum_{i=1}^N |x_i|^2} \prod_{i<j} |x_i - x_j|^\beta. \tag{1.5}$$

When $\beta = m$ for an even integer $m \in \{2, 4, \ldots\}$, the law of this gas matches the Laughlin wavefunction modeling the fractional quantum Hall effect (FQHE), see for instance [26].

For $\beta = 2$, this gas, known as the complex Ginibre Ensemble, matches the distribution of the eigenvalues of random $N \times N$ complex matrices distributed according to the Gaussian probability measure with density proportional to $M \mapsto \mathrm{e}^{-N\mathrm{Tr}(MM^*)}$ where $M^* = \overline{M}^\top$. In this case P_N has a determinantal structure, see [28,57]. This provides a formula for the density of the mean empirical spectral distribution $\mathbb{E}\mu_N$ under P_N, namely

$$x \in \mathbb{R}^2 \mapsto \frac{\mathrm{e}^{-N|x|^2}}{\pi} \sum_{\ell=0}^{N-1} \frac{|\sqrt{N}x|^{2\ell}}{\ell!}, \tag{1.6}$$

which is the analogue of (1.3) for the Gaussian Unitary Ensemble. Moreover, if $Y_1, \ldots, Y_N$ are independent and identically distributed Poisson random variables of mean $|x|^2$ for some $x \in \mathbb{R}^2$, then (1.6) writes

$$x \in \mathbb{R}^2 \mapsto \frac{1}{\pi}\mathbb{P}\left(\frac{Y_1 + \cdots + Y_N}{N} < 1\right).$$

As $N \to \infty$, by the law of large numbers, it converges to $1/\pi$ if $|x| < 1$ and to 0 if $|x| > 1$, while by the central limit theorem it converges to $1/(2\pi)$ if $|x| = 1$. It follows that $\mathbb{E}\mu_N$ converges weakly as $N \to \infty$ to the uniform distribution on the disk, with density

$$x \in \mathbb{R}^2 \mapsto \frac{\mathbf{1}_{|x|<1}}{\pi}, \tag{1.7}$$

which is the equilibrium measure μ_*. When N is finite, the numerical evaluation of (1.6) is better done by mean of the Gamma law. Namely, by induction and integration by parts, (1.6) writes

$$x \in \mathbb{R}^2 \mapsto \frac{1}{\pi(N-1)!}\int_{N|x|^2}^{\infty} u^{N-1}\mathrm{e}^{-u}\mathrm{d}u = \frac{\Gamma(N, N|x|^2)}{\pi},$$

where Γ is the normalized incomplete Gamma function and where we used the identity

$$\mathrm{e}^{-r}\sum_{\ell=0}^{N-1}\frac{r^\ell}{\ell!} = \frac{1}{(N-1)!}\int_r^{\infty} u^{N-1}\mathrm{e}^{-u}\mathrm{d}u.$$

Note that $t \mapsto 1 - \Gamma(N, t)$ is the cumulative distribution function of the Gamma distribution with shape parameter N and scale parameter 1. Figure 4 illustrates the difference between the limiting distribution (1.7) and the mean empirical spectral distribution (1.6) for a finite N. Beyond the case $\beta = 2$, we no longer have a formula for the density of $\mathbb{E}\mu_N$, but a simple scaling argument reveals that the equilibrium measure μ_* is in this case the uniform distribution on the centered disk of radius $\sqrt{\beta/2}$.

2 Simulating Log-Gases and Coulomb Gases

Regarding simulation of log-gases or Coulomb gases such as (1.1), it is natural to use the random matrix models when they are available. There exist also methods specific to determinantal processes which cover the log-gases of random matrix theory with $\beta = 2$, see [2,18,32,37,45,59,66]. Beyond these specially structured cases, a great variety of methods are available for simulating Boltzmann–Gibbs measures, such as overdamped Langevin diffusion algorithm, Metropolis–Hastings algorithm, Metropolis adjusted Langevin algorithm (MALA), and kinetic versions called Hybrid or Hamiltonian Monte Carlo (HMC) which are based on a kinetic (or underdamped) Langevin diffusion, see for instance [10,52]. Other possibilities exist, such as Nosé-Hoover dynamics [40] or piecewise deterministic Markov processes [9].

Two difficulties arise when sampling measures as (1.1). First, the Hamiltonian H_N involves all couples, so the computation of forces and energy scales quadratically with the number of particles. A natural way to circumvent this numerical problem is to use clusterization procedures such as the “fast multipole methods”, see for instance [35]. A second difficult feature of such a Hamiltonian is the singularity of the interacting function W, which typically results in numerical instability. A standard stabilization procedure is to «tame» the dynamics [11,38], which is the strategy adopted in [55]. However, this smoothing of the force induces a supplementary bias in the invariant measure, as shown in [11] for regular Hamiltonians. This requires using small time steps, hence long computations. In the present note, we explore for the first time the usage of HMC for general Coulomb gases in the context of random matrices, in the spirit of [71], the difficulty being the singularity of the interaction. This method has the advantage of sampling the exact invariant measure (1.1), while allowing to choose large time steps, which reduces the overall computational cost [27].

In Sect. 2.1, we review standard methods for sampling measures of the form $\mathrm{e}^{-\beta_N H_N}$, before presenting in detail the HMC algorithm in Sect. 2.2.

2.1 Standard Sampling Methods

To simplify and from now on, we suppose the support set S in (1.1) to be $\mathbb{R}^d$. We introduce the methods based on the overdamped Langevin dynamics. To sample approximately (1.1), the idea is to exploit the fact that P_N in (1.1) is the reversible invariant probability measure of the Markov diffusion process $(X_t)_{t\geq 0}$ solution to the stochastic differential equation:

$$\mathrm{d}X_t = -\alpha_N \nabla H_N(X_t)\,\mathrm{d}t + \sqrt{2\frac{\alpha_N}{\beta_N}}\,\mathrm{d}B_t, \tag{2.1}$$

or in other words

$$X_t = X_0 - \alpha_N \int_0^t \nabla H_N(X_s)\,\mathrm{d}s + \sqrt{2\frac{\alpha_N}{\beta_N}} B_t,$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion on S^N and $\alpha_N > 0$ is an arbitrary time scaling parameter (for instance $\alpha_N = 1$ or $\alpha_N = \beta_N$). The infinitesimal generator associated with (2.1) is

$$Lf = \frac{\alpha_N}{\beta_N}\Delta f - \alpha_N \nabla H_N \cdot \nabla f.$$

The difficulty in solving (2.1) lies in the fact that the energy H_N involves a singular interaction W, which may lead the process to explode. Actually, under certain conditions on β_N and V, the Eq. (2.1) is well posed, the process $(X_t)_{t\geq 0}$ is well defined, and

$$X_t \xrightarrow[t\to\infty]{\text{Law}} P_N,$$

for all non-degenerate initial condition X_0. See for instance [1,13,25] for the case of Beta-Hermite case known as the Dyson Ornstein–Uhlenbeck process, and [6] for the Beta-Ginibre case. We do not discuss these delicate aspects in this note. A convergence in Cesáro mean is provided by the ergodic theorem for additive functionals,

$$\frac{1}{t}\int_0^t \delta_{X_s}\,\mathrm{d}s \xrightarrow[t\to\infty]{\text{weak}} P_N$$

almost surely or, for any test function $f \in L^1(P_N)$,

$$\frac{1}{t}\int_0^t f(X_s)\,\mathrm{d}s \xrightarrow[t\to\infty]{} \int_S f\,\mathrm{d}P_N,$$

almost surely. It is also possible to accelerate the convergence by adding a divergence free term in the dynamics (2.1), see for instance [22,49] and references therein. This modification keeps the same invariant distribution but produces a non-reversible dynamics.

This method of simulation is referred to as an "unadjusted Langevin algorithm", a terminology which will be clarified later on. In practice, one cannot simulate the continuous stochastic process $(X_t)_{t\geq 0}$ solution to (2.1), and resorts to a numerical integration with a finite time step Δt. A typical choice is the Euler–Maruyama scheme [42,58], which reads

$$x_{k+1} = x_k - \nabla H_N(x_k)\alpha_N \Delta t + \sqrt{2\frac{\alpha_N}{\beta_N}\Delta t} G_k, \tag{2.2}$$

where (G_k) is a family of independent and identically distributed standard Gaussian variables, and x_k is an approximation of $X_{k\Delta t}$. Note that α_N and Δt play the same role here. However, because of the singularity of H_N, this sampling scheme leads to important biases in practice, and (2.2) may even lack an invariant measure [56, Sect. 6]. One way to stabilize the dynamics is to use a tamed version of (2.2), which typically takes the following form:

$$x_{k+1} = x_k - \frac{\nabla H_N(x_k)\alpha_N \Delta t}{1 + |\nabla H_N(x_k)|\alpha_N \Delta t} + \sqrt{2\frac{\alpha_N}{\beta_N}\Delta t} G_k. \tag{2.3}$$

This strategy is used in [55] but, as noted by the authors, the number of time steps needed to run a trajectory of fixed time T scales as $\Delta t \sim N^{-2}$, which makes the study of large systems difficult.

Another strategy is to add a selection step at each iteration. This is the idea of the Metropolis Adjusted (overdamped) Langevin Algorithm (MALA) [63], which prevents irrelevant moves with a Metropolis step. One can also view the MALA algorithm as a Metropolis algorithm in which the proposal is produces by using a one step discretization of the Langevin dynamics (2.1). Let us make this precise; more details can be found *e.g.* in [61,63].

Algorithm 2.1 (*Metropolis Adjusted* (*overdamped*) *Langevin Algorithm—MALA*) *Let K be the Gaussian transition kernel associated to the Markov chain of the Euler discretization* (2.2) *of the dynamics* (2.1). *For each step k,*

- *draw a proposal $\tilde{x}_{k+1}$ according to the kernel $K(x_k, \cdot)$,*
- *compute the probability*

$$p_k = 1 \wedge \frac{K(\tilde{x}_{k+1}, x_k)\mathrm{e}^{-\beta_N H_N(\tilde{x}_{k+1})}}{K(x_k, \tilde{x}_{k+1})\mathrm{e}^{-\beta_N H_N(x_k)}}, \tag{2.4}$$

- *set*

$$x_{k+1} = \begin{cases} \tilde{x}_{k+1} & \textit{with probability } p_k; \\ x_k & \textit{with probability } 1 - p_k. \end{cases}$$

Note that the "reversed" kernel $K(\cdot, x)$ is Gaussian only if H_N is a quadratic form. Note also that if the proposal kernel K is symmetric in the sense that $K(x, y) = K(y, x)$ for all x, y then it disappears in (2.4), and it turns out that this is the case for the Hybrid Monte Carlo algorithm described next (up to momentum reversal)!

A natural issue with these algorithms is the choice of Δt: if it is too large, an important fraction of the proposed moves will be rejected, hence poor convergence properties; conversely, if Δt is too small, many steps will be accepted but the physical ellapsed time will be small, hence a large variance for a fixed number of iterations. This algorithm actually has a nice scaling of the optimal time step Δt with the dimension of the system. Indeed, it can be shown that it scales as $\Delta t \sim N^{-\frac{1}{3}}$, at least for product measures (see [62] and references therein). Although this algorithm is already efficient, we propose to use a kinetic version with further advantages.

2.2 Hybrid Monte Carlo Algorithm

Hybrid Monte Carlo is built on Algorithm 2.1, but using a kinetic version of (2.1). For this, a momentum variable is introduced so as to improve the exploration of the space. Namely, set $E = \mathbb{R}^{dN}$, and let $U_N : E \to \mathbb{R}$ be smooth and such that $\mathrm{e}^{-\beta_N U_N}$ is Lebesgue integrable. Let $(X_t, Y_t)_{t\geq 0}$ be the diffusion process on $E \times E$ solution to the stochastic differential equation

$$\begin{cases} \mathrm{d}X_t = \alpha_N \nabla U_N(Y_t)\,\mathrm{d}t, \\ \mathrm{d}Y_t = -\alpha_N \nabla H_N(X_t)\,\mathrm{d}t - \gamma_N \alpha_N \nabla U_N(Y_t)\,\mathrm{d}t + \sqrt{2\dfrac{\gamma_N\alpha_N}{\beta_N}}\,\mathrm{d}B_t, \end{cases} \tag{2.5}$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion on E, and $\gamma_N > 0$ is an arbitrary parameter which plays the role of a friction, and which may depend a priori on N and $(X_t)_{t\geq 0}$, even if we do not use this possibility here. In addition, H_N and β_N are as in (1.1), while U_N plays the role of a generalized kinetic energy [71]. This dynamics admits the following generator:

$$Lf = \underbrace{-\alpha_N \nabla H_N(x)\cdot\nabla_y f + \alpha_N \nabla U_N(y)\cdot\nabla_x f}_{L_1} + \underbrace{\frac{\gamma_N\alpha_N}{\beta_N}\Delta_y f - \gamma_N\alpha_N\nabla U_N(y)\cdot\nabla_y f}_{L_2} \tag{2.6}$$

where L_1 is known as the Hamiltonian part while L_2 is called the fluctuation-dissipation part. The dynamics leaves invariant the product Boltzmann–Gibbs measure

$$R_N = P_N \otimes Q_N \quad \text{where} \quad Q_N(\mathrm{d}y) = \frac{\mathrm{e}^{-\beta_N U_N(y)}}{Z'_N}\mathrm{d}y,$$

see for instance [71]. In other words

$$R_N(\mathrm{d}x, \mathrm{d}y) = \frac{\mathrm{e}^{-\beta_N \widetilde{H}_N(x,y)}}{Z_N Z'_N}\mathrm{d}x\,\mathrm{d}y \quad \text{with} \quad \widetilde{H}_N(x,y) = H_N(x) + U_N(y). \tag{2.7}$$

As for the overdamped dynamics, the ergodic theorem for additive functionals gives

$$\frac{1}{t}\int_0^t \delta_{(X_s,Y_s)}\,\mathrm{d}s \underset{t\to\infty}{\overset{\text{weak}}{\longrightarrow}} R_N \quad \text{almost surely.}$$

Remark 2.2 (Terms: Hamiltonian, Langevin, overdamped, underdamped, kinetic) The dynamics (2.5) is called "Hamiltonian" when we turn off the noise by taking $\gamma_N = 0$. On the other hand, when $\gamma_N \to \infty$ and $\alpha_N \to 0$ with $\alpha_N\gamma_N = 1$, we recover (2.1) from (2.5) with Y_t and U_N instead of X_t and H_N. Both (2.1) and (2.5) are known as Langevin dynamics. To be more precise, (2.1) is generally called overdamped while (2.5) is referred to as kinetic or underdamped.

When $U_N(y) = \frac{1}{2}|y|^2$ then $Y_t = \mathrm{d}X_t/\mathrm{d}t$, and in this case X_t and Y_t can be interpreted respectively as the *position* and the *velocity* of a system of N points in S at time t. In this case we say that U_N is the *kinetic energy*. For simplicity, we specialize in what follows to this "physical" or "kinetic" case and refer to [71] for more possibilities.

As before, to simulate $(X_t, Y_t)_{t\geq 0}$, one can discretize (2.5) and sample from a trajectory. This will provide a proposal for the HMC scheme as the Euler discretization (2.2) did for Algorithm 2.1. A good way of doing this is a splitting procedure. First, one integrates the Hamiltonian part *i.e.* the operator L_1 in (2.6), which amounts to a standard Hamiltonian dynamics, before integrating the fluctuation-dissipation part *i.e.* the operator L_2 in (2.6). For discretizing the Hamiltonian dynamics over a time step, a standard approach is the Verlet integrator [31,50], which we describe now. For a time step $\Delta t > 0$, this scheme reads, starting from a state (x_k, y_k) at time k:

$$\begin{cases} y_{k+\frac{1}{2}} = y_k - \nabla H_N(x_k)\alpha_N \dfrac{\Delta t}{2}, \\ x_{k+1} = x_k + y_{k+\frac{1}{2}}\alpha_N \Delta t, \\ \tilde{y}_{k+1} = y_{k+\frac{1}{2}} - \nabla H_N(x_{k+1})\alpha_N \dfrac{\Delta t}{2}. \end{cases}$$

This corresponds to updating the velocity over half a time step, then the positions over a time step, and again the velocity over half a time-step. Given that this scheme only corresponds to the Hamiltonian part, it remains to integrate the fluctuation-dissipation part, corresponding to L_2 in (2.6). For quadratic energies, it is a simple Ornstein–Uhlenbeck process whose variance can be computed explicitly. Therefore, we add to the previous scheme the following velocity update which comes from the Mehler formula[1]:

$$y_{k+1} = \eta \tilde{y}_{k+1} + \sqrt{\frac{1-\eta^2}{\beta_N}} G_k, \quad \eta = \mathrm{e}^{-\gamma_N \alpha_N \Delta t},$$

where G_k is a standard Gaussian random variable. Like the numerical scheme (2.2), because of the singularity of the interactions, this integrator may not have an invariant measure [56], or its invariant measure may be a poor approximation of R_N depending on the time step [48]. Note that, here again, α_N and Δt play the same role.

Hybrid or Hamiltonian Monte Carlo (HMC) methods, built on the later integration, appeared in theoretical physics in lattice quantum chromodynamics with [19], see also [64], and are still actively studied in applied mathematics, see for instance [4,8,17,23,34,50,71] and references therein. The HMC algorithm can be thought of in a sense as a special Metropolis Adjusted (underdamped) Langevin Algorithm. Indeed, inspired by the MALA Algorithm 2.1, a way to avoid the stability problem of the discretization of the kinetic Langevin dynamics mentioned above is to add an acceptance-rejection step. A surprising advantage of this approach is that the Verlet integration scheme is time reversible up to momenta reversal [50, Sect. 2.1.3 and Eq. (2.11)], hence when computing the acceptance probability as in (2.4), the transition kernel does not appear. Note that the Verlet algorithm has been widely used for years by statistical physicists, and goes back to the historical works of Verlet [73] and Levesque and Verlet [53,54]. Let us now describe the algorithm.

Algorithm 2.3 [*HMC*] *Start from a configuration* (x_0, y_0) *and perform the following steps for each time* $k \geq 0$*:*

(1) *update the velocities with*

$$\tilde{y}_k = \eta y_k + \sqrt{\frac{1-\eta^2}{\beta_N}} G_k, \quad \eta = \mathrm{e}^{-\gamma_N \alpha_N \Delta t};$$

[1] The Mehler formula states that the Ornstein–Uhlenbeck process $(Z_t)_{t\geq 0}$ in $\mathbb{R}^n$ solution of the stochastic differential equation $\mathrm{d}Z_t = \sqrt{2\sigma^2}\mathrm{d}B_t - \rho Z_t \mathrm{d}t$ satisfies $\mathrm{Law}(Z_{t+s} \mid Z_s = z) = \mathcal{N}(z\mathrm{e}^{-\rho t}, \frac{1-\mathrm{e}^{-2\rho t}}{\rho}\sigma^2 I_n)$.

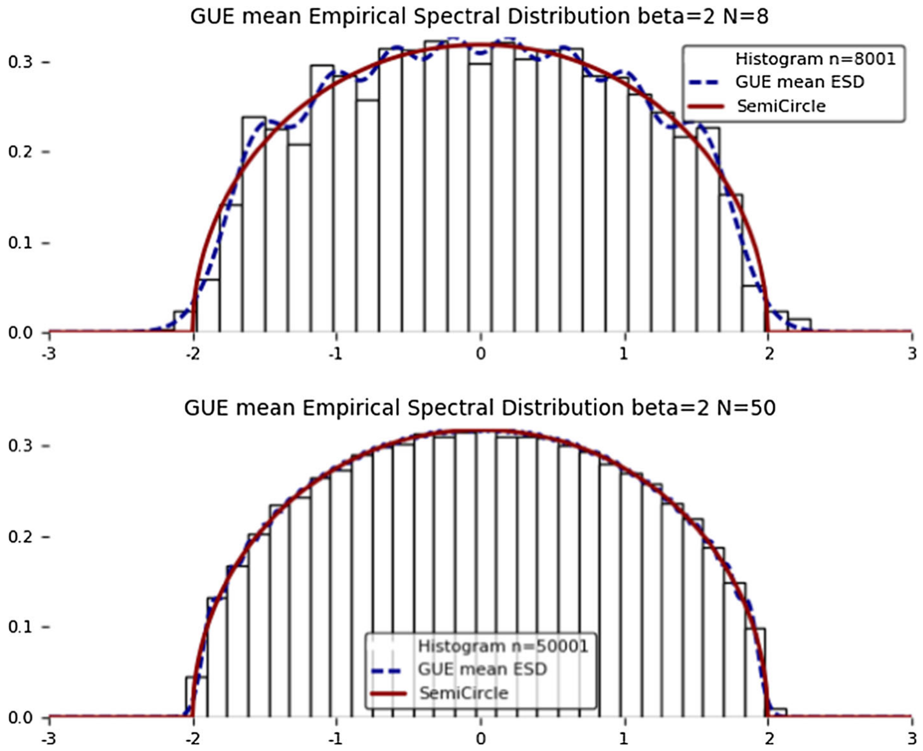

Fig. 1 Study of the Gaussian unitary ensemble with $N = 8$ (top) and $N = 50$ (bottom). The solid line is the plot of the limiting spectral distribution (1.4) while the dashed line is the plot of the mean empirical distribution (1.3). The bars form the histogram of simulations obtained using our HMC algorithm. This algorithm was run once with final-time $T = 10^6$ and time-step $\Delta t = 0.5$. The histogram was produced by looking at the last half of the trajectory and retaining the positions each 1000 time-steps, producing n values, namely $\approx 8 \times 10^3$ and $\approx 5 \times 10^4$ respectively

(2) *run one step of the Verlet scheme:*

$$\begin{cases} \tilde{y}_{k+\frac{1}{2}} = \tilde{y}_k - \nabla H_N(x_k)\alpha_N \dfrac{\Delta t}{2}; \\ \tilde{x}_{k+1} = x_k + \tilde{y}_{k+\frac{1}{2}}\alpha_N \Delta t; \\ \tilde{y}_{k+1} = \tilde{y}_{k+\frac{1}{2}} - \nabla H_N(x_{k+1})\alpha_N \dfrac{\Delta t}{2}; \end{cases} \tag{2.8}$$

(3) *compute the probability ratio*

$$p_k = 1 \wedge \exp\left[-\beta_N\left(H_N(\tilde{x}_{k+1}) + \frac{\tilde{y}_{k+1}^2}{2} - H_N(x_k) - \frac{\tilde{y}_k^2}{2}\right)\right];$$

(4) *set*

$$(x_{k+1}, y_{k+1}) = \begin{cases} (\tilde{x}_{k+1}, \tilde{y}_{k+1}) & \textit{with probability } p_k; \\ (x_k, -\tilde{y}_k) & \textit{with probability } 1 - p_k. \end{cases}$$

As noted in the various references above, the Metropolis step acts as a corrector on the energy conservation of the Hamiltonian step. In this, it helps avoiding irrelevant moves,

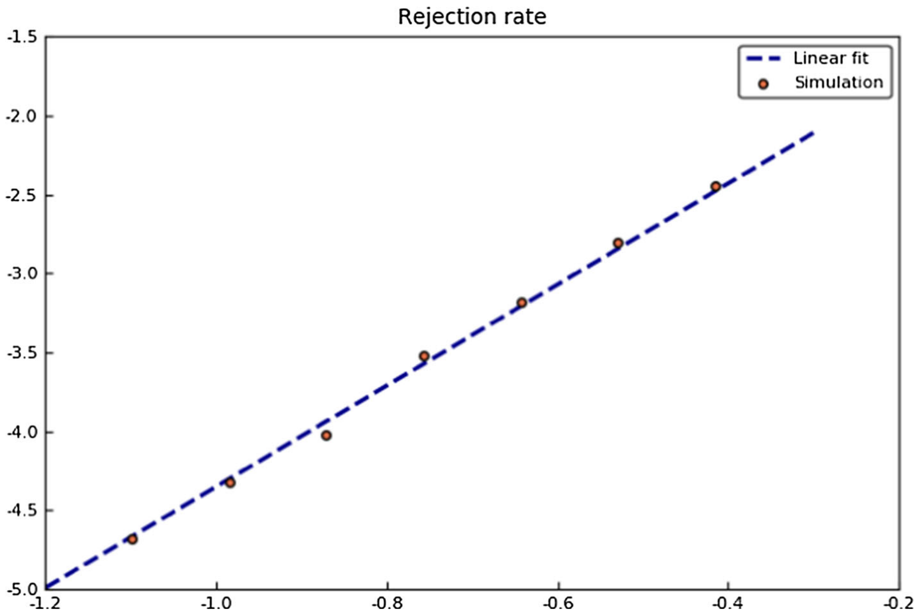

Fig. 2 Evolution of the rejection rate in Algorithm 2.3 as Δt goes to zero, for the Gaussian unitary ensemble with $N = 50$, $\beta = 2$ and $T = 10^5$ (in log–log coordinate)

while enhancing the exploration capacities of the dynamics through the speed variable. A more precise argument in favor of this algorithm is the scaling of the time step Δt with respect to the system size N. Indeed, as shown in [4] for product measures, the optimal scaling is as $\Delta t \sim N^{-\frac{1}{4}}$, which makes the algorithm appealing for large systems. Since the Hamiltonian computational cost scales as N^2, we see that the cost of the algorithm for a fixed time T and $N = \lceil [T/\Delta t] \rceil$ is in $\mathcal{O}(N^{\frac{9}{4}})$, which has to be compared to the $\mathcal{O}(N^4)$ cost reached in [55]. Finally, the parameter γ_N can also be tuned in order to optimize the speed of convergence – we leave this point here and stick to $\gamma_N = 1$.

The control of the error or rate of convergence for the HMC algorithm is the subject of active research, see for instance [47] and [7,23] for some results under structural assumptions.

From a practical point of view, the algorithm can be tested in the following way. First, when only the Hamiltonian part of the dynamics is integrated with the Verlet scheme (2.8), it can be checked that the energy variation over one time step scales as Δt^3 as $\Delta t \to 0$. Then, if the selection step is added, the rejection rate should also scale as Δt^3. When the momentum resampling is added, this rejection rate scaling should not change. For completeness, we illustrate some of these facts in Sect. 3.

3 Numerical Experiments on Remarkable Models

In this section, we start testing Algorithm 2.3 for the two cases described in Sect. 1.4. Since the equilibrium measures are known for any $N \geq 2$, we will be able to compare accurately our results with the expected one. We will also consider models for which the empirical spectral distribution and the equilibrium distribution are not known. We remind that when $S = \mathbb{R}^d$ with $d \geq 1$ we have the following formulas that hold in any dimension:

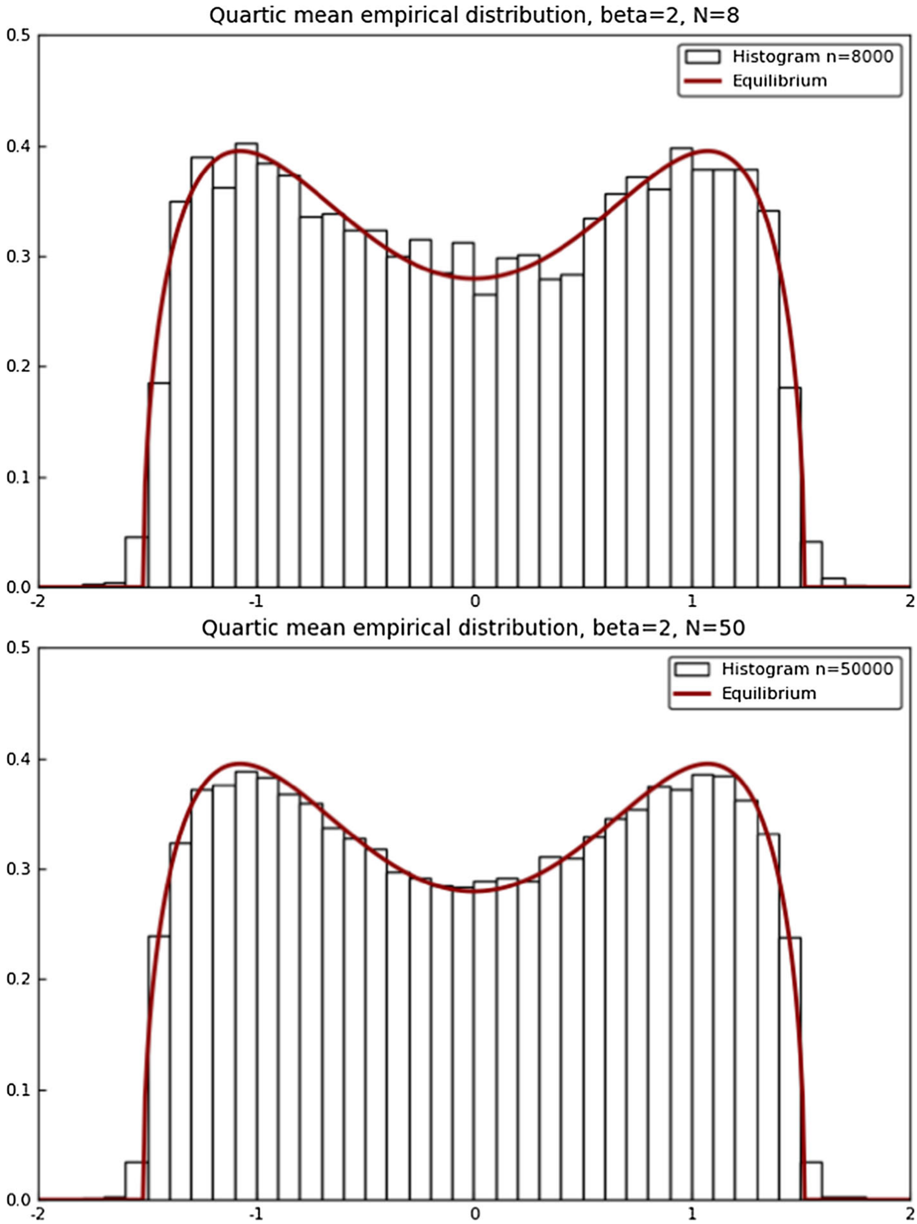

Fig. 3 Study of the quartic confinement with $N = 8$ (top) and $N = 50$ (bottom). The solid line is the plot of the limiting spectral distribution (1.4). The bars form the histogram of simulations obtained using our HMC algorithm. This algorithm was run once with final-time $T = 10^6$ and time-step $\Delta t = 0.5$. The histogram was produced by looking at the last half of the trajectory and retaining the positions each 1000 time-steps, producing n values namely $\approx 8 \times 10^3$ and $\approx 5 \times 10^4$ respectively. We do not have a formula for the mean empirical distribution for this model. This gas describes the law of the eigenvalues of a random symmetric tridiagonal matrix model but its entries are not independent, see [43, Prop. 2]

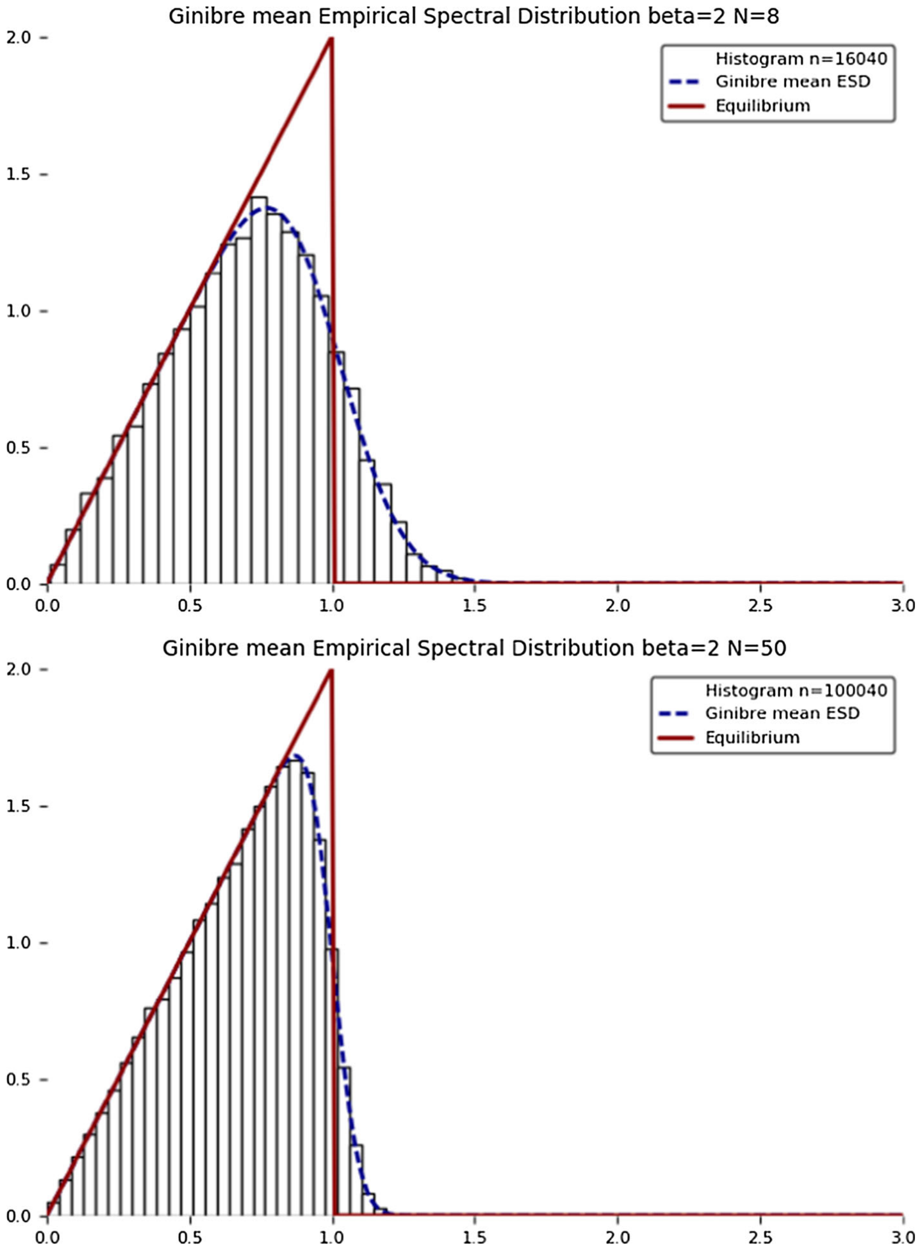

Fig. 4 Study of the complex Ginibre ensemble with $N = 8$ (top) and $N = 50$ (bottom). The solid line is the plot of the limiting spectral distribution (1.7) while the dashed line is the plot of the mean empirical distribution (1.6), both as functions of the radius $|z|$ and scaled by 2π (in order to obtain a radial density). The bars form the histogram of simulations obtained using our HMC algorithm. This algorithm was run 40 times with final-time $T = 10^5$ and time-step $\Delta t = 0.1$. The histogram was produced by looking at the last halves of the 40 trajectories and retaining the positions each 10000 time-steps, producing n values namely $\approx\ 16 \times 10^3$ and $\approx\ 10^5$ respectively

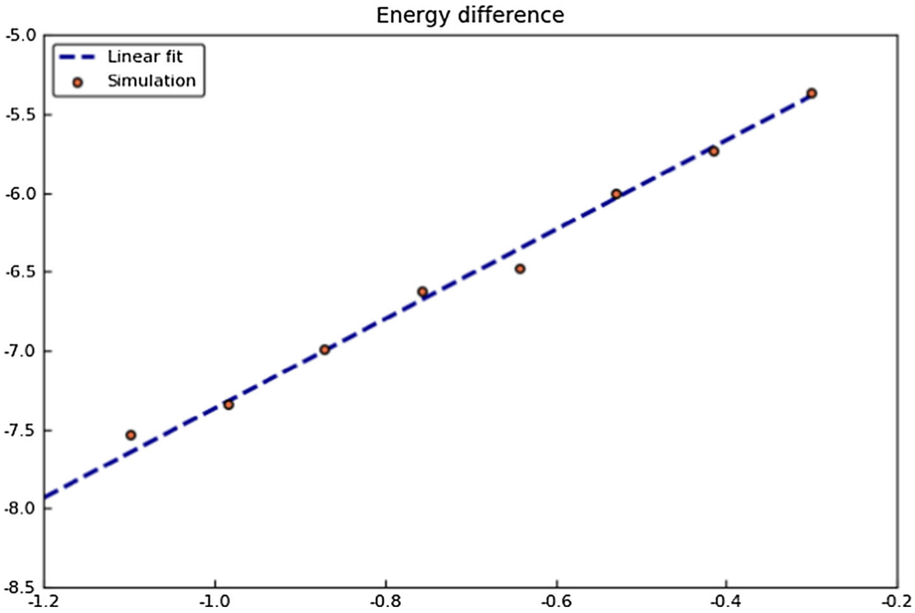

Fig. 5 Evolution of the energy difference in Algorithm 2.3 as Δt goes to zero, for the complex Ginibre ensemble with $N = 50$, $\beta = 2$ and $T = 10^3$ (in log–log coordinate)

$$\nabla |x|^2 = 2x, \quad \nabla \log \frac{1}{|x|} = -\frac{x}{|x|^2}, \quad \nabla \frac{1}{|x|} = -\frac{x}{|x|^3}.$$

3.1 Case Study: 1D

We test the numerical method by looking at the mean empirical distribution in the case of the Gaussian Unitary Ensemble (1.2) with $\beta = 2$, $N = 8$, for which the exact expression of $\mathbb{E}\mu_N$ under P_N is provided by (1.3). The results in Fig. 1 show a very good agreement between the exact result and the algorithm. For completeness, we study the rejection rate of the algorithm as Δt goes to zero, as mentioned at the end of Sect. 2.2. More precisely, we compute over a trajectory the rate of rejected moves in the Step 4 of Algorithm 2.3. The logarithmic plot in Fig. 2 shows a linear fit with a slope of about 3.1, which confirms the expected scaling in Δt^3.

We also study the quartic confinement potential $V(x) = x^4/4$, as in [55]. In this case, the empirical spectral distribution is not known, but the equilibrium distribution has density with respect to the Lebesgue measure given by

$$x \in \mathbb{R} \mapsto (2a^2 + x^2)\frac{\sqrt{4a^2 - x^2}}{2\pi}\mathbf{1}_{x \in [-2a,2a]}, \quad a = 3^{-\frac{1}{4}}.$$

The results of the numerical simulations, see Fig. 3, show a good agreement with the equilibrium measure when N is large. Note that a tridiagonal random matrix model is known but it does not have independent entries, see [43, Prop. 2.1].

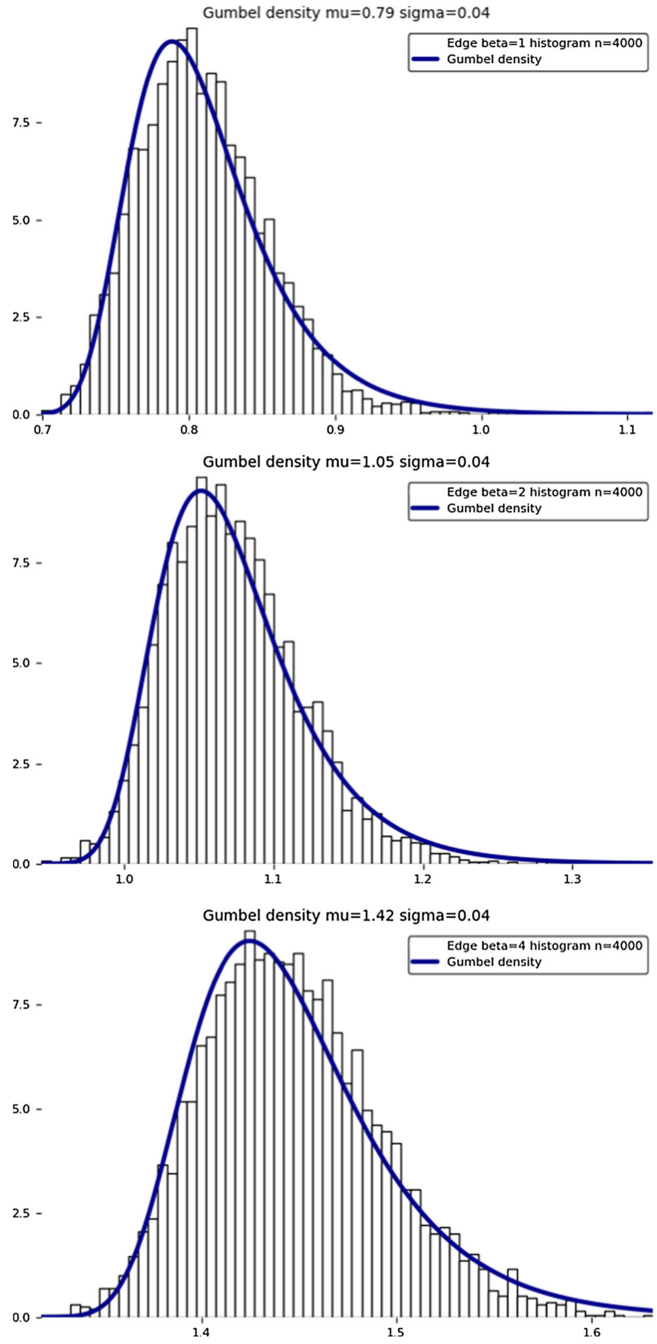

Fig. 6 Study of the fluctuation of the largest particle in modulus for the β complex Ginibre ensemble with $N = 50$, in the cases $\beta \in \{1, 2, 4\}$. The solid line is the plot of the fit with a translation-scale Gumbel distribution. The Gumbel fluctuation is proved only in the case $\beta = 2$, see [15,60]. These simulations suggest to conjecture that the Gumbel fluctuation is valid for any $\beta > 0$. The simulation matches pretty well the edge support at $\sqrt{\beta/2}$ and suggests that the variance is not very sensitive to β

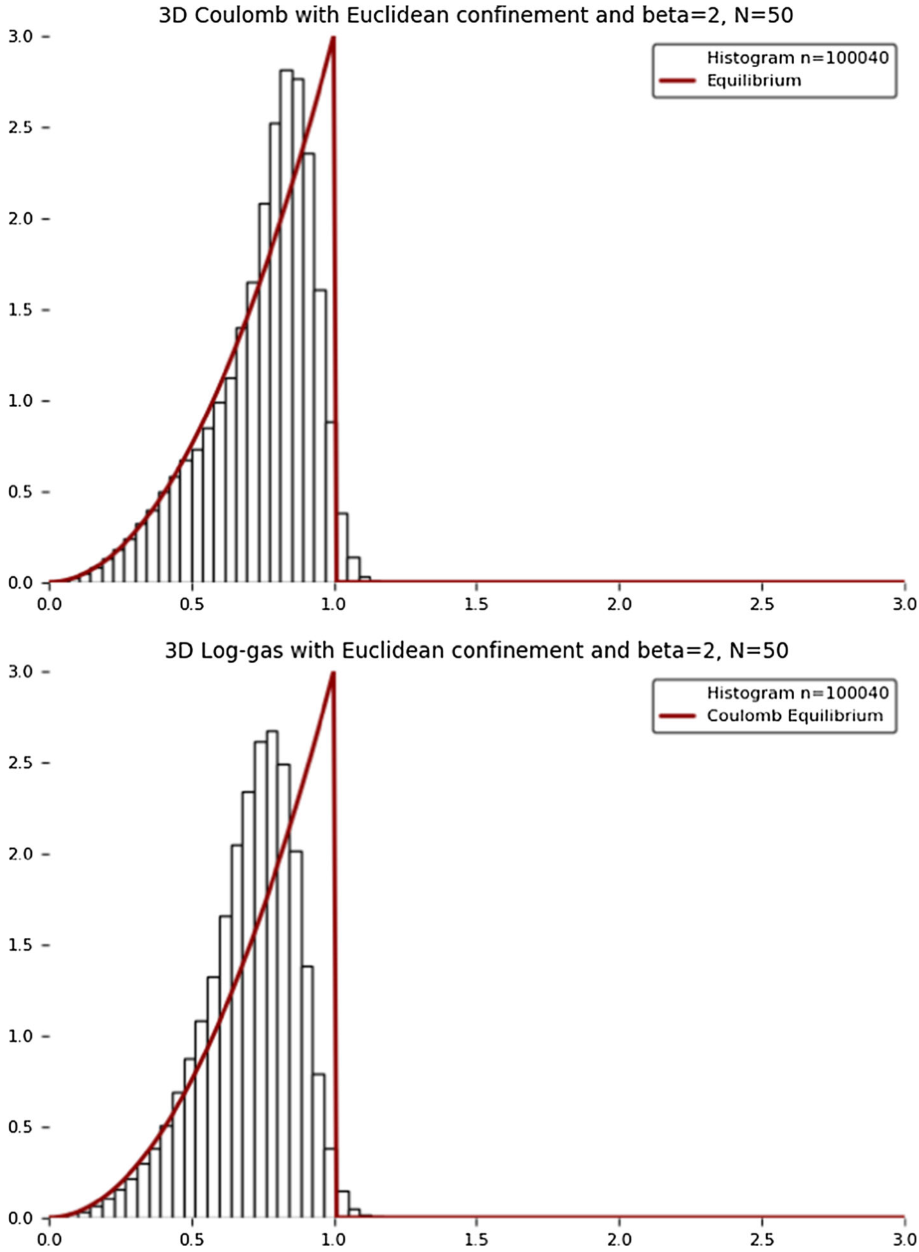

Fig. 7 Study of the 3D Coulomb case (top) and 3D Log-gas (bottom) with Euclidean confinement and $\beta = 2$ and $N = 50$. Equilibrium measure in solid line and histogram obtained with our HMC algorithm with $N = 50$ and same simulation parameters as for Fig. 4. In contrast with the GUE case and the Ginibre case, we do not have a formula for the mean empirical distribution at fixed N for both cases, and for the Log-gas (bottom) the equilibrium measure is not known

3.2 Case Study: 2D

We next consider in Fig. 4 the mean empirical distribution in the case of the Complex Ginibre Ensemble (1.5) with $\beta = 2$, $N = 8$. In this case, we also know a theoretical formula for $\mathbb{E}\mu_N$ under P_N, given by (1.6). For completeness, we investigate the scaling of the relative energy difference in the Step 3 of Algorithm 2.3 (by turning off the selection procedure of Step 4). The logarithmic plot in Fig. 5 shows a slope of about 2.9, which confirms the expected scaling in Δt^3 that corresponds to the error of energy conservation, over one time step, of the Verlet integrator (2.8).

We explore next in Fig. 6 the Gumbel fluctuation at the edge, which is proved for $\beta = 2$ and conjectured for $\beta \neq 2$, see [15,20,60] (note that in this case we have a formula for μ_* but not for $\mathbb{E}\mu_N$ under P_N). One could also explore the crystallization phenomenon, see [5] and references therein.

3.3 Case Study: 3D

In Fig. 7, we finally turn to the Coulomb gas which corresponds to $S = \mathbb{R}^3$, $d = n = 3$, $V = |\cdot|^2/\beta$, $W = 1/|\cdot|$ and to the log-gas for which $W = -\log|\cdot|$. In the first case the equilibrium measure μ_* is uniform on the centered ball of $\mathbb{R}^d$ of radius $(\beta(d-2)/2)^{1/d}$, see for instance [14, Cor. 1.3], while in the second case the equilibrium measure is not know yet, see however [16]. In both cases we do not have a formula for $\mathbb{E}\mu_N$ under P_N. One could study the fluctuation at the edge, which is conjectured to be Gumbel, just like for the complex Ginibre ensemble in 2D.

Acknowledgements We warmly thank Gabriel Stoltz for his encouragements and for very useful discussions on the theoretical and numerical sides of this work. We are also grateful to Thomas Leblé and Laure Dumaz for their comments on the first version.

Appendix A: Julia Code

Here is a program written in the Julia language[2] illustrating our method. It allows to exploit the multiple cores of modern processors and works in parallel on clusters. Beware that this code is not fully optimized, for instance the energy and its gradient could be computed simultaneously for better performance.

```
#---------------------------------------------------------------------------#
#-------------- Simulating coulomb gases with HMC algorithm ----------------#
#---------------------------------------------------------------------------#

# Tested with Julia 1.0. D. Chafai + G. Ferre : https://arxiv.org/abs/1806.05985

using Distributed # for @everywhere and nprocs()
@everywhere using Printf # for @sprintf()
@everywhere using LinearAlgebra # for norm()
@everywhere using DelimitedFiles # for Base.writedlm()

@everywhere begin # for parallel computing: julia -p NumberOfAdditionalProcesses
    #-----------------------------------------------------------------------#
    # Customization part : parameters, confinement, and interaction         #
    # ----------------------------------------------------------------------#

    ## Parameters. Note that in this code U_N(y)=|y|^2/2.

    # Final time and time step
    const T = 1e4
    const dt = 0.1
    # Number of eigenvalues
```

[2] http://JuliaLang.org/

```
const N = 8
# Dimension of the physical space
const dim = 1 # works for dimensions 1, 2, 3
# Temperature and friction
const beta = 2.
# Riesz parameter for Riesz interaction
const s = 1.

## Functions

# Confinement potential V and its gradient
@inline function confinement(x)
    return dot(x,x)/(2*beta) # 1D Beta-Hermite
    # return dot(x,x)/beta  # 2D Beta-Ginibre, 3D Beta-Coulomb
end
@inline function confinement_gradient(x)
    return x/beta      # 1D Beta-Hermite
    # return 2*x/beta # 2D Beta-Ginibre, 3D Beta-Coulomb
end

# Interaction potential W and its gradient
@inline function interaction(x,y)
    return -log(norm(x-y)) # 1D Beta-H., 2D Beta-Gin., 2D/3D Beta log-gas.
    # return 1/norm(x-y)   # 3D Beta-Coulomb
    # return 1/norm(x)^s # Riesz
end
@inline function interaction_gradient(x,y)
    v = x-y
    return -v/norm(v)^2 # 1D Beta-H., 2D Beta-Gin
    # return -v/norm(v)^3  # 3D Beta-Coulomb
    # return -s*x/norm(x)^(s+2) # Riesz
end

#--------------------------------------------------------------------#
#--- Parameters computed from inputs --------------------------------#
#--------------------------------------------------------------------#

const alphan = 1.
const betan = beta * N^2
const gamman = 1. / alphan
# Parameters for discretisation of fluctuation-dissipation part L2
const etan = exp(- gamman * alphan * dt)
const sdn = sqrt((1-etan^2)/betan)
#-- I/O parameter, write the configuration every niterio steps
const niterio = 1000
# Number of iterations and number of outputs
const niter = Int64(round(T/dt))
const nsteps = Int64(round(niter/niterio))

#--------------------------------------------------------------------#
#----------- Core part - Be careful and good luck ! -----------------#
#--------------------------------------------------------------------#

## Functions

# Potential energy H_N
@inline function energy(X)
    ener = 0
    @inbounds for i = 1:N
        @inbounds for j = i+1:N
            ener += interaction(X[i],X[j]) /N
        end
        ener += confinement(X[i])
    end
    return ener /N
end # function energy()

# Kinetic energy U_N
@inline function kinetic(Y)
    return norm(Y)^2 /2.
end # function kinetic()

# Force applied on particle at X[i] from all others at positions X[j] j!=i
@inline function compute_force!(X,F) # -Grad H_N
    # Computation of interaction forces between each pairs
    Fpairs = Array{Vector{Float64}}(undef, N,N) # we use only N(N-1)/2 entries
    @inbounds for i = 1:N
        @inbounds for j = 1:i-1
            Fpairs[i,j] = -interaction_gradient(X[i],X[j])
        end
    end
    # Computation of total force on each particle
    @inbounds for i = 1:N
        F[i] = zeros(dim)
        # Interaction
        @inbounds for j = 1:i-1
            F[i] += Fpairs[i,j]
        end
        @inbounds for j = i+1:N
            F[i] -= Fpairs[j,i]
        end
        F[i] /= N
        # Confinement
```

```
            F[i] -= confinement_gradient(X[i])
            F[i] /= N
        end
    end # function compute_force!()

    # compute the new force and speed
    @inline function verlet_integrator!(Fnew, Fcur, Xnew, Ynew, X, Y)
        @inbounds for i=1:N
            Ynew[i] = Y[i] + Fcur[i] * alphan * dt/2.
            Xnew[i] = X[i] + Ynew[i] * alphan * dt
        end
        compute_force!(Xnew,Fnew)
        @inbounds for i=1:N
            Ynew[i] += Fnew[i] * alphan * dt/2
        end
    end # function verlet_integrator!()

    # update positions and speed
    function update!(X, Y, Fcur, Xnew, Ynew, Fnew, Epot, acceptrate)
        #-- Speed resampling
        @inbounds for i = 1:N
            Y[i] = etan * Y[i] + sdn * randn(dim)
        end
        Ekin = kinetic(Y)
        Energy = Epot + Ekin
        #-- Verlet integrator. Position-speed proposal will be in (Xnew,Ynew).
        verlet_integrator!(Fnew, Fcur, Xnew, Ynew, X, Y)
        # New energy
        Epotnew = energy(Xnew)
        Ekinnew = kinetic(Ynew)
        NewEnergy = Epotnew + Ekinnew
        # Metropolis ratio
        r = beta * (- NewEnergy + Energy)
        # Selection-rejection step
        if log(rand()) <= r
            # acceptation
            @inbounds @simd for i = 1:N
                X[i] = Xnew[i]
                Y[i] = Ynew[i]
                Fcur[i] = Fnew[i]
            end
            acceptrate[1] += 1
            Epot = Epotnew
        else # rejection: speed inversion
            @inbounds @simd for i = 1:N
                Y[i] = -Y[i]
            end
        end
        return Epot
    end # function update()

    # Runs a trajectory of HMC algorithm and compute averages
    function HMC(runid)
        #-- For output : for positions/velocities every niterio steps
        TrajectoryX = Array{Float64}(undef, nsteps, N*dim)
        TrajectoryY = Array{Float64}(undef, nsteps, N*dim)
        #-- For output : Acceptation rate for the HMC selection step
        acceptrate = zeros(1)
        # Local variables
        #-- configuration and speed
        X = Vector{Vector{Float64}}(undef, N)
        Y = Vector{Vector{Float64}}(undef, N)
        #-- initial forces
        Fcur = Vector{Vector{Float64}}(undef, N)
        #-- Same quantities for the proposal
        Ynew = Vector{Vector{Float64}}(undef, N)
        Xnew = Vector{Vector{Float64}}(undef, N)
        Fnew = Vector{Vector{Float64}}(undef, N)
        # random initial configuration with uniform law on a square
        for i = 1:N
            X[i] = -1 .+ 2 * rand(dim)
        end
        if dim == 1
            X = sort(X)
        end
        # initial zero speed and forces
        for i = 1:N
            Y[i] = zeros(dim)
            Fcur[i] = zeros(dim)
            Xnew[i] = X[i]
            Ynew[i] = Y[i]
            Fnew[i] = Fcur[i]
        end
        #-- initialization of different quantities
        Ekin = kinetic(Y)
        Epot = energy(X)
        Energy = Epot + Ekin
        #-- Loop over time
        @fastmath @inbounds for n = 1:niter
            #-- save configuration every niterio steps
            if n % niterio == 0
                for i = 1:N
                    for k = 1:dim
```

```
                        TrajectoryX[ Int64(n/niterio), (i-1) * dim + k ] = X[i][k]
                        TrajectoryY[ Int64(n/niterio), (i-1) * dim + k ] = Y[i][k]
                    end
                end
            end
            #--- update positions and speeds
            Epot = update!( X, Y, Fcur, Xnew, Ynew, Fnew, Epot, acceptrate)

        end
        ## Post-processing
        print("Percentage of rejected steps: ", 1 .- acceptrate/niter, "\n")
        # Write the data in text files - Whole trajectory sample
        writedlm(@sprintf("positions-%i",runid), TrajectoryX, " ")
        writedlm(@sprintf("velocities-%i",runid), TrajectoryY, " ")
    end # function HMC()
#
end # @everywhere

### Main part - runs only on main Julia process.
Nprocs = nprocs()
print("Number of processes is ",Nprocs,".\n")
print("Time of the simulation is ", T, ".\n")
print("Number of time steps is ", T/dt ,".\n")
## Launching computations on Nprocs parallel processes.
output = @time pmap(HMC,1:Nprocs)
### EOF
```

References

1. Anderson, G.W., Guionnet, A., Zeitouni, O.: An Introduction to Random Matrices. Cambridge Studies in Advanced Mathematics, vol. 118. Cambridge University Press, Cambridge (2010)
2. Bardenet, R., Hardy, A.: Monte Carlo with determinantal point processes. arXiv:1605.00361v1 (2016)
3. Berman, R.J.: On large deviations for Gibbs measures, mean energy and Gamma-convergence. arXiv:1610.08219v1 (2016)
4. Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., Stuart, A.: Optimal tuning of the hybrid Monte Carlo algorithm. Bernoulli **19**(5A), 1501–1534 (2013)
5. Blanc, X., Lewin, M.: The crystallization conjecture: a review. EMS Surv. Math. Sci. **2**(2), 225–306 (2015)
6. Bolley, F., Chafï, D., Fontbona, J.: Dynamics of a planar Coulomb gas. Ann. Appl. Probab. arXiv:1706.08776v3 (2017)
7. Bou-Rabee, N., Eberle, A., Zimmer, R.: Coupling and convergence for Hamiltonian Monte Carlo. arXiv:1805.00452v1 (2018)
8. Bou-Rabee, N., Sanz-Serna, J.M.: Geometric integrators and the Hamiltonian Monte Carlo method. arXiv:1711.05337v1 (2017)
9. Bouchard-Côté, A., Vollmer, S.J., Doucet, A.: The bouncy particle sampler: a nonreversible rejection-free Markov chain Monte Carlo method. J. Am. Stat. Assoc. **113**(522), 855–867 (2018)
10. Brooks, S., Gelman, A., Jones, G.L., Meng, X.L. (eds.): Handbook of Markov Chain Monte Carlo. Chapman Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton (2011)
11. Brosse, N., Durmus, A., Moulines, É., Sabanis , S.: The tamed unadjusted Langevin algorithm arXiv:1710.05559v2 (2017)
12. Chafaï, D., Hardy, A., Maïda, M.: (2018) Concentration for Coulomb gases and Coulomb transport inequalities. J. Funct. Anal. arXiv:1610.00980v3
13. Chafaï, D., Lehec, J.: On Poincaré and logarithmic Sobolev inequalities for a class of singular Gibbs measures. arXiv:1805.00708v2 (2018)
14. Chafaï, D., Gozlan, N., Zitt, P.-A.: First-order global asymptotics for confined particles with singular pair repulsion. Ann. Appl. Probab. **24**(6), 2371–2413 (2014)
15. Chafaï, D., Péché, S.: A note on the second order universality at the edge of Coulomb gases on the plane. J. Stat. Phys. **156**(2), 368–383 (2014)
16. Chafaï, D., Saff, E.: Aspects of an Euclidean log-gas. Work in progress (2018)
17. Dalalyan, A., Riou-Durand, L.: On sampling from a log-concave density using kinetic Langevin diffusions. arXiv:1807.09382v1 (2018)
18. Decreusefond, L., Flint, I., Vergne, A.: Vergne: a note on the simulation of the Ginibre point process. J. Appl. Probab. **52**(4), 1003–1012 (2015)
19. Duane, S., Kennedy, A., Pendleton, B.J., Roweth, D.: Hybrid Monte Carlo. Phys. Lett. B **195**(2), 216–222 (1987)
20. Dubach, G.: Powers of Ginibre Eigenvalues. arXiv:1711.03151v2 (2017)

21. Dumitriu, I., Edelman, A.: Matrix models for beta ensembles. J. Math. Phys. **43**(11), 5830–5847 (2002)
22. Duncan, A.B., Lelièvre, T., Pavliotis, G.A.: Variance reduction using nonreversible Langevin samplers. J. Stat. Phys. **163**, 457–491 (2016)
23. Durmus, A., Moulines, E., Saksman, E.: On the convergence of Hamiltonian Monte Carlo. arXiv:1705.00166v1 (2017)
24. Edelman, A., Rao, N.R.: Random matrix theory. Acta Numer. **14**, 233–297 (2005)
25. Erdős, L., Yau, H.-T.: A dynamical approach to random matrix theory. Courant Lecture Notes in Mathematics, vol. 28. American Mathematical Society, Providence, RI (2017)
26. Ezawa, Z.E.: Quantum Hall Effects. Field Theoretical Approach and Related Topics, 2nd edn. World Scientific Publishing Co. Pt. Ltd., Hackensack, NJ (2008)
27. Fathi, M., Homman, A.-A., Stoltz, G.: Error analysis of the transport properties of Metropolized schemes. ESAIM Proc. Surv. **48**, 341–363 (2015)
28. Forrester, P .J.: Log-Gases and Random Matrices. London Mathematical Society Monographs Series, vol. 34. Princeton University Press, Princeton, NJ (2010)
29. Forrester, P.J.: Analogies between random matrix ensembles and the one-component plasma in two-dimensions. Nuclear Phys. B **904**, 253–281 (2016)
30. García-Zelada, D.: A large deviation principle for empirical measures on Polish spaces: application to singular Gibbs measures on manifolds. arXiv:1703.02680v2 (2017)
31. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations, Springer Series in Computational Mathematics, vol. 31. Springer Science Business Media, Berlin (2006)
32. Hardy, A.: Polynomial ensembles and recurrence coefficients. arXiv:1709.01287v1 (2017)
33. Helms, L.L.: Potential Theory, 2nd edn. Springer, London (2014)
34. Hoffman, M.D., Gelman, A.: The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. J. Mach. Learn. Res. **15**, 1593–1623 (2014)
35. Höft, T.A., Alpert, B.K.: Fast updating multipole Coulombic potential calculation. SIAM J. Sci. Comput. **39**(3), A1038–A1061 (2017)
36. Horowitz, A.M.: A generalized guided Monte Carlo algorithm. Phys. Lett. B **268**, 247–252 (1991)
37. Hough, J.B., Krishnapur, M., Peres, Y., Virág, B.: Determinantal processes and independence. Probab. Surv. **3**, 206–229 (2006)
38. Hutzenthaler, M., Jentzen, A., Kloeden, P.E.: Strong convergence of an explicit numerical method for SDEs with nonglobally Lipschitz continuous coefficients. Ann. Appl. Probab. **22**(4), 1611–1641 (2012)
39. Jiang, T., Qi, Y.: Spectral radii of large non-Hermitian random matrices. J. Theoret. Probab. **30**(1), 326–364 (2017)
40. Jones, A., Leimkuhler, B.: Adaptive stochastic methods for sampling driven molecular systems. J. Chem. Phys. **135**(8), 084125 (2011)
41. Kapfer, S.C., Krauth, W.: Cell-veto Monte Carlo algorithm for long-range systems. Phys. Rev. E **94**, 031302 (2016)
42. Kloeden, P.E., Platen, E., Schurz, H.: Numerical Solution of SDE Through Computer Experiments. Universitext. Springer, Berlin (1994)
43. Krishnapur, M., Rider, B., Virág, B.: Universality of the stochastic Airy operator. Commun. Pure Appl. Math. **69**(1), 145–199 (2016)
44. Landkof, N.S.: Foundations of Modern Potential Theory. Springer, New York (1972) (Translated from the Russian by A, p. 180. P. Doohovskoy, Die Grundlehren der mathematischen Wissenschaften, Band)
45. Lavancier, F., Møller, J., Rubak, E.: Determinantal point process models and statistical inference. J. R. Stat. Soc. Ser. B. Stat. Methodol **77**(4), 853–877 (2015)
46. Ledoux, M.: Differential operators and spectral distributions of invariant ensembles from the classical orthogonal polynomials. The continuous case. Electron. J. Probab **9**(7), 177–208 (2004)
47. Lee, Y.T., Vempala, S.S.: Convergence rate of Riemannian Hamiltonian Monte Carlo and faster polytope volume computation. arXiv:1710.06261v1 (2017)
48. Leimkuhler, B., Matthews, C., Stoltz, G.: The computation of averages from equilibrium and nonequilibrium Langevin molecular dynamics. IMA J. Numer. Anal. **36**(1), 13–79 (2015)
49. Lelièvre, T., Nier, F., Pavliotis, G.A.: Optimal non-reversible linear drift for the convergence to equilibrium of a diffusion. J. Stat. Phys. **152**, 237–274 (2013)
50. Lelièvre, T., Rousset, M., Stoltz, G.: Free Energy Computations. A Mathematical Perspective. Imperial College Press, London (2010)
51. Lelièvre, T., Rousset, M., Stoltz, G.: Langevin dynamics with constraints and computation of free energy differences. Math. Comput. **81**(280), 2071–2125 (2012)
52. Lelièvre, T., Stoltz, G.: Partial differential equations and stochastic methods in molecular dynamics. Acta Numer. **25**, 681–880 (2016)

53. Levesque, D., Verlet, L.: On the theory of classical fluids II. Physica **28**(11), 1124–1142 (1962)
54. Levesque, D., Verlet, L.: Computer experiments on classical fluids. III. Time-dependent self-correlation functions. Phys. Rev. A **2**, 2514 (1970)
55. Li, X.H., Menon, G.: Numerical solution of Dyson Brownian motion and a sampling scheme for invariant matrix ensembles. J. Stat. Phys. **153**(5), 801–812 (2013)
56. Mattingly, J., Stuart, A., Higham, D.: Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise. Stoch. Proc. Appl. **101**(2), 185–232 (2002)
57. Mehta, M.L.: Random Matrices, 3rd edn. Pure and Applied Mathematics (Amsterdam), vol. 142. Elsevier/Academic Press, Amsterdam (2004)
58. Milstein, G .N., Tretyakov, M .V.: Stochastic Numerics for Mathematical Physics. Springer Science Business Media, Berlin (2013)
59. Olver, S., Nadakuditi, R .R., Trogdon, T.: Sampling unitary ensembles. Random Matrices Theory Appl. **4**(1), 1550002–22 (2015)
60. Rider, B.: A limit theorem at the edge of a non-Hermitian random matrix ensemble. Random matrix theory. J. Phys. A **36**(12), 3401–3409 (2003)
61. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer Texts in Statistics, 2nd edn. Springer, New York (2004)
62. Roberts, G.O., Rosenthal, J.S.: Optimal scaling for various Metropolis-Hastings algorithms. Stat. Sci. **16**(4), 351–367 (2001)
63. Roberts, G.O., Tweedie, R.L., et al.: Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli **2**(4), 341–363 (1996)
64. Rossky, P .H., Doll, J .D., Friedman, H .L.: Brownian dynamics as smart Monte Carlo simulation. J. Chem. Phys. **69**, 4628 (1978)
65. Saff, E.B, Totik, V.: Logarithmic Potentials with External Fields. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 316, Springer, Berlin (1997) (Appendix B by Thomas Bloom)
66. Scardicchio, A., Zachary, C .E., Torquato, S.: Statistical properties of determinantal point processes in high-dimensional Euclidean spaces. Phys. Rev. E (3) **79**(4), 041108 (2009). 19
67. Serfaty, S.: Coulomb Gases and Ginzburg-Landau Vortices. Zurich Lectures in Advanced Mathematics. Euro. Math. Soc. (EMS), Zürich (2015)
68. Serfaty, S.: Systems of points with Coulomb interactions. arXiv:1712.04095v1 (2017)
69. Smale, S.: Mathematical problems for the next century. Math. Intell. **20**(2), 7–15 (1998)
70. Smale, S.: Mathematical problems for the next century. In: Arnold, V., Atiyah, M., Lax, P., Mazur, B. (eds.) Mathematics: Frontiers and Perspectives, pp. 271–294. Am. Math. Soc., Providence, RI (2000)
71. Stoltz, G., Trstanova, Z.: Stable and accurate schemes for Langevin dynamics with general kinetic energies. arXiv:1609.02891v1 (2016)
72. Vanetti, P., Bouchard-Côté, A., Deligiannidis, G., Doucet, A.: Piecewise-deterministic Markov Chain Monte Carlo. arXiv:1707.05296v1 (2018)
73. Verlet, L.: Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Phys. Rev. **159**(98), 9 (1967)