

Monte Carlo techniques for real-time quantum dynamics

Mark R. Dowling ^{*}, Matthew J. Davis, Peter D. Drummond, Joel F. Corney

*ARC Centre of Excellence for Quantum-Atom Optics, Department of Physics, School of Physical Sciences,
University of Queensland, Brisbane, Qld 4072, Australia*

Received 1 July 2005; received in revised form 3 February 2006; accepted 17 May 2006
Available online 7 July 2006

Abstract

The stochastic-gauge representation is a method of mapping the equation of motion for the quantum mechanical density operator onto a set of equivalent stochastic differential equations. One of the stochastic variables is termed the “weight”, and its magnitude is related to the importance of the stochastic trajectory. We investigate the use of Monte Carlo algorithms to improve the sampling of the weighted trajectories and thus reduce sampling error in a simulation of quantum dynamics. The method can be applied to calculations in real time, as well as imaginary time for which Monte Carlo algorithms are more-commonly used. The Monte-Carlo algorithms are applicable when the weight is guaranteed to be real, and we demonstrate how to ensure this is the case. Examples are given for the anharmonic oscillator, where large improvements over stochastic sampling are observed.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Quantum dynamics; Monte Carlo; Metropolis algorithm; Branching algorithm; Stochastic gauges; Bose–Einstein condensation

1. Introduction

The problem of calculating quantum dynamics is of fundamental interest in theoretical physics [1]. It is a notoriously difficult problem due to the fact that the dimension of the appropriate Hilbert space scales exponentially with the number of modes in the system. This is enormous for most realistic physical systems and prevents the complete representation of the evolving many-body quantum state as a numerical state-vector. Successful quantum simulation methods can thus only hope to sample the quantum evolution, to some finite precision, by use of stochastic methods.

Such quantum Monte Carlo (QMC) techniques have a long history in first-principles, microscopic calculations of thermal equilibrium and ground states in quantum systems [2,3]. Certain classes of QMC methods, such as diffusion or Green’s function approaches [4] (projector methods) are restricted to calculation of ground state properties. Methods based on path integrals [5] are a simulation through imaginary time, and

^{*} Corresponding author. Tel.: +61 7 33467347.

E-mail address: dowling@physics.uq.edu.au (M.R. Dowling).

can calculate correlations at any temperature. However, for real-time calculations (i.e. dynamics), path integral QMC methods become difficult because of sign or phase problems [6,7].

An alternative approach is provided by phase-space representations [8–10], which can be used to map quantum dynamics to a set of equivalent stochastic differential equations. The number of phase-space equations scale polynomially with the number of modes, allowing computationally tractable simulations. Phase-space methods have proved useful in the past for simulating quantum dynamics, particularly in the field of quantum optics [11–13]. A natural extension of these techniques is to the field of degenerate quantum gases, where the interacting particles are atoms or molecules rather than photons [14,15]. Recently it has been discovered that Fermi gases can be treated with related techniques [16,17].

The mapping of a quantum problem to phase-space equations is far from unique. This non-uniqueness can be exploited to tailor the form the stochastic equation without affecting the physical, ensemble result. The different choices correspond to different “stochastic gauges”. In this paper, we use this freedom to generate stochastic equations with real weights, which we then sample with Monte Carlo techniques. The real weights avoid the sign or phase problem encountered in other QMC approaches to quantum dynamics. Since the stochastic gauge method is a relatively new technique, we choose to focus here on an especially simple case with known exact solutions, in order to clarify the problems and advantages of this real-weight approach.

The example we choose is the single-mode quantum anharmonic oscillator. Even though this system has an exact solution, it displays highly nontrivial behaviour when treated as a stochastic problem. In particular, this problem is of interest as it is relevant to current experiments on the dynamics of ultra-cold atoms trapped in optical lattices. These are described rather accurately by the so-called Bose–Hubbard model, which reduces to the type of single-mode theory treated here in the Mott-insulator limit in which inter-well tunneling is suppressed. We simulate the decay of coherence due to phase-diffusion, a physical effect that has already been experimentally observed in recent experiments with Bose–Einstein condensates [18]. The present paper focuses on this relatively straightforward case in order to demonstrate the important principles behind first-principles quantum dynamical simulation in real time. Due to the linear scaling of these methods with increasing numbers of modes, we expect that the same basic ideas will apply to multi-mode or multi-well situations where there are no known exact solutions in general.

This paper is organised as follows. In Section 2, we review the stochastic-gauge representation [19] and motivate the use of Monte Carlo techniques for sampling the weighted stochastic trajectories. In Section 3, we introduce the anharmonic oscillator as a simple model for interacting quantum dynamics, and describe the design of a gauge that results in real weights. We then simulate using this gauge with regular stochastic sampling and find that the sampling becomes poor over short time scales. In Section 4, we review the Metropolis–Hastings algorithm [20–22] and describe its application to stochastic-gauge simulations. We then demonstrate improvements over the usual stochastic sampling for the same type of gauge. In Section 5, we describe an alternative scheme, based on stochastic sampling, but with a branching algorithm for efficient handling of the weights. Finally, in Section 6 we discuss the advantages and disadvantages of each Monte Carlo method in the context of future applications.

2. The Stochastic-gauge representation

The *stochastic-gauge* (or *gauge-P*) representation is a generalisation of the positive- P ($+P$) representation [9,10], where the density operator is expanded as a positive distribution over an off-diagonal, over-complete basis set of coherent states. In a $+P$ representation of an M mode bosonic quantum system, there are two complex phase-space variables (α_k, β_k) for each mode, $k = 1 \dots M$. Throughout this paper the $+P$ variables $\vec{\alpha} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ are referred to as *mode variables*. The essential difference between the $+P$ and the stochastic-gauge representation is that the latter is defined over a quantum phase space with an additional dimension termed the *weight* Ω , such that the total phase space is of complex dimension $2M + 1$. For a complete description of the method we refer the reader to Ref. [19], however we briefly summarise the main features of the method below.

The procedure for calculating quantum dynamics using the stochastic-gauge representation is similar to that using the $+P$ representation and involves deriving, via a Fokker–Planck equation, a set of stochastic differential equations equivalent to the original quantum master equation. The most general quantum-dynamical evolution may be written as a master equation of the form [10]

$$\frac{\partial \hat{\rho}}{\partial t} = L[\hat{\rho}], \tag{1}$$

where L is a Liouville superoperator (e.g. $L[\hat{\rho}] = -i\hbar[\hat{H}, \hat{\rho}]$ for unitary evolution). To calculate the quantum dynamics using the stochastic-gauge representation we expand the density operator in an over-complete basis set as

$$\hat{\rho} = \int d^{4M} \vec{\alpha} d^2 \Omega G(\vec{\alpha}, \Omega) \hat{\Lambda}(\vec{\alpha}, \Omega), \tag{2}$$

where

$$\hat{\Lambda}(\vec{\alpha}, \Omega) = \Omega ||\alpha\rangle \langle \beta^* || \exp[-\alpha \cdot \beta]$$

is the kernel, $G(\vec{\alpha}, \Omega)$ is the (non-unique) gauge distribution function, and

$$||\alpha\rangle = \exp[\alpha \cdot \hat{a}^\dagger] |0\rangle \tag{3}$$

is a multi-mode Bargmann coherent state [10], where \hat{a}^\dagger is a vector of creation operators for the M modes, and $|0\rangle$ is the vacuum state.

By use of operator identities for creation and annihilation operators acting on the kernel and subsequent integration by parts (with the assumption that boundary terms vanish), it is possible to show that any master equation involving only two-body terms is equivalent to a gauge distribution function evolving according to a positive-definite Fokker–Planck equation. The dynamical moments may therefore be obtained by evolving an equivalent set of stochastic differential equations (SDEs) and taking stochastic averages of an appropriate product of stochastic variables [23].

At this point it is possible to add arbitrary terms to the Fokker–Planck differential operator that give zero when acting on the kernel. These terms cannot affect the quantum averages, but add stochastic gauges – arbitrary functions on phase space – to the drift part of the corresponding SDEs [19]. The central result is easy to state. For an M mode quantum system with $+P$ Ito equations of the form

$$\dot{\vec{\alpha}} = \vec{A}^{(+)} + \mathbf{B}^{(+)} \vec{\xi}, \tag{4}$$

where $\vec{A}^{(+)} = \vec{A}^{(+)}(\vec{\alpha})$ and $\mathbf{B}^{(+)} = \mathbf{B}^{(+)}(\vec{\alpha})$ are the positive- P drift vector and diffusion matrix, respectively, then the stochastic-gauge equations for the system are

$$\dot{\vec{\alpha}} = \vec{A}^{(+)} + \mathbf{B}^{(+)} (\vec{\xi} - \vec{g}), \tag{5}$$

$$\dot{\Omega} = \Omega \vec{g}^T \vec{\xi}. \tag{6}$$

Here $\vec{g} = \vec{g}(\vec{\alpha})$ is a vector of *stochastic gauges* and $\vec{\xi}$ is a vector of Gaussian noises with the correlations

$$\langle \xi_i(t) \xi_j(t') \rangle = \delta_{ij} \delta(t - t'). \tag{7}$$

The stochastic gauges \vec{g} can be used to modify the deterministic evolution of the stochastic trajectories and are therefore called *drift gauges*.

Also possible are *diffusion gauges* which arise from the non-unique factorisation of drift matrix \mathbf{D} appearing in the Fokker–Planck equation into a noise matrix \mathbf{B} for the SDEs, where

$$\mathbf{D} = \mathbf{B} \mathbf{B}^T. \tag{8}$$

Diffusion gauges can be used, for example, to “squeeze” the noise between stochastic variables to improve sampling [24]. More general types of diffusion gauge are possible in the full stochastic-gauge formalism [19,25].

For a single mode, quantum-dynamical averages of normally ordered products of creation and annihilation operators are calculated as stochastic averages in the following manner

$$\langle (\hat{a}^\dagger)^m \hat{a}^n \rangle_{\text{QM}} = \frac{\langle \Omega \beta^m \alpha^n + (\Omega \beta^n \alpha^m)^* \rangle_{\text{stoch}}}{\langle \Omega + \Omega^* \rangle_{\text{stoch}}}. \tag{9}$$

In principle any gauge that does not introduce boundary terms on partial integration will reproduce the exact quantum averages in the limit that an infinite number of trajectories are simulated, and so all gauges represent

the same physics. However, in practice we may only simulate a finite number of trajectories due to limited computing resources, and so we would like to choose a gauge that gives rise to the most compact phase-space distribution G possible. A narrower distribution means that fewer stochastic trajectories need to be sampled to obtain quantum averages with a given precision. The situation is similar to classical or quantum electrodynamics, where a judicious choice of gauge simplifies the solution of certain problems.

2.1. Monte Carlo techniques for the weight variable

The motivation for this work is that in the stochastic gauge representation, stochastic trajectories are often generated with weights that vary over many orders of magnitude as illustrated in Fig. 1. From Eq. (9) we can see that the trajectories with relatively high weight contribute more to the overall stochastic averages than those with relatively low weight.

The situation is analogous to path-integral calculations of quantum averages in imaginary time (thermal quantum averages), where quantum Monte Carlo techniques have long been used to sample the weight parameter more efficiently [26,4]. In the standard stochastic gauge prescription the stochastic trajectories are generated randomly. If the weight parameter can be interpreted as a probability, then this suggests that more sophisticated Monte Carlo techniques can be used to efficiently sample high-weight trajectories, thus improving the sampling of physical averages.

In this paper we focus on real-time dynamics, although the same techniques apply to imaginary-time stochastic gauge calculations [17]. Previously, Monte Carlo techniques have had only limited success with real-time quantum dynamics, where path-integral approaches are plagued by sign problems due to the rapidly-oscillating phase [3,27].

However, there is a complication – in the stochastic-gauge representation the weight is complex in general. In principle it is possible to apply Monte Carlo techniques to problems with a complex weight – by treating the modulus of the weight as the ‘importance’. In practice, however, sign problems are encountered when the phase of the weight eventually becomes evenly distributed around the unit circle in the complex plane. In order to strictly interpret the weight as a probability distribution, we must ensure that our choice of gauge leads to a weight parameter that is real. This means that the gauge functions we introduce must also be real.

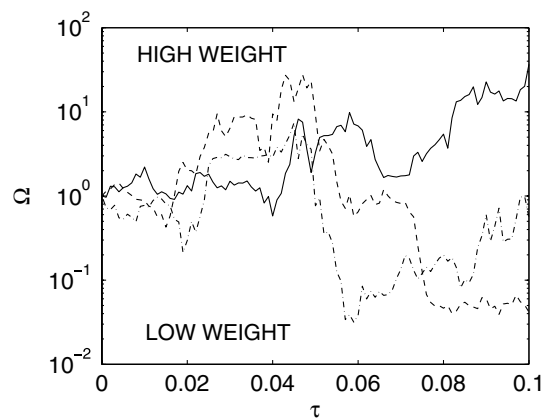


Fig. 1. Illustration of the spreading of weights in a stochastic gauge simulation for the anharmonic oscillator. These three trajectories of Ω were generated using randomly chosen noises for a real-gauge simulation of the anharmonic oscillator ($A = 2, \lambda = 1/2$), as described in Section 3.1, with mean atom number $\bar{n} = 100$. The total simulation time was $\tau_{\text{tot}} = 1/\sqrt{\bar{n}} = 0.1$ and the time step used was $\Delta\tau = \tau_{\text{tot}}/10^3 = 10^{-4}$. Note that the relative weights of the trajectories vary in time so the Metropolis algorithm can only be targeted to sample the distribution at a single chosen time. The branching algorithm clones trajectories continuously in time in proportion to their weight so as to obtain continuous-time samples of the distribution.

3. Example: The Kerr anharmonic oscillator

In this section, we introduce a single-mode boson model for quantum nonlinear dynamics – the Kerr anharmonic oscillator. The Hamiltonian is

$$\hat{H} = \hbar\omega_0 \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right) + \frac{1}{2} \kappa \hat{a}^{\dagger 2} \hat{a}^2 = \hbar\omega_0 \left(\hat{n} + \frac{1}{2} \right) + \frac{1}{2} \kappa \hat{n}(\hat{n} - 1), \tag{10}$$

where $\hat{n} = \hat{a}^\dagger \hat{a}$ is the number operator. Traditionally this Hamiltonian has been used to describe the Kerr effect in nonlinear optics. It has received renewed interest recently, as it is the restriction to a single site of the Bose–Hubbard model. This has been shown to describe ultra-cold bosonic atoms in an optical lattice [28], a topic of recent theoretical and experimental interest [29,30].

Throughout this paper we work in the interaction picture with

$$H_{\text{int}} = \frac{1}{2} \kappa \hat{a}^{\dagger 2} \hat{a}^2. \tag{11}$$

A number state is an eigenstate of this Hamiltonian and evolves according to

$$|n(\tau)\rangle = \exp(-in(n-1)\tau/2)|n(0)\rangle, \tag{12}$$

where we have introduced the dimensionless time $\tau = \kappa t/\hbar$. Hence, any initial state can be decomposed into number states and an exact solution found. In particular, for an initial coherent state $|\psi(0)\rangle = |\alpha\rangle$, the solution is

$$|\psi(\tau)\rangle = e^{-|\alpha|^2/2} \sum_n \frac{\alpha^n}{\sqrt{n!}} \exp(-in(n-1)\tau/2)|n\rangle. \tag{13}$$

As this model has an exact solution and includes the nonlinearity that is a feature of models such as the Bose–Hubbard Hamiltonian, it forms an excellent testing ground for quantum simulation methods [31].

An important quantum feature of this Hamiltonian is that given an initial coherent state $|\alpha\rangle$ with mean boson number $\bar{n} = |\alpha|^2$, the dynamics display a series of collapses and revivals. Defining the quadrature variables

$$\hat{X} = (\hat{a} + \hat{a}^\dagger)/2, \quad \hat{Y} = (\hat{a} - \hat{a}^\dagger)/2i, \tag{14}$$

we find there are three characteristic timescales for anharmonic oscillator dynamics. The quadratures initially undergo oscillations with period $\tau_{\text{osc}} \sim O(1/\bar{n})$, which are damped to zero over a time of $\tau_{\text{coll}} \sim O(1/\sqrt{\bar{n}})$. However, the oscillations revive at time $\tau_{\text{rev}} \sim O(1)$, which for a large mean boson number can be many times the collapse time.

Because the period of oscillation of the quadratures can be very short for large mean atom number, and because we are most interested in the envelope of the collapse, we choose to perform the calculations in a rotating frame. The angular frequency of the rotation is equal to the mean atom number, and the X -quadrature, whose exact solution is

$$\langle \hat{X}(\tau) \rangle = \sqrt{\bar{n}} e^{\bar{n}(\cos(\tau)-1)} \cos(\bar{n}(\sin(\tau) - \tau)), \tag{15}$$

collapses dynamically for large mean atom number according to:

$$\langle \hat{X}(\tau) \rangle \simeq \sqrt{\bar{n}} e^{-\bar{n}\tau^2/2}. \tag{16}$$

For the anharmonic oscillator Hamiltonian the stochastic-gauge Stratonovich SDEs are

$$\dot{\alpha} = -i\alpha^2\beta + i\alpha/2 + \sqrt{-i}\alpha[\cosh(A)(\xi_x - g_x) - i \sinh(A)(\xi_\beta - g_\beta)], \tag{17}$$

$$\dot{\beta} = i\beta^2\alpha - i\beta/2 + \sqrt{i}\beta[\cosh(A)(\xi_\beta - g_\beta) + i \sinh(A)(\xi_x - g_x)], \tag{18}$$

$$\dot{\Omega} = S_\Omega + \Omega \sum_{j=x,\beta} g_j \xi_j. \tag{19}$$

Here A is a diffusion gauge that we choose to be constant in this paper, $g_j, j = \alpha, \beta$, are drift gauges to be chosen subsequently, and ξ_j are dimensionless Gaussian noise terms chosen so that:

$$\langle \xi_i(\tau) \xi_j(\tau') \rangle = \delta_{ij} \delta(\tau - \tau'). \quad (20)$$

The term S_Ω is a Stratonovich correction factor which depends on the gauge choice.

For better numerical performance we choose to work with *log variables*:

$$\theta = (1/2) \log(\alpha\beta), \quad \phi = (1/2i) \log(\alpha/\beta), \quad \omega = \log(\Omega). \quad (21)$$

These obey the following Stratonovich SDEs:

$$\dot{\theta} = \frac{1}{2} e^{-A} (\xi_1 - g_1 - i(\xi_2 - g_2)), \quad (22)$$

$$\dot{\phi} = -e^{2\theta} + \frac{1}{2} - \frac{1}{2} e^A (\xi_1 - g_1 + i(\xi_2 - g_2)), \quad (23)$$

$$\dot{\omega} = S_\omega + \sum_{j=1,2} g_j \xi_j, \quad (24)$$

where we have defined the linearly transformed noises $\xi_1 = (\xi_\alpha + \xi_\beta)/\sqrt{2}$ and $\xi_2 = (\xi_\alpha - \xi_\beta)/\sqrt{2}$, which obey the same statistics, and similarly $g_1 = (g_\alpha + g_\beta)/\sqrt{2}$ and $g_2 = (g_\alpha - g_\beta)/\sqrt{2}$. We note here that, unlike in the corresponding classical oscillator equations, both $\theta = \theta_X + i\theta_Y$ and $\phi = \phi_X + i\phi_Y$ are intrinsically complex. The notation of X and Y superscripts to refer to the real and imaginary parts of a phase-space variable, respectively, is used throughout this paper.

3.1. Choice of gauge

In this section, we discuss possible choices of diffusion gauge for the anharmonic oscillator with a view to using Monte Carlo techniques to sample the weight.

In Ref. [32] Drummond and Deuar investigated the following drift-gauge choice for the anharmonic oscillator

$$g_1 = ig_2 = ie^{A+2\theta_X} \sin(2\theta_Y). \quad (25)$$

They found that this gauge extended simulation times by many orders of magnitude for the same number of stochastic trajectories. In particular, they could simulate well past the collapse time for an initial coherent state.

As noted in Section 2.1, for the Monte Carlo methods to be successful it is highly desirable to have a real weight. Unfortunately the above choice of gauge is complex and thus leads to complex weights. We have trialled the use of Monte Carlo techniques based on using the modulus of the weight, but found these to be unsuccessful due to the phase problem described in Section 2.1.

In order to have real weights, we would like to design a real gauge with similar properties to the above complex gauge to give a similar extension of simulation time. The reason that the complex gauge Eq. (25) improves simulation times is that it removes a driving term from the imaginary part of the $+P$ ϕ equation. The presence of this term forces trajectories to diverge to infinity in a non-classical direction in phase-space ($\phi_Y \rightarrow \pm\infty$), thus resulting in large sampling errors. At the same time the equations for θ and the real part of ϕ are unaffected due to a cancellation between g_1 and g_2 . We could remove the offending term from the ϕ_Y equation with a single real gauge g_2 , however this would necessarily appear in the θ_Y equation leading to an instability that causes poor sampling.

Thus it seems that a drawback to choosing real gauges is that there is substantially less control over the stochastic trajectories. This could have been anticipated because, in general, with a complex gauge we have as many gauge degrees of freedom as we do real phase space dimensions, $4M$ (excluding the weight dimensions). However with real gauges we only have half as many gauge degrees of freedom, $2M$, meaning that we cannot independently control the drift in the real and imaginary components of each of the mode variables.

An additional source of gauge freedom is obtained from choosing a non-square noise matrix in deriving the SDEs. This freedom was pointed out in [19] but not explored. Specifically, it is possible to take the noise matrix to be of the general form

$$\mathbf{B} = [\mathbf{B}_0, \mathbf{Q}], \tag{26}$$

where \mathbf{B}_0 is a square ($2M \times 2M$) noise matrix such that $\mathbf{B}_0 \mathbf{B}_0^T = \mathbf{D} - \mathbf{Q} \mathbf{Q}^T$, and \mathbf{Q} is a $2M \times W$ matrix, where W is any positive integer, whose entries are arbitrary complex functions. This choice reproduces the correct moments in the limit of a large number of stochastic trajectories, but introduces more than the minimum number of noise terms into the stochastic equations. Naïvely this could be expected to lead to worse sampling errors. However, this additional gauge freedom allows us to overcome the restrictions of the standard real gauges and improve the sampling overall.

For the anharmonic oscillator we choose

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ \lambda & i\lambda \end{bmatrix}, \tag{27}$$

in the (θ, ϕ) variables, where λ could be an arbitrary complex function on phase space, although we choose it to be constant in our example. Here \mathbf{Q} satisfies $\mathbf{Q} \mathbf{Q}^T = 0$ so the other noise terms are unaffected. In doing so we have added a term of the form $\lambda(\xi_3 + i\xi_4)$ to the equation of motion for ϕ , Eq. (23). An equivalent way of understanding this additional noise is that due to the analytic nature of the stochastic gauge kernel we have

$$\lambda \left(\frac{\partial^2}{\partial \phi_x^2} + \frac{\partial^2}{\partial \phi_y^2} \right) A(\vec{x}, \Omega) = 0. \tag{28}$$

Thus we are free to add $\lambda(\partial^2/\partial \phi_x^2 + \partial^2/\partial \phi_y^2)$ to the Fokker–Planck differential operator without affecting the physical moments.

We are now able to introduce additional gauges to the $\dot{\phi}$ equation in the manner described in [19] so that the extra term in the $\dot{\phi}$ equation becomes

$$\lambda((\xi_3 - g_3) + i(\xi_4 - g_4)), \tag{29}$$

and the gauges g_3 and g_4 enter the weight equation in the same way as the other drift gauges. The extra noise allows us to control the ϕ_Y divergence using only a real gauge, without affecting the θ_Y equation. Specifically, we choose

$$g_1 = g_2 = g_3 = 0, \quad g_4 = -\frac{e^{2\theta_x}}{\lambda} \sin(2\theta_Y). \tag{30}$$

The final SDEs that we focus on sampling for the rest of this paper are summarised as

$$\dot{\theta} = \frac{1}{2} e^{-A} (\xi_1 - i\xi_2), \tag{31}$$

$$\dot{\phi} = -e^{2\theta_x} \cos(2\theta_Y) + \frac{1}{2} - \frac{1}{2} (e^A (\xi_1 + i\xi_2) - 2\lambda(\xi_3 + i\xi_4)), \tag{32}$$

$$\dot{\omega} = S_\omega - \frac{e^{2\theta_x}}{\lambda} \sin(2\theta_Y) \xi_4, \tag{33}$$

where $S_\omega = -e^{4\theta_x} \sin^2(2\theta_Y)/2\lambda^2$ is the Stratonovich correction in the weight equation. Note that although we choose $g_3 = 0$, it is still necessary to include the noise ξ_3 for the mapping to be exact. Of course there are other possible choices of real gauges, but we have found this combination to be well-suited to illustrating the improvements possible with Monte Carlo sampling.

3.2. Results

In Fig. 2, we illustrate stochastic sampling of the above SDEs. We begin with an initial coherent state with mean atom number $\bar{n} = 100$, and simulate to a final time of $\tau = 1/\sqrt{\bar{n}} = 0.1$, which is of the order of the collapse time. The error bars plotted are calculated by assuming approximately Gaussian statistics – i.e.

$$\Delta\langle f \rangle = \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{n - 1}}, \tag{34}$$

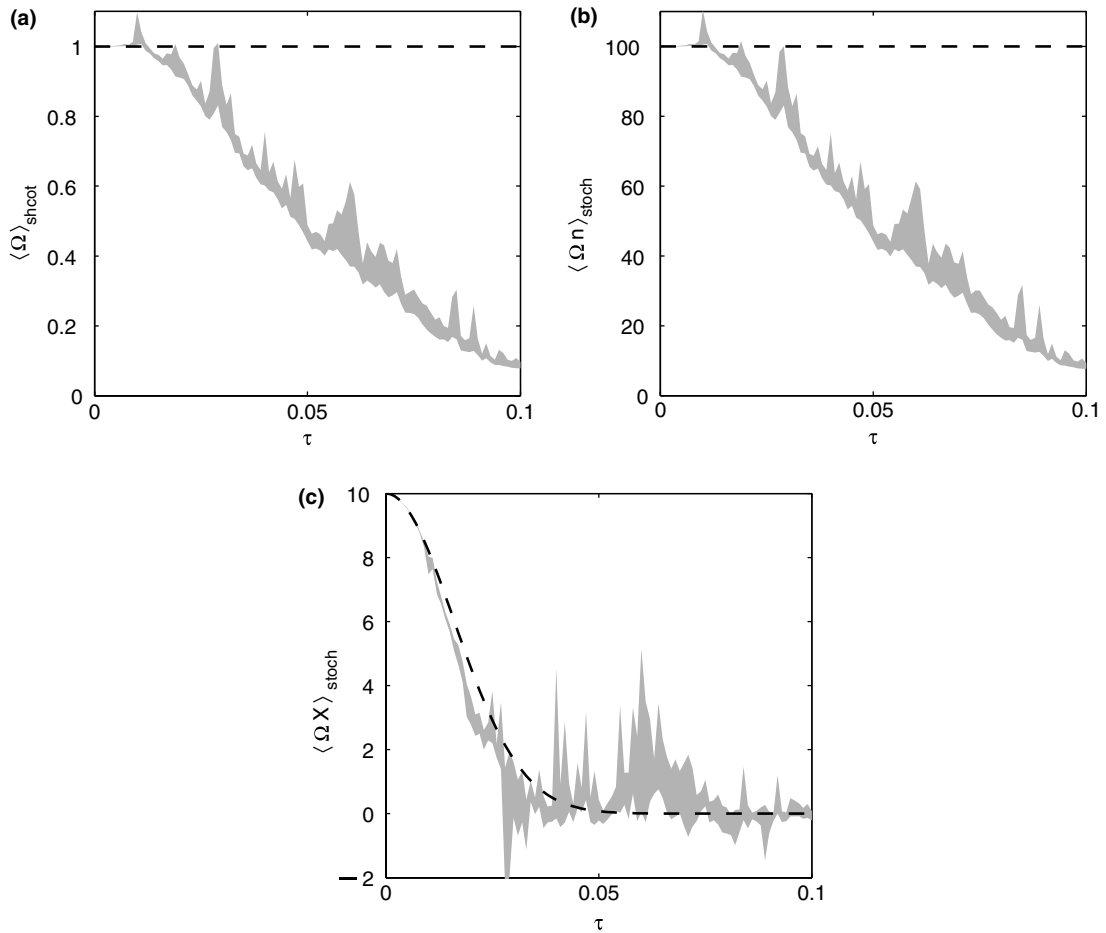


Fig. 2. Real-gauge simulation of anharmonic oscillator dynamics for an initial coherent state with mean atom number $\bar{n} = 100$ ($A = 2, \lambda = 1/2$). (a) Mean weight, (b) mean atom number (corresponding stochastic variable $n = \alpha\beta$) and (c) mean X -quadrature (corresponding stochastic variable $X = (\alpha + \beta)/2$). Averages were carried out over 10^6 stochastic trajectories. The total simulation time was $\tau_{\text{tot}} = 1/\sqrt{\bar{n}} = 0.1$ and the time step used was $\Delta\tau = \tau_{\text{tot}}/10^3 = 10^{-4}$. The shaded region indicates the estimated error in the simulation and the stochastic means are in the centre of the region. The analytic results are shown as dashed lines. The sampling is poor except for short times due to the skewed weight distribution.

where f is the function of the stochastic phase-space variables to be averaged, $\Delta\langle f \rangle$ is the error in the mean, and n is the number of trajectories. Clearly the sampling of the solution is only accurate for short times, perhaps $\tau < 0.01$. In particular, the sampled $\langle \Omega \rangle_{\text{stoch}}$ decays towards zero when it should remain at one for all times in a simulation of unitary dynamics. The decay in mean atom number, which should remain at 100, is almost identical to the decay in mean weight, suggesting that poor sampling of the weights is the cause of the discrepancy. Furthermore, the estimated errors in the means are small despite the fact that they are clearly far from the analytic results, which indicates that either (34) is not a good estimate of the error in the sampled means or that there is some systematic error such as a boundary term.

In principle this undesirable behaviour could be due to neglected boundary terms that are assumed to vanish in the general formulation, as noted in Section 2. Unfortunately there is no analytic method for determining whether or not boundary terms vanish. Numerical signatures of non-vanishing boundary terms, such as stochastic trajectories that diverge to infinity in a finite amount of time, are discussed in [33]. Such trajectories are not observed in our simulations.

One can understand the poor sampling by considering the nature of the distribution of weights. We find that the weight parameter evolves with time towards zero for most trajectories. However, the mean weight

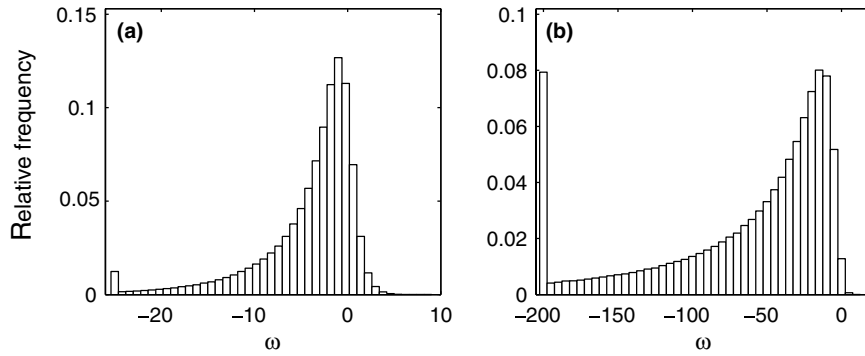


Fig. 3. Histogram of the distribution of the stochastic variable $\omega = \ln \Omega$ at (a) an intermediate time, $\tau = 0.025$, and (b) the end ($\tau = 0.1$) of the simulation plotted in Fig. 2. The last bins of (a) at $\omega = -25$ and (b) at $\omega = -200$ include all trajectories with lower weights; the minimum was $\omega \approx -100$ in (a) and $\omega \approx -1700$ in (b). The weights continuously spread out over many orders of magnitude as time progresses. It is the “skewness” of the weight distribution that causes difficulty in stochastic sampling and which we aim to account for by using Monte Carlo techniques.

must be one, and so there must exist a small number of trajectories with large weights. Thus the distribution of weights is skewed towards zero with a long tail extending to large weights. The skewed distribution of weights is illustrated in Fig. 3 where we plot a histogram of the logarithms of the weights at an intermediate time ($\tau = 0.025$) and at the end ($\tau = 0.1$) of the stochastic trajectories of the simulation plotted in Fig. 2. Such distributions are difficult to sample, so it is not surprising that we underestimate the errors in the means as these assume Gaussian statistics. For example only approximately 1 in 10 of the trajectories had $\Omega > e^{-10}$ by the end of the simulation. In other words 9 in 10 trajectories make negligible contribution to the averages and yet are still included in the total number of trajectories in (34). This explains why (34) typically vastly underestimates the error for these simulations.

To further quantify the skewness of the distributions in Fig. 2 we calculated a number of standard measures. Note that it is actually the distribution of $\Omega = \exp(\omega)$ that we are interested in, however we have plotted a histogram of ω because Ω is so peaked that little can be seen from a histogram. The *skewness* of a distribution is its third central moment divided by the cube of its standard deviation ($=0$ for a normal distribution). The skewness was 200 at the intermediate time and 240 at the end. The *kurtosis*, defined as the fourth central moment divided by the fourth power of the standard deviation ($=3$ for a normal distribution), is a measure of how peaked a distribution. The kurtosis was 5.5×10^4 and 7.4×10^4 at the intermediate and end times, respectively. The sampled mean is 0.86 at the intermediate time compared with the median of 0.09, while at the end time the mean is 0.09 compared with median of 5×10^{-19} . Finally, the truncated mean (where the largest and smallest 1% of the samples are discarded) is 0.49 at the intermediate time and 0.001 at the end. These last two measures, the difference between the sampled mean and median and the sensitiveness to discarding the most outlying samples, are perhaps more relevant to how difficult it is to sample the distribution accurately.

Fortunately, Monte Carlo techniques are particularly successful in sampling skewed distributions. Below we describe the application of the Metropolis–Hastings algorithm and a Monte Carlo branching algorithm to our SDEs for the anharmonic oscillator. As discussed in Section 6 the improvements in sampling that we see provide further evidence that the skewed weight distribution, rather than a boundary term, is the cause of the sampling problems in the simulation of Fig. 2.

4. The Metropolis–Hastings algorithm

The Metropolis–Hastings algorithm is a well-known technique for generating samples of a multi-dimensional probability distribution. Here we briefly describe the algorithm and discuss how to estimate errors in quantities derived from such samples.

4.1. Algorithm

The Metropolis algorithm was first described in 1953 [20] and was used to sample a thermal Boltzmann distribution, which can be viewed as a probability distribution over the space of all thermally accessible states of a system. Subsequently, the Metropolis algorithm has been generalised to many types of probability distributions/densities over both discrete and continuous spaces, and there is a vast amount of mathematical literature on Markov Chain Monte Carlo (MCMC) techniques (see e.g. [6,7]).

In this section, we summarise the Metropolis algorithm in a more general form due to Hastings [21]. A well-written introduction to the Metropolis–Hastings algorithm may be found in [22], and here we follow their notation. Formally, the Metropolis algorithm is a MCMC technique for efficiently sampling a probability distribution, $\pi(s)$, where $s \in S$ represents the state of some system, and S is the domain of the distribution known as the state space. To be a true probability distribution $\pi(\cdot)$ must be normalised $\int ds\pi(s) = 1$, however one of the virtues of the algorithm is that the functional form of the distribution need only be known up to a constant factor in order to apply the algorithm.¹

A Markov chain can be thought of as a random walk through state space where the probability of making a particular step depends only on the current location. It is a sequence of points, $s_1, s_2, \dots, s_n \in S$, where the s_i are random variables such that

$$p(s_i | s_1, s_2, \dots, s_{i-1}) = p(s_i | s_{i-1}). \quad (35)$$

The conditional density $p(s_i | s_{i-1})$ is known as the *transition density* of the Markov chain. If there exists an *invariant density*

$$\pi^*(s) = \int ds' \pi^*(s') p(s | s'), \quad (36)$$

that satisfies the condition of *detailed balance*

$$p(s' | s) \pi^*(s) = p(s | s') \pi^*(s'), \quad (37)$$

then it may be shown that the Markov chain *converges* to the distribution $\pi(\cdot)$. More precisely, the elements of the chain $\{s_i | i = b + 1, \dots, n\}$ are unbiased samples from the distribution $\pi^*(\cdot)$ to within some given precision, where $b \geq 0$ is known as the *burn-in* and represents the number of steps required for the chain to converge to within that precision.

The Metropolis–Hastings algorithm solves the problem of determining what transition density to use so as to generate a given invariant distribution. It is concerned with generating samples from some *target density* $\pi(\cdot)$, known apart from a constant factor, and does so by determining a suitable transition density for a Markov chain to converge to this distribution. To do this we require a *candidate generating density*, $q(s, s')$, where $\int q(s, s') ds' = 1$, which selects the next point in the Markov chain. The *acceptance probability* of this step is

$$\alpha(s, s') = \min \left(\frac{\pi(s') q(s', s)}{\pi(s) q(s, s')}, 1 \right). \quad (38)$$

The Metropolis–Hastings algorithm can be summarised as follows:

- (1) repeat for $i = 1, 2, \dots, n$
- (2) generate s' from $q(s_i, \cdot)$ and a uniform random variable u between 0 and 1.
- (3) if $u \leq \alpha(s_i, s')$ then set $s_{i+1} = s'$;
- (4) else set $s_{i+1} = s_i$;
- (5) return $\{s_i | i = 1, 2, \dots, n\}$

The Metropolis–Hastings algorithm generates a Markov chain in the state space with a transition density $p_{mh}(s' | s) = q(s, s') \alpha(s, s')$. When a proposed move is rejected the chain remains where it is. This choice of tran-

¹ E.g. the partition function $\mathcal{Z} = \sum_{s \in S} e^{-E(s)/k_B T}$, where $E(s)$ is the energy of the system in state s , need not be known in order to sample the Boltzmann distribution, where the probability that the system is in state s is given by $P(s) = e^{-E(s)/k_B T} / \mathcal{Z}$.

sition density ensures that the condition of detailed balance is satisfied and hence the Markov chain converges to the required target density. The algorithm generates a Markov chain that finds regions in state space where the probability distribution is peaked and samples these region with the correct frequency.

The efficiency of the algorithm is dependent on the choice of candidate generating function. Indeed the optimal choice of $q(s, s')$ for a particular target density remains an active area of research today. An important special case occurs if the candidate generating distribution is symmetric, $q(s, s') = q(s', s)$, and so

$$\alpha(s, s') = \min \left(\frac{\pi(s')}{\pi(s)}, 1 \right). \tag{39}$$

This was Metropolis’s original formulation which was generalised to the case of an asymmetric candidate generating function in 1970 by Hastings [21].

4.2. Estimating sampling error

In general we wish to determine the weighted average over all possible states of some observable of the system

$$\langle O \rangle_\pi = \int ds \pi(s) O(s).$$

Because the Metropolis algorithm produces a set samples of the distribution $\pi(\cdot)$, $\{s_i | i = b + 1 \dots b + n\}$, such averages can be estimated as

$$\langle O \rangle_\pi \simeq \frac{\sum_{i=b+1}^{b+n} O(s_i)}{n}. \tag{40}$$

The statistical uncertainty in such a quantity can be difficult to estimate because the samples are generally correlated with one another, as the samples produced by the Metropolis algorithm are not independent. One way to obtain less correlated samples is to only take every g th point in the chain after the burn-in for the purpose of calculating averages, where g is known as the *gap*. However, this does not usually produce more accurate estimates of averages than could have been obtained by simply taking every sample after the burn-in.

Alternatively, one can accept the fact that the samples are correlated, but attempt to account for the correlations in some quantitative way. A scheme for estimating errors in means calculated from correlated Monte Carlo data may be found in [34].

For large dimensional state spaces one may still not be satisfied with estimating errors from a single Markov chain, even if the correlations are accounted for. An example is the sampling of probability density with multiple peaks, which may not individually give correct averages for state-space dependent quantities. A related issue is that the Markov chain generated by the algorithm may not be ergodic over the noise space. To reduce the severity of these possibilities the simplest procedure is to run multiple Markov chains of the same type but with statistically uncorrelated starting points, and treat the averages obtained from each chain as samples of the mean of the quantity of interest. The overall mean is calculated by

$$\overline{\langle O \rangle} = \frac{\sum_{i=1}^{\tilde{N}} \langle O \rangle_i}{\tilde{N}}, \tag{41}$$

where $\langle O \rangle_i$, $i = 1 \dots \tilde{N}$ are the means from the independent Markov chains as in Eq. (40), and the subscript π has been omitted for clarity. By the central limit theorem one would expect samples obtained in this way to approach a Gaussian distribution about the true mean. Therefore an estimate of the error in the mean, for \tilde{N} independent Markov chains is

$$\Delta[\overline{\langle O \rangle}] = \sqrt{\frac{\sum_{i=1}^{\tilde{N}} (\langle O \rangle_i - \overline{\langle O \rangle})^2}{\tilde{N} - 1}}, \tag{42}$$

where $\Delta[\cdot]$ denotes the error in the mean of the quantity of interest. This notation for error in the mean is used throughout this paper when referring to means obtained from Metropolis data.

4.3. Metropolis–Hastings algorithm for the stochastic gauge formalism

The appearance of the weight Ω as a multiplicative factor in the stochastic averages in Eq. (9) suggests an interpretation as a probability distribution. In this section, we show that it is possible to interpret the weight as a probability distribution over the space of all noise, and hence apply the Metropolis–Hastings algorithm.

For the purposes of computer simulation, time is necessarily discretised. To simulate the time evolution of a system for a period T we divide the time domain into $N + 1$ points so that each step forward in time is of length $\Delta\tau = T/N$. For an M mode system with the standard set of drift gauges (i.e. without the extra gauges and noises discussed in Section 3.1) we require $2M$ Gaussian distributed random numbers for each step forward in time, and thus $2MN$ random numbers to evolve an entire trajectory. Hence our fundamental object is a $2MN$ -component vector of Gaussian distributed random numbers $\vec{w} \in \mathbb{R}^{2MN}$, called the *noise vector*. The different realisations of \vec{w} give rise to different stochastic trajectories. The values of all stochastic phase space variables after N time steps are thus functions of \vec{w}

$$\vec{\alpha} = \vec{\alpha}[\vec{w}], \quad \Omega = \Omega[\vec{w}].$$

The stochastic average of some quantity, $\langle O \rangle_{\text{stoch}}$, is

$$\langle O \rangle_{\text{stoch}} = \int d^{2MN} \vec{w} P(\vec{w}) O(\vec{w}) = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n O[\vec{w}_i]}{n},$$

where the \vec{w}_i are drawn from a multi-dimensional Gaussian normal distribution.²

$$P(\vec{w}) = \frac{1}{(2\pi)^{MN}} \exp\left(-\frac{\vec{w}^2}{2}\right), \quad (43)$$

and $O = O[\vec{w}]$ is some quantity depending on the noise.

Hence the stochastic sampling of moments in stochastic gauge simulations, Eq. (9), is the sampling of a multi-dimensional integral

$$\langle \Omega(\vec{w}) O_{mn}(\vec{w}) \rangle_{\text{stoch.}} = \int d^{2MN} \vec{w} P(\vec{w}) \Omega(\vec{w}) O_{mn}(\vec{w}), \quad (44)$$

where $O_{mn}(\vec{w}) = \beta[\vec{w}]^m \alpha[\vec{w}]^n + (\beta[\vec{w}]^n \alpha[\vec{w}]^m)^*$.

The Metropolis–Hastings algorithm can be applied by identifying the state space S as \mathbb{R}^{2MN} and the state s as the vector of noises \vec{w} . The probability distribution we wish to sample is

$$\pi(\vec{w}) = \frac{P(\vec{w}) \Omega[\vec{w}]}{\mathcal{N}}, \quad (45)$$

where $\mathcal{N} = \int d^{2MN} \vec{w} P(\vec{w}) \Omega[\vec{w}]$ is a normalisation constant. The Metropolis algorithm can be used to generate a set of samples of $\pi(\vec{w})$, $\{\vec{w}_i | i = 1 \dots n\}$ and estimate quantum averages as

$$\langle (\hat{a}^\dagger)^m \hat{a}^n \rangle_{\text{QM}} = \frac{\mathcal{N} \langle O_{mn} \rangle_\pi}{2\mathcal{N}} = \frac{\langle O_{mn} \rangle_\pi}{2} \simeq \frac{\sum_{i=1}^n O_{mn}[\vec{w}_i]/n}{2}. \quad (46)$$

None of these steps require explicit knowledge of the normalisation constant \mathcal{N} , although in real-time unitary calculations we know analytically that it should always be one. Typically, we run multiple Markov chains as described in Section 4 to obtain means and standard-deviation error estimates, $\langle O_{mn} \rangle_\pi = \overline{\langle O_{mn} \rangle} \pm \Delta[\overline{\langle O_{mn} \rangle}]$.

For comparison, ordinary stochastic sampling uses a Gaussian normal random number generator to generate samples, $\{\vec{w}_i | i = 1 \dots N\}$, of $P(\vec{w})$ and calculates quantum averages as

$$\langle (\hat{a}^\dagger)^m \hat{a}^n \rangle_{\text{QM}} \simeq \frac{\sum_{i=1}^n \Omega[\vec{w}_i] O_{mn}[\vec{w}_i]/n}{2}.$$

Finally, we note that the Metropolis–Hastings algorithm as outlined above only optimises the stochastic sampling of moments at the final *target time*. At earlier times the probability distribution $\bar{\pi}(\vec{w})$ (where \vec{w} is the

² Other noise distributions are possible as the central limit theorem ensures Gaussian statistics in the limit of infinitesimal step size – however we use Gaussian statistics here.

noise vector \vec{w} truncated at the point corresponding to the earlier time) is different in that $\bar{\pi}(\vec{w}')/\bar{\pi}(\vec{w}) \neq \pi(\vec{w}')/\pi(\vec{w})$, and so we should not expect the Metropolis algorithm targeted to the later time to correctly sample the moments at earlier times. This is also illustrated in Fig. 1 where the relative weights vary considerably as time progresses.

4.4. Designing a candidate generating function

The choice of candidate generating function $q(\vec{w}, \vec{w}')$ is a subtle problem and perhaps the most crucial element of applying the Metropolis–Hastings algorithm to the stochastic gauge formalism. The Metropolis algorithm generates a Markov chain in noise space, where at each step the candidate generating function proposes a new noise \vec{w}' for the stochastic trajectory which is then accepted or rejected based on the acceptance probability defined by Eq. (38)

$$\alpha(\vec{w}, \vec{w}') = \min \left(1, \frac{\pi(\vec{w}')q(\vec{w}', \vec{w})}{\pi(\vec{w})q(\vec{w}, \vec{w}')} \right) = \min \left(1, \frac{P(\vec{w}')\Omega(\vec{w}')q(\vec{w}', \vec{w})}{P(\vec{w})\Omega(\vec{w})q(\vec{w}, \vec{w}')} \right).$$

The evaluation of the weight at the target time for the proposed noise $\Omega(\vec{w}')$ is a non-local procedure; even if only a single noise is altered in the time domain the entire trajectory from that time on has to be evolved until the target time in order to evaluate the new weight.

To separate the issue of sampling high weight trajectories from the Gaussian nature of the noise it is advisable to choose a candidate generating function such that

$$\frac{q(\vec{w}, \vec{w}')}{q(\vec{w}', \vec{w})} = \frac{P(\vec{w})}{P(\vec{w}')}.$$
(47)

For this class of generating functions, the probability of a move being accepted is

$$\alpha(\vec{w}, \vec{w}') = \frac{\pi(\vec{w}')q(\vec{w}, \vec{w}')}{\pi(\vec{w})q(\vec{w}', \vec{w})} = \frac{\Omega(\vec{w}')P(\vec{w}')q(\vec{w}, \vec{w}')}{\Omega(\vec{w})P(\vec{w})q(\vec{w}', \vec{w})} = \frac{\Omega(\vec{w}')}{\Omega(\vec{w})},$$
(48)

which is only dependent on the weight rather than the Gaussian distribution, $P(\cdot)$.

4.5. Time and frequency domain noise functions

A simple generating function of this type is selecting a number of entries in the current noise vector and generating new noises for these. This procedure can be carried out in either the time or the frequency domain of the noise. If the noises are altered in the time domain the stochastic trajectory is clearly unaltered up until the point of the first change. If noises are altered in the frequency domain then every noise in the time domain is affected to some extent. We consider altering noises in the frequency domain to be a more natural “small-step” for the Metropolis algorithm as all time-domain noises are changed by a small amount on average. All results presented in this paper use a generating function operating in the frequency domain.

In more detail, consider the R independent noise vectors of length N , $\vec{w}_j \in \mathbb{R}^N$, $j = 1 \dots R$, separately, instead of a single noise vector of length RN . Insight into strategies for altering noises may be gained by taking the discrete Fourier transform of each noise vector

$$\bar{K}_j(n) = \sum_{k=0}^{N-1} e^{2\pi i kn/N} \vec{w}_j(k).$$
(49)

Here a subscript is used to denote each of the R noise vectors, and should not be confused with the notation \vec{w}_i used previously to denote the sequence of noise vectors in a Markov chain. Brackets (\cdot) denote the components of the noise/spectrum vectors. The $n = 0$ component of \bar{K} is the mean of the noise, and the fact that the noise is real places constraints on the components. Let $\mathcal{N}(\mu, \sigma)$ denote a Gaussian distribution of mean μ and standard deviation σ . The Fourier transform of a vector whose components are normally distributed real variables ($\vec{w}(n)$ distributed as $\mathcal{N}(0, 1)$) is complex. For even N the real and imaginary parts of each component in the

range $2 \leq n \leq N - 1$ are distributed as $\mathcal{N}(0, \sqrt{N/2})$ and subject to the constraint $\vec{K}(N - n) = \vec{K}(n)^*$, while the $n = 1$ and $n = N$ components are real and distributed as $\mathcal{N}(0, \sqrt{N})$. For odd N , the real and imaginary parts of each component in the range $2 \leq n \leq N$ are distributed as $\mathcal{N}(0, \sqrt{N/2})$ and subject to the constraint $\vec{K}(N - n) = \vec{K}(n)^*$, while only the $n = 1$ component is real and distributed as $\mathcal{N}(0, \sqrt{N})$. These constraints preserve the total number of independent components N .

It is informative to study an individual trajectory and alter noise elements individually in frequency space to gauge the effect on the final weight. One might expect that the low frequency noise should be more important than the high frequency noise, as high frequency noise should “average out” over a shorter time scale and thus not affect the dynamics as much. We find that this is true and Fig. 4 quantifies the potential of frequency components of the noise to affect the final value of the weight. To obtain this graph four random noise vectors were generated and the corresponding stochastic trajectory evolved using real gauge discussed in Section 3.1 with mean atom number, $\bar{n} = 100$, and diffusion gauge $A = 2$. Using this initial trajectory as a starting point, a particular component of the noise spectrum was examined for its effect on the final weight by choosing 10^3 random values for that component and evolving the corresponding stochastic trajectory, with all other components of the noise spectrum the same. In Fig. 4 the standard deviation of the final weights obtained in this manner is plotted on a log scale against the spectrum component number that was altered. It is clear from this figure that the low frequency noises have greater potential to affect the final weight. This trend is independent of the initial trajectory, however we note the actual values of the standard deviations obtained vary considerably depending on the initial trajectory.

Another question to consider is how many sites in the frequency domain to alter in proposing a new noise vector, and how to select these. As the low-frequency noise seems to affect the final value of the weight more than the high-frequency noise, one might be tempted to consider strategies where low-frequency noise is altered more often so as to more effectively explore the noise space. In practice, we found that such strategies were no more effective than selecting sites randomly. Typically, we chose to alter of the order 1–10% of the total components of the noise. A good guiding principle is that Metropolis sampling is thought to be most efficient when approximately 50% of proposed moves are accepted during sampling [6,7]. We therefore experimented with different percentages with this principle in mind.

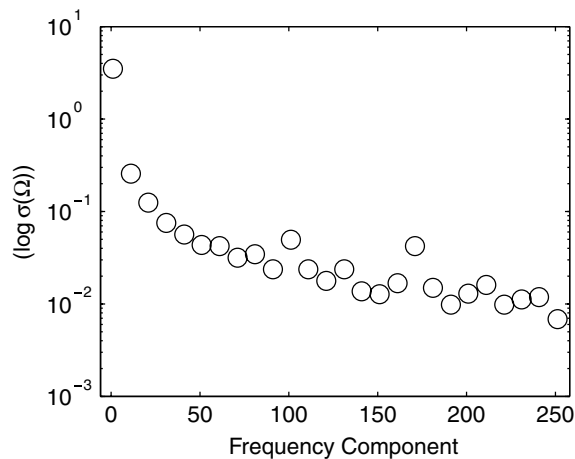


Fig. 4. Illustration of the effect of varying frequency components of the noise vectors on the final value of the weight. An initial stochastic trajectory from a real-gauge simulation of the anharmonic oscillator with mean atom number $\bar{n} = 100$ was taken as a starting point. The value of the weight at the end of the trajectory, $T = 0.5/\sqrt{\bar{n}} = 0.05$, was $\Omega = 0.9936$. The time step was $\Delta\tau = 10^{-4}$ so each noise vector contained 500 components. In frequency space each noise vector contains 250 complex components. A new random noise was chosen for a particular component of the noise vector, the entire trajectory re-evolved and the new final weight recorded. This procedure was repeated 10^4 times for each frequency component and the variance in the final weight, $\sigma(\Omega)$, calculated. This variance is plotted versus frequency component to give a measure of the potential of each frequency component to affect the final weight. Clearly low-frequency noise has more effect than high-frequency noise.

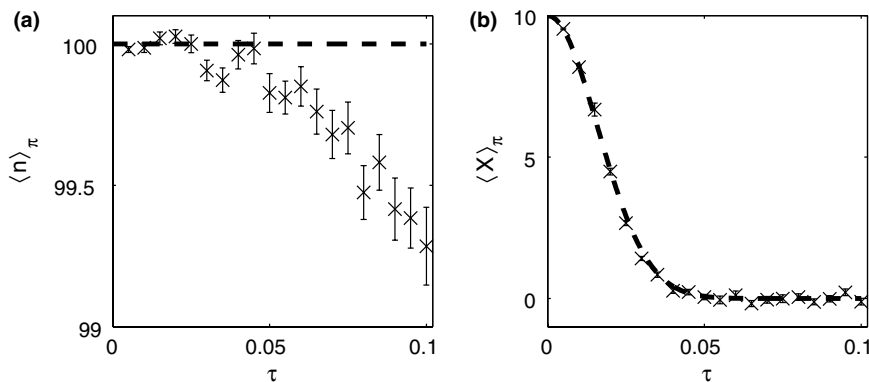


Fig. 5. Metropolis sampling of real-gauge simulation of anharmonic oscillator dynamics for an initial coherent state with mean atom number $\bar{n} = 100$ ($A = 2, \lambda = 1/2$). (a) Mean atom number. Note the difference in the vertical scale compared with Fig. 2(a). (b) Mean X -quadrature. Metropolis averages and error estimates (crosses with error bars) were calculated from the means of 10^2 Markov chains, each taking 10^4 samples after a burn-in of 10^4 proposals. The total simulation time was $T = 1/\sqrt{\bar{n}} = 0.1$ and the time step used was $\Delta\tau = T/10^3 = 10^{-4}$. Proposals were generated by uniformly selecting 10% of the components of the noise vectors in the frequency domain and generating new random noise for those sites. The analytic results are shown as dashed lines.

4.6. Results

We now present the results of a Metropolis sampling of the real-time dynamics of the anharmonic oscillator using the real gauge discussed in Section 3.1 and the candidate generating functions discussed in Section 4.4. To allow for the distribution $\pi(\vec{w}) = P(\vec{w})\Omega(\vec{w})/\mathcal{N}$ to be multiply-peaked we use the more robust technique of estimating means and errors using multiple Markov chains as discussed in Section 4. The starting points of the Markov chains were generated as statistically uncorrelated Gaussian noise so as to ensure that no bias is introduced in the use of multiple Markov chains for the estimation of errors, as discussed in Section 4.2.

Fig. 5 shows the Metropolis sampling of $\langle \hat{n} \rangle = \langle \hat{a}^\dagger \hat{a} \rangle$ and $\langle \hat{X} \rangle = \langle (\hat{a} + \hat{a}^\dagger)/2 \rangle$ for a real-gauge simulation of the anharmonic oscillator with $\bar{n} = 100$. We targeted the Metropolis sampling to 20 time points in intervals of $\Delta\tau = 0.005$ from 0 to the final time of $\tau = T = 1/\sqrt{\bar{n}} = 0.1$. The average at each time point is completely independent of the other time points. Altogether $n = 10^6$ samples were used for the average at each point so statistically the sampling is comparable to the stochastic sampling in Fig. 2. However, because the Metropolis sampling has to be run independently for each time point and each Markov chain has to burn in before sampling begins, there is clearly a much greater computational effort required to obtain the Metropolis results. Nevertheless the Metropolis results are far more reliable than the stochastic results – the reliable simulation time appears to be at least $\tau = 0.05$, which is approximately a factor of 5 improvement over the stochastic sampling.³

We note that even with the Metropolis algorithm the sampled mean atom number appears to decay slightly at longer times. Although the results are still accurate to within 1% the estimated error bars do not account for the difference from the analytic result. We believe that this behaviour is due to the inherent difficulty in sampling the skewed weight distribution, even with Monte Carlo algorithms. In particular, as noted in Section 3.2, the sampled mean and median of the weights were different by many orders of magnitude at the end time of the stochastic simulation, and the calculated mean was very sensitive to discarding the tails. It seems that the Metropolis sampling was not able to completely overcome these difficulties at the end time. The results are nonetheless vastly improved compared with stochastic sampling. We also note that at the intermediate time ($\tau = 0.025$) where the above measures were not so drastic but the stochastic sampling was still poor, the Metropolis sampling has actually produced quite accurate results.

We experimented with different distributions of the 10^6 allowed samples between different numbers of Markov chains. There is a trade-off between number of Markov chains and the length of the chain – if longer

³ The numerical value of this factor will depend on the mean atom number in the simulation, as well as the values of the various parameters in the Metropolis simulation.

chains are run (e.g. 10^2 chains each with 10^4 samples), then the means obtained from each chain are more accurate, and closer to Gaussian about the true mean. However, there are less chains to average over compared with a run of a larger number of shorter chains (e.g. 10^3 chains each with 10^3 samples). It was a matter of empirical observation to determine the best balance.

We also experimented with the fraction of noises to be altered in the frequency domain when proposing a new noise vector. As a rough guide we aimed to have 50% of proposed moves accepted during sampling. However at short times we found that even if a very large fraction of the noises were altered (e.g. 50% or more) a large fraction of proposed changes were accepted ($\sim 90\%$). This large acceptance rate is due to the distribution of weights being relatively narrow at short times. At longer times the distribution has spread out sufficiently such that lower acceptance rates are possible.

Clearly there are many parameters to be optimised in Metropolis sampling of stochastic gauge equations and we have only scratched the surface. We have aimed to present conceptually-simple approaches to illustrate the principle rather than exhaustively optimise all parameters. The fact that significant improvements were seen over stochastic sampling – even with our very simple Metropolis schemes – gives us confidence that further improvements are possible.

5. Branching algorithm

The second Monte Carlo technique that we investigate for real-time stochastic-gauge simulations is a branching algorithm similar to that used in Green's function Monte Carlo, see e.g. [35]. Corney and Drummond [17] have previously used this algorithm for stochastic simulations in imaginary time using a Gaussian basis (a generalisation of the stochastic-gauge basis).

The branching algorithm is simpler to describe and more straightforward to apply to stochastic gauge simulations. The branching algorithm works by concurrently evolving a “population” of stochastic trajectories in time, and periodically cloning those that acquire a large weight and killing those that acquire a small weight.

5.1. Algorithm

We define T to be the total simulation time, $\Delta\tau_b$ be the time interval between branching events and $\Delta\tau$ the fundamental time-step for integrating the SDEs. In practice, it is desirable for the number of branching events $B = T/\Delta\tau_b$ and the number of time steps in a branching period $\Delta\tau_b/\Delta\tau$ to be integers. Formally the branching algorithm can be stated as

- (1) Begin with an initial population of N_{pop} stochastic trajectories.
- (2) Evolve all stochastic trajectories forward in time for a period $\Delta\tau_b$.
- (3) for $i = 1, 2, \dots, N_{\text{pop}}$
 - Generate a uniform random variable u between 0 and 1.
 - Make $m_i = \lfloor \Omega_i/\bar{\Omega} + u \rfloor$ clones of trajectory i .
 - Set $\Omega_i = 1$.
- (4) Set $N_{\text{pop}} = \sum_{i=1}^{N_{\text{pop}}} m_i$.

Here we set $\bar{\Omega} = \langle \Omega \rangle$ to ensure that the number of trajectories in the population N_{pop} remains constant on average. Because Ω does not couple into the SDEs for the mode variables, further evolution is not affected by the resetting of the weights at each branching time. The statistical equivalence between the weight and the multiplicity of paths means that the physical moments are unchanged on average by the branching procedure. To see this note that the average of the mean of an observable O after the branching event is

$$\langle O \rangle = \sum_{i=1}^{N_{\text{pop}}} \int_0^1 m_i(u) du O_i \Big/ \sum_{i=1}^{N_{\text{pop}}} \int_0^1 m_i(u) du = \sum_{i=1}^{N_{\text{pop}}} (\Omega_i/\bar{\Omega}) O_i \Big/ \sum_{i=1}^{N_{\text{pop}}} \Omega_i/\bar{\Omega} = \frac{1}{N_{\text{pop}}} \sum_{i=1}^{N_{\text{pop}}} \Omega_i O_i, \quad (50)$$

which is identical to the mean before the branching event.

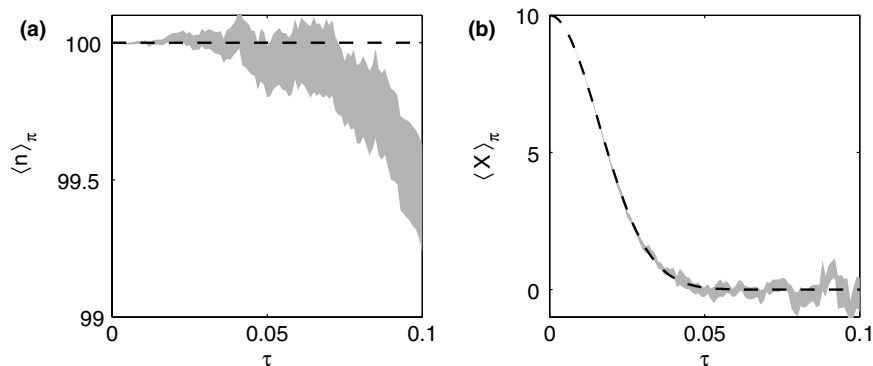


Fig. 6. Branching-algorithm sampling of real-gauge simulation of anharmonic oscillator dynamics for an initial coherent state with mean atom number $\bar{n} = 100$ ($A = 2$, $\lambda = 1/2$). (a) Mean atom number. Again note the difference in the vertical scale compared with Fig. 2(a). (b) Mean X -quadrature. Branching averages and error estimates, indicated by the shaded region, were calculated from the means of 10^2 populations, each containing 10^4 trajectories on average. The total simulation time was $\tau = 1/\sqrt{\bar{n}} = 0.1$ and the time step used was $\Delta\tau = \tau/10^3 = 10^{-4}$. The branching time was $\Delta\tau_b = 10^{-3}$. The analytic results are shown as dashed lines.

5.2. Results

We now present the results of a branching-algorithm sampling of real-time dynamics of the anharmonic oscillator using the real gauge discussed in Section 3.1. Similarly to the Metropolis sampling, we calculate averages and error estimates from multiple independent populations.

Fig. 6 shows a branching-algorithm sampling of $\langle \hat{n} \rangle = \langle \hat{a}^\dagger \hat{a} \rangle$ and $\langle \hat{X} \rangle = \langle (\hat{a} + \hat{a}^\dagger)/2 \rangle$ for a real-gauge simulation of the anharmonic oscillator with $\bar{n} = 100$. Again we see a large improvement in the sampling compared with stochastic sampling, Fig. 2. A clear advantage of the branching algorithm over Metropolis is that it generates physical moments at every time step as opposed to being targeted to a single time.

At each time there are, on average, 10^6 stochastic trajectories contributing to the stochastic averages. As with the Metropolis, it was again a matter of experimenting with different ratios of the number of trajectories to the number of populations to determine the best balance. We present results for the same division of trajectories amongst populations (10^2 populations of 10^4 trajectories) as samples amongst Markov chains for the Metropolis algorithm so that the results are as comparable as possible.

The only free parameter in the algorithm itself is the time between branching events, $\Delta\tau_b$. There is a trade-off between making this interval large enough that the weights spread out sufficiently for the branching to be meaningful and small enough to improve the sampling continuously in time. The branching interval used for this simulation, $\Delta\tau_b = 10^{-3}$, is small on the scale of the dynamics of the system but large enough to allow the weights to diverge significantly between branching events.

6. Discussion and outlook

In this paper, we have demonstrated the use of two Monte Carlo techniques, the Metropolis algorithm and a branching algorithm, for real-time calculations of quantum dynamics with the stochastic-gauge method. This work should be considered as a proof of principle rather than a fully optimised ‘recipe’. It is part of a larger program of optimising bases, gauges and algorithms for stochastic simulations of quantum dynamics in real and imaginary time. The general goal of this program is to increase the useful simulation time for stochastic methods. Results in this direction are necessarily limited to demonstrating improvements with particular techniques for particular systems [19,24,25] as there are no known analytic methods for determining the time scale at which sampling errors lead to deviations from the exact solution.

In this work we have restricted ourselves to real gauges so that the weights remain real and can be interpreted as probabilities, and have considered the single-mode anharmonic oscillator as an example of our methods. In order to control the divergence of stochastic trajectories using real gauges we have explored a previously untested gauge freedom resulting from the choice of a non-square noise matrix. The resulting dis-

tribution of weights becomes highly skewed on a time scale proportional to the inverse of the mean atom number. The weight parameter for most stochastic trajectories tends towards zero, whereas very few tend towards a large weight. Such distributions are likely to be ubiquitous for unitary real-time stochastic-gauge simulations with real gauges, as the weight distribution necessarily broadens with time while the mean weight must remain unity.

While such skewed distributions are difficult to sample with the usual stochastic methods, they are suited to Monte Carlo importance sampling techniques that preferentially sample high-weight trajectories. Indeed we found encouraging improvements over stochastic sampling using both the Metropolis and branching algorithms. From comparing the results of stochastic sampling plotted in Fig. 2 to those using Monte Carlo techniques in Figs. 5 and 6 we can see that we have extended the useful simulation time by about a factor of three. These improvements are strongly indicative that the errors found with these techniques are statistical in origin, rather than due to systematic effects like boundary terms, which would not improve simply due to improved sampling methods.

The branching algorithm is the more straightforward to apply because it has only one free parameter (the branching interval) and produces results at every time step. We suggest it as the best starting point for future Monte Carlo simulations. By contrast the Metropolis algorithm has to be targeted to a particular time and so seems less useful. However, there is a lot more freedom in the Metropolis algorithm and a vast literature exists on optimising sampling for particular distribution (see e.g. [22] and references therein). Thus it seems quite possible that it will be better suited to some problems, especially when further improvements over the branching algorithm are desirable.

A timely application of these methods is in theoretical calculations for ultra-cold atomic gases [17,36]. QMC methods have been used to calculate some static properties of ultra-cold gases, e.g. see [37,38]. These systems are quantum-many body by nature and hence few exact theoretical results exist. They are a useful testing ground for theory due to their purity and well-understood controllable interactions.

Traditionally, Monte Carlo techniques have been highly successful in imaginary-time calculations for thermal equilibrium. This paper has extended the use of these techniques to real-time quantum-dynamical calculations and thus opens a new domain of application for these algorithms. In future, these techniques need to be extended to many-mode, many-particle problems where exact solutions are not known.

Acknowledgements

We thank Piotr Deuar for helpful discussions, and the Australian Research Council, the Queensland state government and the University of Queensland for financial support.

References

- [1] R.P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* 21 (6&7) (1982) 467.
- [2] W. von der Linden, A quantum Monte Carlo approach to many-body physics, *Phys. Rep.* 220 (1992) 53.
- [3] D.M. Ceperley, Microscopic simulations in physics, *Rev. Mod. Phys.* 71 (1999) 438.
- [4] D. Ceperley, B. Alder, Quantum Monte-Carlo, *Science* 231 (1986) 555.
- [5] D.M. Ceperley, Path integrals in the theory of condensed helium, *Rev. Mod. Phys.* 67 (2) (1995) 279.
- [6] W.R. Gilks, S. Richardson, D.J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, first ed., Chapman & Hall, London, 1996.
- [7] D.P. Landau, K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, first ed., Cambridge University Press, New York, 2000.
- [8] E. Wigner, On the quantum correction for thermodynamic equilibrium, *Phys. Rev.* 40 (1932) 749.
- [9] P.D. Drummond, C.W. Gardiner, Generalized P-representations in quantum optics, *J. Phys. A* 17 (1980) 2353.
- [10] C.W. Gardiner, P. Zoller, *Quantum Noise*, second ed., Springer, Berlin, 1999.
- [11] S.J. Carter, P.D. Drummond, M.D. Reid, R.M. Shelby, Squeezing of quantum solitons, *Phys. Rev. Lett.* 58 (1987) 1841.
- [12] P. Kinsler, P.D. Drummond, Comment on “Quantum noise in the parametric oscillator: from squeezed states to coherent-state superpositions”, *Phys. Rev. Lett.* 64 (1990) 236.
- [13] P.D. Drummond, P. Kinsler, Quantum tunneling and thermal activation in the parametric oscillator, *Phys. Rev. A* 40 (1989) 4813.
- [14] M.J. Steel, M.K. Olsen, L.I. Plimak, P.D. Drummond, S.M. Tan, M.J. Collett, D.F. Walls, R. Graham, Dynamical quantum noise in trapped Bose–Einstein condensates, *Phys. Rev. A* 58 (6) (1998) 4824.
- [15] P.D. Drummond, J.F. Corney, Quantum dynamics of evaporatively cooled Bose–Einstein condensates, *Phys. Rev. A* 60 (4) (1999) R2661.

- [16] J.F. Corney, P.D. Drummond, Gaussian quantum operator representation for bosons, *Phys. Rev. A* 68 (2003) 063822.
- [17] J.F. Corney, P.D. Drummond, Gaussian quantum Monte Carlo methods for fermions and bosons, *Phys. Rev. Lett.* 93 (2004) 260401.
- [18] M. Greiner, O. Mandel, T.W. Hänsch, I. Bloch, Collapse and revival of the matter wave field of a Bose–Einstein condensate, *Nature* 419 (6902) (2002) 51.
- [19] P. Deuar, P.D. Drummond, Gauge P-representations for quantum-dynamical problems: removal of boundary terms, *Phys. Rev. A* 66 (2002) 033812.
- [20] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* 21 (1953) 1087.
- [21] W.K. Hastings, Monte-Carlo sampling methods using Markov chains and their applications, *Biometrika* 57 (1970) 97.
- [22] S. Chib, E. Greenberg, Understanding the Metropolis–Hastings algorithm, *The American Statistician* 49 (1995) 327.
- [23] C.W. Gardiner, *Handbook of Stochastic Methods*, second ed., Springer, Berlin, 1985.
- [24] L.I. Plimak, M.K. Olsen, M.J. Collett, Optimization of the positive-P representation for the anharmonic oscillator, *Phys. Rev. A* 64 (2001) 025801.
- [25] P. Deuar, First-principles quantum simulations of many-mode open interacting Bose gases using stochastic gauge methods, Ph.D. Thesis, University of Queensland, 2004.
- [26] E.L. Pollock, D.M. Ceperley, Simulation of quantum many-body systems by path-integral methods, *Phys. Rev. B* 30 (1984) 2555.
- [27] R. Egger, C.H. Mak, Low-temperature dynamical simulation of spin-boson systems, *Phys. Rev. B* 50 (1994) 15210.
- [28] D. Jaksch, C. Bruder, J.I. Cirac, C.W. Gardiner, P. Zoller, Cold bosonic atoms in optical lattices, *Phys. Rev. Lett.* 81 (15) (1998) 3108.
- [29] M. Greiner, O. Mandel, T. Esslinger, T.W. Hänsch, I. Bloch, Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms, *Nature* 415 (2002) 39.
- [30] M. Greiner, I. Bloch, O. Mandel, T.W. Hänsch, T. Esslinger, Exploring phase coherence in a 2D lattice of Bose–Einstein condensates, *Phys. Rev. Lett.* 87 (16) (2001) 160405.
- [31] M.R. Dowling, P.D. Drummond, M.J. Davis, P. Deuar, Time-reversal test for stochastic quantum dynamics, *Phys. Rev. Lett.* 94 (2005) 130401.
- [32] P.D. Drummond, P. Deuar, Quantum dynamics with stochastic gauge simulations, *J. Opt. B* 5 (3) (2003) S281.
- [33] A. Gilchrist, C.W. Gardiner, P.D. Drummond, *Phys. Rev. A* 55 (1997) 3014.
- [34] G.J. Daniell, A.J.G. Hey, J.E. Mandula, Error analysis for correlated Monte Carlo data, *Phys. Rev. D* 30 (1984) 2230.
- [35] N. Trivedi, D.M. Ceperley, Ground-state correlations of quantum antiferromagnets: a Green-function Monte Carlo study, *Phys. Rev. B* (1990) 4552.
- [36] P.D. Drummond, P. Deuar, K.V. Kheruntsyan, Canonical Bose gas simulations with stochastic gauges, *Phys. Rev. Lett.* 92 (2004) 40405.
- [37] J. Carlson, S.-Y. Chang, V.R. Pandharipande, K.E. Schmidt, Superfluid Fermi gases with large scattering length, *Phys. Rev. Lett.* 91 (2003) 050401.
- [38] G.E. Astrakharchik, J. Boronat, J. Casulleras, S. Giorgini, Equation of state of a Fermi gas in the BEC-BCS crossover: a quantum Monte Carlo study, *Phys. Rev. Lett.* 93 (2004) 200404.