Transactions of Mathematics and Its Applications (2019) **3**, 1–17
doi:10.1093/imatrm/tnz001

Bold diagrammatic Monte Carlo in the lens of stochastic iterative methods

Yingzhou Li*
Department of Mathematics, Duke University
*Corresponding author: yingzhouli0417@gmail.com

and

Jianfeng Lu
Department of Mathematics, Department of Chemistry, and Department of Physics, Duke University

[Received on 14 October 2017; revised on 2 October 2018; accepted on 25 December 2018]

This work aims at understanding of bold diagrammatic Monte Carlo (BDMC) methods for stochastic summation of Feynman diagrams from the angle of stochastic iterative methods. The convergence enhancement trick of the BDMC is investigated from the analysis of condition number and convergence of the stochastic iterative methods. Numerical experiments are carried out for model systems to compare the BDMC with related stochastic iterative approaches.

Keywords: bold diagrammatic Monte Carlo; stochastic iterative method; diagrammatic Monte Carlo; quantum Monte Carlo; fixed-point iteration.

1. Introduction

The bold(-line) diagrammatic Monte Carlo (BDMC) method (Prokof'ev & Svistunov, 2007) employs bold-line trick in the diagrammatic Monte Carlo (DMC) method to simulate integrands represented by a diagrammatic structure. Such a method adopts mathematical tools including Monte Carlo sampling of the diagram and iterative method for the bold-line trick. This note first establishes a solid mathematical understanding of the iterative method proposed in the original BDMC paper (Prokof'ev & Svistunov, 2007). Second, this note clarifies the relationship between the iterative method in BDMC and stochastic iterative methods. Based on the explicit connection, a few stochastic iterative methods (Polyak, 1964; Duchi *et al.*, 2011; Kingma & Ba, 2015; Bottou *et al.*, 2016; Tan *et al.*, 2016), widely used and extensively tested in the field of machine learning, are reintroduced in this note as potential alternatives to BDMC with potentially faster convergence.

Both DMC and BDMC are proposed for 'many-electron problem' that involves interacting electrons. In order to describe an interacting electron system, the dimension of the Hilbert space grows exponentially in the system size; the high dimensionality becomes a fundamental difficulty for numerical treatment. The quantum Monte Carlo methods are thus natural candidates for these problems. Conventional quantum Monte Carlo methods calculate solutions on finite-size lattices and then estimate the solution of the thermodynamic limit (thus infinite system) via extrapolations; see, e.g., reviews Foulkes *et al.* (2001), Ceperley (2010), Kolorenc & Mitas (2011) and Austin *et al.* (2012). On the other hand, the DMC and BDMC sample and sum the truncated Feynman diagram of the infinite system (Van Houcke *et al.*, 2008). The Feynman diagram is a well-developed and widely used tool in many-body perturbation theory; see, e.g., books Mattuck (1992) and Fetter & Walecka (2003). In particular, the summation of series of Feynman diagrams works well for those that are convergent and sign positive.

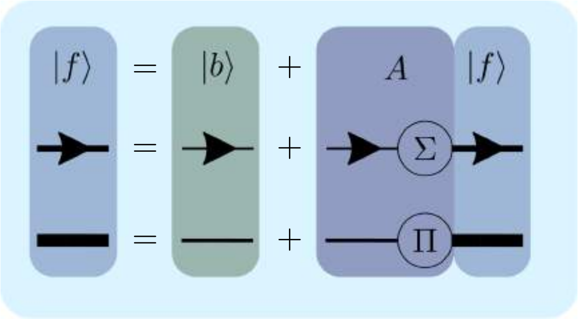


FIG. 1. Relationship between linear system, Dyson equation and Bethe–Salpeter equation.

In order to obtain the summation of the infinite long diagram, the extrapolation technique is applied to a few results corresponding to different numbers of truncation orders. However, for many systems, the series of diagrams are asymptotic (e.g., for strong coupling systems) and sign alternating. No solution, so far, fully addresses these issues. Techniques have been developed to enlarge the radius of the convergence and reduce the number of terms in the diagram. BDMC is one of the promising technique among those. BDMC, instead of summing diagrams directly, sums all the bold-line diagrams for irreducible single-particle self-energy Σ and pair self-energy Π following Dyson and Bethe–Salpeter equation, respectively (Prokof'ev & Svistunov, 2007, 2008a). Based on the 'sign-blessing' phenomenon, BDMC was successfully applied to one-particle s-scattering problem (Prokof'ev & Svistunov, 2007), the Bardeen-Cooper-Schrieffer-Bose-Einstein condensation crossover in the strongly imbalanced regime (Prokof'ev & Svistunov, 2008a,b), unitary Fermi gas (Van Houcke *et al.*, 2012), Fermionized frustrated spins (Kulagin *et al.*, 2013), two-dimensional Hubbard model (LeBlanc *et al.*, 2015), etc.

As from the original paper (Prokof'ev & Svistunov, 2007), BDMC can be viewed as trying to solve a self-consistent linear equation

$$|f\rangle = |b\rangle + A\,|f\rangle\,, \tag{1}$$

where $|f\rangle$ is an unknown vector, $|b\rangle$ is a given vector and A is a linear operator. Figure 1 provides the connection between (1) with either the Dyson equation or the Bethe–Salpeter equation (Van Houcke *et al.*, 2012). Since either Σ or Π involves infinite terms of diagrams, the evaluation is carried out via a stochastic procedure up to a given number of terms. The evaluation of $A\,|f\rangle$, therefore, is stochastic, where the error is controlled by the number of Monte Carlo sampling. Prokof'ev & Svistunov (2007) proposed an iterative method to solve for $|f\rangle$ in (1) under the stochastic setting, whose connection to conventional iterative algorithm in numerical linear algebra is not obvious from the first sight. As BDMC achieves success in many interacting systems and shows great promise, establishing a concrete understanding of the iterative method in a mathematical way is crucial for further improvement of the method and potentially adapt the method to other applications.

In this note, we interpret the 'magic' method proposed in Prokof'ev & Svistunov (2007) as a combination of two crucial steps. The first step replaces the original operator A by a quadratic polynomial of A, $p(A)$, such that $p(0) = 0$, $p(A) \succeq 0$ and potentially $\kappa\,(p(A)) \ll \kappa\,(A^*A)$, where '$A \succeq 0$' means that A is a positive semidefinite matrix and $\kappa\,(A)$ denotes the condition number of matrix A. Here '$p(0) = 0$' guarantees the equality in (1), '$p(A) \succeq 0$' guarantees the convergence of the iterative method and '$\kappa\,(p(A)) \ll \kappa\,(A^*A)$' enables faster convergence rate. Based on this understanding, we suggest another form of the quadratic polynomial such that the similar properties can be achieved for a wider range of A. In the second step, a fixed-point iteration with adaptive stepsize is applied to (1) with A being replaced by $p(A)$ and the corresponding update on $|b\rangle$. When A is a Hermitian matrix, the second step can be viewed as a method of stochastic gradient descent. Hence, later in the note, we employ stochastic gradient descent methods from machine learning as alternative methods. All methods are tested on synthetic stochastic matrix A instead of diagrams for real physical systems; those will be considered for future works.

In this note, we will provide mathematical understanding of the stochastic iterative method (Prokof'ev & Svistunov, 2007) in Section 2. Section 3 lists several alternative stochastic iterative methods. All the mentioned methods are tested and compared in Section 4. Finally, in Section 5, we conclude the note together with discussion of possible future works.

2. Numerical method of BDMC

Recall the fixed-point problem BDMC tries to solve

$$|f\rangle = |b\rangle + A\,|f\rangle\,. \tag{2}$$

In the viewpoint of linear algebra, we rewrite the equation as

$$M\,|f\rangle = |b\rangle\,, \tag{3}$$

where $M = I - A$, I is the identity matrix of the same size as A. BDMC proposes replacements $|b\rangle \to |\widetilde{b}\rangle = |b\rangle - \lambda A\,|b\rangle$ and $A \to \widetilde{A} = (1+\lambda)A - \lambda A^2$ to ensure the convergence of the iterative method, where λ is a constant related to the spectrum of A. Then a simple fixed-point iteration is coupled with a special Nørlund means to solve (2) with $\widetilde{A}$ and $|\widetilde{b}\rangle$. In the following subsections, we reinterpret the former as a preconditioning step and the latter as a stochastic gradient descent method with diminishing stepsize.

In the rest of this note, we would stick to linear algebra notations as in (3). Accordingly, we have $|\widetilde{b}\rangle = (1-\lambda)\,|b\rangle + \lambda M\,|b\rangle$, $\widetilde{M} = I - \widetilde{A} = (1-\lambda)M + \lambda M^2$. Additionally, we follow the assumption as in Prokof'ev & Svistunov (2007) that A is Hermitian, i.e., $A^* = A$. Therefore, both M and $\widetilde{M}$ are Hermitian as well.

2.1 *Preconditioning indefinite matrices*

For almost all first-order iterative methods, positivity of M is required for convergence. Methods that work for indefinite matrices, such as Minimum residual method (MINRES) (Paige & Saunders, 1975) and Generalized minimal residual method (GMRES) (Saad & Schultz, 1986), adopt some transforms of M in their objective function, e.g., M^*M, to maintain convexity of the problem. Another important property of M or $\widetilde{M}$ related to convergence rate is the condition number. In general, smaller condition number leads to faster convergence. However, treatment as M^*M or M^2 squares the condition number that is undesirable in practice. In this section, we analyze the positivity of $\widetilde{M}$ and its condition number comparing to $\kappa\left(M^2\right)$.

Assume M is an indefinite invertible matrix of size n by n. According to the earlier assumption, M is Hermitian. Let $M = Q\Lambda Q^*$ be the eigenvalue decomposition of M, where Q is a unitary matrix of size n by n and Λ is a diagonal matrix with M's eigenvalues, $\{m_1, m_2, \ldots, m_n\}$, in decreasing order, i.e.,

$m_1 \geq m_2 \geq \cdots \geq m_\ell > 0 > m_{\ell+1} \geq \cdots \geq m_n$ for $1 < \ell < n$. To simplify the presentation in the sequel, we introduce handy notations as $L_+ = \max_{1 \leq i \leq n} m_i$, $L_- = \min_{1 \leq i \leq n} m_i$, $\tau_+ = \min_{m_i > 0} m_i$ and $\tau_- = \max_{m_i < 0} m_i$, which define the boundaries of the positive and negative spectrum of M.

$\widetilde{M} = (1-\lambda)M + \lambda M^2$ inherits the same eigenvectors as M. The eigenvalues of $\widetilde{M}$ are $\left\{(1-\lambda)m_i + \lambda m_i^2\right\}_{i=1}^n$. Denote the quadratic polynomial depending on parameter λ as $p_\lambda(x) = \lambda x^2 + (1-\lambda)x$. The eigenvalues of $\widetilde{M}$, therefore, are polynomial $p_\lambda(x)$ acting on the eigenvalues of M. $\widetilde{M}$ being a positive definite matrix is equivalent to $p_\lambda(m_i) > 0$ for all $m_1, \ldots, m_n$. Since $p_\lambda(x)$ is a quadratic polynomial with zero being one of its root, $p_\lambda(\tau_-) > 0$ and $p_\lambda(\tau_+) > 0$ imply that $\lambda > 0$. At the same time, the second root of $p_\lambda(x)$, $\frac{\lambda-1}{\lambda}$ must lie in the interval (τ_-, τ_+). Hence, the equivalent condition for $\widetilde{M}$ being positive definite is that

$$\tau_- < \frac{\lambda - 1}{\lambda} < \tau_+ \Leftrightarrow \begin{cases} \lambda > \frac{1}{1-\tau_-} & \\ \lambda < \frac{1}{1-\tau_+} & \text{if } \tau_+ < 1 \end{cases}. \tag{4}$$

We now move on to the second concern, the condition number of $\widetilde{M}$ comparing to that of M^2. Using the notations above, the condition number of M^2 is

$$\kappa\left(M^2\right) = \frac{\max\left(L_+^2, L_-^2\right)}{\min\left(\tau_+^2, \tau_-^2\right)},$$

and the condition number of $\widetilde{M}$ is

$$\kappa_\lambda\left(\widetilde{M}\right) = \frac{\max(p_\lambda(L_+), p_\lambda(L_-))}{\min(p_\lambda(\tau_+), p_\lambda(\tau_-))}. \tag{5}$$

The optimal choice $\lambda^* = \arg\min_{\lambda \text{ satisfies(4)}} \kappa_\lambda\left(\widetilde{M}\right)$ is difficult to determine. On the other hand, a simple choice

$$\widehat{\lambda} = \begin{cases} \frac{1}{1-\tau_+-\tau_-} & \text{if } \tau_+ + \tau_- < 1 - \frac{1}{C} \\ C & \text{if } \tau_+ + \tau_- \geq 1 - \frac{1}{C} \end{cases}$$

leads to

$$\min\left(p_{\widehat{\lambda}}(\tau_+), p_{\widehat{\lambda}}(\tau_-)\right) = \begin{cases} -\widehat{\lambda}\tau_+\tau_- & \text{if } \tau_+ + \tau_- < 1 - \frac{1}{C} \\ \widehat{\lambda}\tau_-(\tau_- - 1 + \frac{1}{C}) & \text{if } \tau_+ + \tau_- \geq 1 - \frac{1}{C} \end{cases},$$

where C is a sufficiently large constant. When $\left|\tau_-\right|$ is orders of magnitude larger than τ_+ and $\max\left(L_+, -L_-\right) \gg \left|\tau_-\right|$, the condition number $\kappa_{\widehat{\lambda}}\left(\widetilde{M}\right)$ is roughly $\frac{|\tau_-|}{\tau_+}$ times smaller than $\kappa\left(M^2\right)$. More extreme example is that when $\left|\tau_-\right| \sim \left|L_-\right| > L_+ \gg \tau_+$, the condition number of $\widetilde{M}$ is roughly constant, $\kappa_{\widehat{\lambda}}\left(\widetilde{M}\right) = \mathcal{O}(1)$, whereas the condition number $\kappa\left(M^2\right)$ could be gigantic if the ratio $\left|L_-\right|/\tau_+$ is gigantic. Figure 2 shows the comparison between $p_\lambda(M)$ and M^2. The largest eigenvalue of M^2 is obviously larger than that of $p_\lambda(M)$, and the smallest eigenvalue of M^2 is also smaller than that of $p_\lambda(M)$ (shown in the zoom-in subfigure). Therefore, in this case, the condition number of M^2 is much larger than that of $p_\lambda(M)$. However, when we swap the position of τ_+ and τ_-, e.g., τ_+ is orders of magnitude larger than $\left|\tau_-\right|$ and $\max\left(L_+, -L_-\right) \gg \tau_+$, the condition number $\kappa_{\widehat{\lambda}}\left(\widetilde{M}\right)$ could be of the same order as $\kappa\left(M^2\right)$ if $\left|\tau_-\right| \sim 1$. The limitation comes from the restricted expression of $p_\lambda(x)$, where the second root must be smaller than one if $\lambda > 0$.

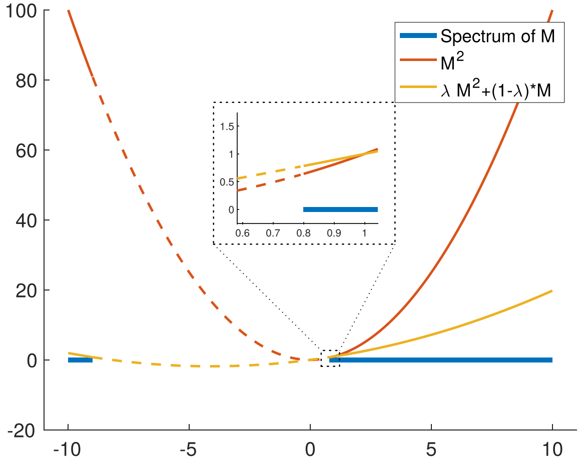


FIG. 2. Two quadratic polynomials act on the spectrum of a matrix M with $L_- = -10$, $\tau_- = -9$, $\tau_+ = 0.8$ and $L_+ = 10$.

In summary of the above analysis, we observe that the quadratic polynomial of the matrix, $p_\lambda(M)$ with a careful choice of λ, turns M into a positive definite matrix. For a certain class of matrices, $p_\lambda(M)$ is much well conditioned than the traditional technique M^2, which is favorable for the latter iterative method. Furthermore, according to the choice of $\widehat{\lambda}$, the improvement of the condition number is more significant when M's close-to-zero eigenvalues are tilted around zero.

Generic quadratic polynomial preconditioning. Inspired by the above analysis, we propose a more generic quadratic polynomial $q_\delta(x) = x^2 - \delta x$ for preconditioning, where δ is a parameter playing the similar role as λ. By abuse of notation, $\widetilde{M} = q_\delta(M)$. Similar as before, $\widetilde{M}$ and M share the same eigenvectors and the eigenvalues of $\widetilde{M}$ are $q_\delta(m_i) = m_i^2 - \delta m_i$. Since δ is the second root of $q_\delta(x)$, $\tau_- < \delta < \tau_+$ guarantees the positivity of $\widetilde{M}$. The definition of the condition number of $\widetilde{M}$ is as (5),

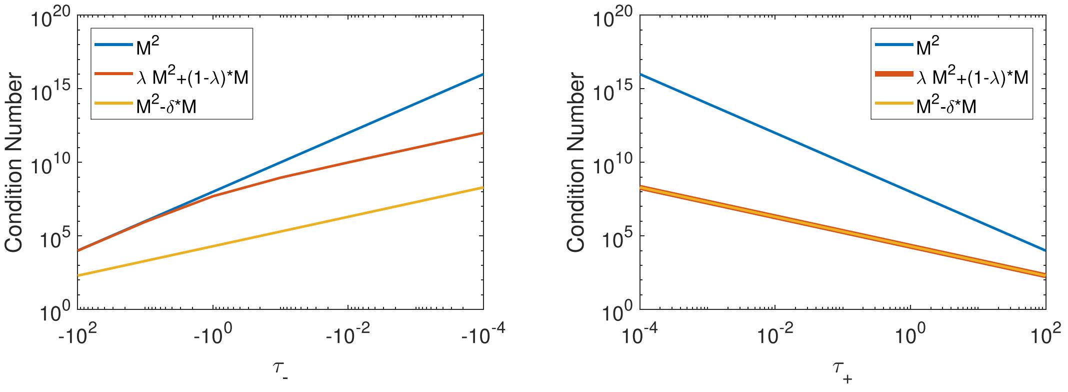
$$\kappa_\delta\left(\widetilde{M}\right) = \frac{\max(q_\delta(L_+), q_\delta(L_-))}{\min(q_\delta(\tau_+), q_\delta(\tau_-))}. \tag{6}$$

The optimal choice of δ, $\delta^* = \arg\min_{\tau_- < \delta < \tau_+} \kappa_\delta\left(\widetilde{M}\right)$ is difficult to determine. We adopt the simple choice $\widehat{\delta} = \tau_+ + \tau_-$, leading to

$$\min(q_{\widehat{\delta}}(\tau_+), q_{\widehat{\delta}}(\tau_-)) = -\tau_-\tau_+.$$

The condition number $\kappa_{\widehat{\delta}}\left(\widetilde{M}\right)$ has similar behavior as $\kappa_{\widehat{\lambda}}\left(\widetilde{M}\right)$ when τ_- is away from zero and τ_+ is close to zero. Different behavior appears when τ_- is closer to zero than τ_+. When τ_+ is orders of magnitude larger than $\left|\tau_-\right|$ and $\max\left(L_+, L_-\right) \gg \tau_+$, the condition number $\kappa_{\widehat{\delta}}\left(\widetilde{M}\right)$ is $\frac{\tau_+}{|\tau_-|}$ times smaller than $\kappa\left(M^2\right)$. Therefore, $q_\delta(x)$ has broader applicable range than $p_\lambda(x)$. [1] In Fig. 3, we demonstrate the advantage of $q_{\widehat{\delta}}(M)$ over $p_{\widehat{\lambda}}(M)$ for some matrix M.

[1] The behavior of $q_\delta(x)$ can be achieved by combining $p_\lambda(x)$ and $p_\lambda(-x)$. The choice of $p_\lambda(x)$ or $p_\lambda(-x)$ depends on spectrum property of M. The resulting numerical method is, however, more complicated than using $q_\delta(x)$ alone.



(a)

(b)

FIG. 3. Condition number of a matrix M with varying asymmetric spectrum. In both (a) and (b), the matrix M has fixed $L_+ = 10^4$, $L_- = -10^4$. (a) fixes the smallest positive eigenvalue $\tau_+ = 10^4 - 1$ and varies τ_-; (b) fixes the largest negative eigenvalue $\tau_- = -10^4 + 1$ and varies τ_+.

REMARK 2.1 Both $q_{\widehat{\delta}}(M)$ and $p_{\widehat{\lambda}}(M)$ take advantage of the asymmetry of the spectrum of M. When the spectrum of M is symmetric around the origin, i.e., $\tau_+ = \tau_-$ and $L_+ = L_-$, the choice of either $q_{\widehat{\delta}}(M)$ or $p_{\widehat{\lambda}}(M)$ falls back to M^2, which has the same condition number as M^*M.

2.2 *BDMC iterative method*

In terms of matrix M as in (3), the two-step iterative method in Prokof'ev & Svistunov (2007) can be written as

$$\text{Step 1:}\quad \left|\widetilde{f}_{k+1}\right\rangle = |b\rangle + (I - M)\left|f_k\right\rangle$$
$$\text{Step 2:}\quad \left|f_{k+1}\right\rangle = \frac{\sum_{j=1}^{k+1} j^t \left|\widetilde{f}_j\right\rangle}{\sum_{j=1}^{k+1} j^t}, \tag{7}$$

where $t > -1$ is a fixedparameter.[2] Step 1 is a fixed-point iteration and Step 2 is a special Nørlund mean with sequence $\{j^t\}$. Let $S_k = \sum_{j=1}^{k} j^t$. We could merge two steps into a single step

$$\left|f_{k+1}\right\rangle = \frac{S_k}{S_{k+1}}\left|f_k\right\rangle + \frac{(k+1)^t}{S_{k+1}}\left|\widetilde{f}_{k+1}\right\rangle = \left|f_k\right\rangle - \frac{(k+1)^t}{S_{k+1}}\left(M\left|f_k\right\rangle - |b\rangle\right). \tag{8}$$

For Hermitian positive definite matrix M, (8) is a gradient descent method for the objective function $\frac{1}{2}\langle f|M|f\rangle - \langle b|f\rangle$ with special stepsize $\alpha_k = \frac{(k+1)^t}{S_{k+1}}$. Such a stepsize asymptotically behaves as

$$\alpha_k \asymp \frac{t+1}{k+1} \quad (k \to +\infty), \tag{9}$$

[2] The notations have been changed from Prokof'ev & Svistunov (2007) as $\alpha \to t$ and $n \to k$ to avoid notation conflicts.

where '$\asymp$' means asymptotically equality, i.e., $f_k \asymp g_k \Leftrightarrow \lim_{k\to+\infty} f_k/g_k = 1$. In fact, the simpler stepsize choice as a modified version of (9) is widely used in the stochastic gradient descent literature. We would denote $\beta = t + 1$ in the following note. The convergence analysis of the iterative method

$$\left|f_{k+1}\right\rangle = \left|f_k\right\rangle - \frac{\beta}{k+\gamma}\left(M\left|f_k\right\rangle - |b\rangle\right) \tag{10}$$

for both deterministic and stochastic M are listed in the next section, where γ is a constant to guarantee the convergence of first iteration.

2.3 *Convergence analysis*

Let us now turn to the convergence analysis of iterative algorithms (8) and (10) for Hermitian positive definite matrix M. Compared to the two, the analysis of (10) would be cleaner due to its simpler choice of stepsize. Prokof'ev & Svistunov (2007) provide asymptotic behavior for $\left|\delta_k\right\rangle = \left|f_k\right\rangle - |f^*\rangle$, which is the difference between step k result $\left|f_k\right\rangle$ and the underlying truth $|f^*\rangle = M^{-1}|b\rangle$. When k approaches $+\infty$ and $\gamma = 1$, $\left|\delta_k\right\rangle$ behaves as

$$e^{-\beta M \log k}\left|\delta_1\right\rangle, \tag{11}$$

where $\left|\delta_1\right\rangle = \left|f_1\right\rangle - |f^*\rangle$ and $\left|f_1\right\rangle$ is the initial guess. Since M is a positive definite matrix and $\beta > 0$, $\left|\delta_k\right\rangle \to |0\rangle$ as $k \to +\infty$. The same asymptotic analysis holds for stepsize (9) as well. According to (11), the slowest converging component behaves as $e^{-\beta\tau_+ \log k}$, where τ_+ is the smallest eigenvalue of M. Careful study of the contraction property of the iterative method shows that either β or the number of non-contraction steps is related to the largest eigenvalue L_+ of M. Overall, smaller condition number of M leads to faster convergence.

Remark 2.2 Based on the above asymptotics, it was suggested in Prokof'ev & Svistunov (2007) the choice of very large β. In that case, for sufficiently large k, the asymptotic rate (11) is achieved. This is, however, only part of the story; as for the iterative method, we are not just interested in asymptotic convergence, the actual decay of error after finite number of steps is more important. Indeed as we will see in Corollary 2.5, the hidden prefactor in (11) depends on β. In particular, the asymptotic analysis fails if t is set as $+\infty$ in the iterative method (8).

Moreover, the asymptotic analysis in (11) only holds for noise-free matrix M. In the stochastic setting, i.e., each evaluation of $M|f\rangle$ involves a stochastic error, we will see in Theorem 2.4 that the expected error is dominated by the stochastic error part. The choice of large β does not impact the convergence rate of the iterative method but enlarges the prefactor. Therefore, choosing large β in the stochastic setting actually has negative influence on the convergence.

The previous asymptotic analysis holds for deterministic matrix M. For stochastic matrix vector multiplication, the analysis is carried out with assumptions on the bias and variance; we present one possible convergence result below, following Theorem 4.7 in the review article Bottou *et al.* (2016).

Let $G(|f\rangle) = \frac{1}{2}\langle f|M|f\rangle - \langle b|f\rangle$ for Hermitian positive definite matrix M. The gradient of $G(|f\rangle)$ is $\nabla G(|f\rangle) = M|f\rangle - |b\rangle$. Hence, both (8) and (10) are gradient descent methods applied to $G(|f\rangle)$. In order to distinguish between the deterministic gradient and stochastic gradient, we denote the stochastic one as $g(|f\rangle, \xi) = M_\xi |f\rangle - |b\rangle$, where ξ is a random variable.

Assumption 2.3 The objective function and stochastic gradient method as (10) satisfies the following:

1. There exist scalars $\mu_G \geqslant \mu > 0$ such that, for all $k \in \mathbb{N}$,

$$\nabla G(|f_k\rangle)^* \mathbb{E}\left[g(|f_k\rangle, \xi_k)\right] \geqslant \mu \|\nabla G(|f_k\rangle)\|_2^2 \quad \text{and}$$
$$\|\mathbb{E}\left[g(|f_k\rangle, \xi_k)\right]\|_2 \leqslant \mu_G \|\nabla G(|f_k\rangle)\|_2.$$

2. There exist scalars $W \geqslant 0$ and $W_V \geqslant 0$ such that, for all $k \in \mathbb{N}$,

$$\mathbb{V}\left[g(|f_k\rangle, \xi_k)\right] = \mathbb{E}\left[\|g(|f_k\rangle, \xi_k)\|_2^2\right] - \|\mathbb{E}\left[g(|f_k\rangle, \xi_k)\right]\|_2^2 \leqslant W + W_V \|\nabla G(|f_k\rangle)\|_2^2.$$

Assumption 2.3 follows Assumption 4.3 in Bottou *et al.* (2016). When $g(|f_k\rangle, \xi_k)$ is an unbiased estimator of $\nabla G(|f_k\rangle)$, both μ_G and μ are one and W_V is zero. We recall the notations $\tau = \tau_+$ and $L = L_+$ as the smallest and largest eigenvalue of M, respectively. Moreover, we denote $W_G = W_V + \mu_G^2$.

Theorem 2.4 Under Assumption 2.3, suppose that the stochastic gradient method is run with a stepsize sequence such that for all $k \in \mathbb{N}$,

$$\alpha_k = \frac{\beta}{\gamma + k}, \quad \beta > \frac{1}{\tau\mu} \quad \text{and } \gamma > 0 \quad \text{such that } \alpha_1 \leqslant \frac{\mu}{LW_G}.$$

Then for all $k \in \mathbb{N}$, the expected optimality gap satisfies

$$\mathbb{E}\left[G(|f_k\rangle) - G_*\right] \leqslant \frac{1}{(\gamma + k)^{\beta\tau\mu}} \left(G(|f_1\rangle) - G_*\right)(\gamma + 1)^{\beta\tau\mu} + \frac{1}{\gamma + k} \frac{\beta^2 LW}{\beta\tau\mu - 1}, \tag{12}$$

where $G_* = \inf_{|f\rangle} G(|f\rangle)$.

Proof. The difference between the objective functions of two consecutive iterations can be bounded as

$$\begin{aligned} G(|f_{k+1}\rangle) - G(|f_k\rangle) =& \nabla G(|f_k\rangle)^* \left(|f_{k+1}\rangle - |f_k\rangle\right) + \frac{1}{2}(\langle f_{k+1}| - \langle f_k|) M (|f_{k+1}\rangle - |f_k\rangle) \\ \leqslant& -\alpha_k \nabla G(|f_k\rangle)^* g(|f_k\rangle, \xi_k) + \frac{\alpha_k^2 L}{2} \|g(|f_k\rangle, \xi_k)\|_2^2. \end{aligned} \tag{13}$$

Taking expectation conditioned on ξ_k on both sides of (13), we obtain

$$\begin{aligned} \mathbb{E}\left[G(|f_{k+1}\rangle)\right] - G(|f_k\rangle) \leqslant& -\alpha_k \nabla G(|f_k\rangle)^* \mathbb{E}\left[g(|f_k\rangle, \xi_k)\right] + \frac{\alpha_k^2 L}{2} \mathbb{E}\left[\|g(|f_k\rangle, \xi_k)\|_2^2\right] \\ \leqslant& -\mu\alpha_k \|\nabla G(|f_k\rangle)\|_2^2 + \frac{\alpha_k^2 L}{2} \mathbb{E}\left[\|g(|f_k\rangle, \xi_k)\|_2^2\right] \\ \leqslant& -\left(\mu - \frac{\alpha_k L W_G}{2}\right) \alpha_k \|\nabla G(|f_k\rangle)\|_2^2 + \frac{\alpha_k^2 LW}{2} \\ \leqslant& -\frac{1}{2}\mu\alpha_k \|\nabla G(|f_k\rangle)\|_2^2 + \frac{1}{2}\alpha_k^2 LW, \end{aligned} \tag{14}$$

where the third inequality adopts Assumption 2.3 and the last inequality is due to the constraints in the theorem. Since M is Hermitian positive definite matrix, we bound $\|\nabla G(|f_k\rangle)\|_2^2$ as follows:

$$\|\nabla G(|f_k\rangle)\|_2^2 = \|M\,|f_k\rangle - M\,|f^*\rangle\|_2^2 \geqslant \tau(\langle f_k| - \langle f^*|)M(|f_k\rangle - |f^*\rangle) = 2\tau(G(|f_k\rangle) - G_*). \tag{15}$$

Substituting (15) into (14), taking total expectation and rearranging, we have

$$\mathbb{E}\left[G(|f_{k+1}\rangle)\right] - G_* \leqslant (1 - \alpha_k \tau\mu)\left(\mathbb{E}\left[G(|f_k\rangle)\right] - G_*\right) + \frac{1}{2}\alpha_k^2 LW. \tag{16}$$

Equation (12) holds for $k = 1$. Then assuming (12) is true for k, it follows from (16) that $(\widehat{k} = \gamma + k)$,

$$\begin{aligned}
\mathbb{E}\left[G(|f_{k+1}\rangle)\right] - G_* \leqslant{}& \left(1 - \frac{\beta\tau\mu}{\widehat{k}}\right)\widehat{k}^{-\beta\tau\mu}\left(G(|f_1\rangle) - G_*\right)(\gamma + 1)^{\beta\tau\mu} \\
&+ \left(1 - \frac{\beta\tau\mu}{\widehat{k}}\right)\frac{1}{\widehat{k}}\frac{\beta^2 LW}{\beta\tau\mu - 1} + \frac{1}{2}\frac{\beta^2}{\widehat{k}^2}LW \\
\leqslant{}& \frac{1}{(\widehat{k}+1)^{\beta\tau\mu}}\left(G(|f_1\rangle) - G_*\right)(\gamma + 1)^{\beta\tau\mu} + \frac{1}{\widehat{k}+1}\frac{\beta^2 LW}{\beta\tau\mu - 1},
\end{aligned}$$

where the last inequality dues to Taylor expansion of $(\widehat{k} + 1)^{-\beta\tau\mu}$ at $\widehat{k}$ and $\frac{\widehat{k}-1}{\widehat{k}^2} < \frac{1}{\widehat{k}+1}$. □

Theorem 2.4 now splits the bound into the convergence of the initial error and stochastic error. Due to the assumption $\beta > \frac{1}{\tau\mu}$, the expected optimality gap is dominated by the stochastic error, which behaves as $O(\frac{1}{\gamma+k})$. At the same time, both the prefactor $\frac{\beta^2 LW}{\beta\tau\mu-1}$ and the parameter of initial stepsize $\gamma \geq \frac{LW_G}{\tau\mu^2} - 1$ relies on the condition number of M, i.e., $\frac{L}{\tau}$. Therefore, the smaller condition number of M leads to the faster convergence in stochastic gradient descent method. A direct corollary can be derived for non-stochastic gradient descent method, where $\mu = \mu_G = 1, W = 0, W_G = 1$.

Corollary 2.5 Suppose the gradient descent method is run with a stepsize sequence such that, for all $k \in \mathbb{N}$,

$$\alpha_k = \frac{\beta}{\gamma + k} \quad \text{and } \gamma > 0 \quad \text{such that } \alpha_1 \leqslant \frac{1}{L}.$$

Then for all $k \in \mathbb{N}$, the optimality gap satisfies

$$G(|f_k\rangle) - G_* \leqslant \frac{1}{(\gamma + k)^{\beta\tau}}\left(G(|f_1\rangle) - G_*\right)(\gamma + 1)^{\beta\tau}.$$

Corollary 2.5 coincides with (11). The impact of the condition number of M to the convergence is more significant in the non-stochastic gradient descent method. The constant $(\gamma + 1)^{\beta\tau}$ and parameter γ are influenced by the condition number in a similar way as the stochastic one. The rate of the convergence of the non-stochastic gradient descent method is also impacted by the smallest eigenvalue of M. In general, the larger of τ leads to faster convergence rate of the gradient descent method. Such an argument agrees with the asymptotic analysis of (11).

3. Alternative stochastic iterative methods

Gradient descent method with stochastic gradient is widely explored in many areas. Especially, machine learning researchers established many variant stochastic gradient descent methods to minimize the loss function in a big data setting. The raise of deep learning further accelerates the development of stochastic gradient descent methods. In these context, many loss functions are non-convex functions. Stochastic gradient descent methods, without accessing full gradient at each step, sample a few components of the full gradient and move along the sampled gradient direction. This strategy reduces the computational cost each step and potentially avoids many local minima. The problem BDMC addresses, in contrast with deep learning, has a quadratic convex objective function. On the other hand, the gradient of the objective function in BDMC can only be accessed via a Monte Carlo procedure. The underlying true gradient is unknown. Here we would like to compare BDMC with a few well-established stochastic gradient descent methods from machine learning literature. The cross-fertilization between machine learning and computational physics is rather natural since both are dealing with high-dimensional problems. In particular, diagrammatic summation methods like BDMC could potentially benefit from other stochastic iterative methods to either improve convergence or allow larger Monte Carlo error.

3.1 *Heavy ball method*

The heavy ball method (Polyak, 1964) adds a momentum term to the gradient descent method

$$\left|f_{k+1}\right\rangle = \left|f_k\right\rangle - \alpha(M\left|f_k\right\rangle - |b\rangle) + \beta\left(\left|f_k\right\rangle - \left|f_{k-1}\right\rangle\right), \tag{17}$$

where α is the stepsize and β is the weight for momentum. The heavy ball method actually has the same convergence rate as the gradient descent method. Unlike the gradient descent method that depends on the condition number of M, the heavy ball method depends on the square root of the condition number of M. Such a property is attractive when M is ill-conditioned. However, due to the momentum, the heavy ball method is not strictly decreasing, i.e., $G(\left|f_{k+1}\right\rangle)$ is not necessarily smaller than $G(\left|f_k\right\rangle)$.

3.2 *Stochastic Barzilai-Borwein method*

In 1988, Barzilai & Borwein (1988) proposed a two-point stepsize gradient method, which is inspired by the secant equation underlying quasi-Newton methods. The stochastic version of the Barzilai and Borwein (sBB) method (Tan *et al.*, 2016), instead of updating the stepsize every iteration with the difference of stochastic gradients, updates the stepsize every m iteration with the difference of aggregated gradients. The detailed iteration is as follows:

$$\begin{aligned}
\alpha_k &= \begin{cases} \alpha_{k-1} & \text{if } k \not\equiv 0 \mod m \\ \frac{1}{m}\frac{\langle \Delta_k^m f | \Delta_k^m f\rangle}{\langle \Delta_k^m f | \Delta_k^m g\rangle} & \text{if } k \equiv 0 \mod m \end{cases} \\
\left|f_{k+1}\right\rangle &= \left|f_k\right\rangle - \alpha_k \nabla G(\left|f_k\right\rangle) \\
\left|g_{k+1}\right\rangle &= \beta \nabla G(\left|f_k\right\rangle) + (1-\beta)\left|g_k\right\rangle,
\end{aligned} \tag{18}$$

where $|\Delta_k^m f\rangle = |f_k\rangle - |f_{k-m}\rangle$ and $|\Delta_k^m g\rangle = |g_k\rangle - |g_{k-m}\rangle$. β is the weight for momentum and m is the updating frequency. Notice that in Tan *et al.* (2016), a smoothing technique is suggested for the stepsize, which is a technique enforce diminishing stepsize. According to our tests, this technique is crucial for the convergence when the gradient is noisy. Therefore, our implementation of sBB adopts the smoothing technique.

3.3 *AdaGrad method*

All previously mentioned methods adopt the same stepsize for all entries of the vector, which might not be favorable in situations when the natural scaling of the entries are different. The adaptive gradient (AdaGrad) method (Duchi *et al.*, 2011) provides a specific entry-dependent adaptive stepsize based on the knowledge of past gradient information. The idea behind AdaGrad replaces the 2-norm in the standard projected gradient method by the Mahalanobis norm, i.e., $\| |f\rangle \|_S^2 = \langle f | S | f \rangle$ for positive semidefinite matrix S and update the S matrix during the iteration. The corresponding iteration becomes

$$|f_{k+1}\rangle = \arg\min_{|f\rangle} \left\| |f\rangle - \left(|f_k\rangle - \alpha S_k^{-1} \nabla G(|f_k\rangle) \right) \right\|_{S_k}^2 . \tag{19}$$

When the upper bound on regret function is minimized under the positivity constraint of S_k and $\mathrm{tr}(S_k) \leq 1$, it can be shown that S_k is of the form

$$S_k = \left(\sum_{i=1}^{k} \nabla G(|f_i\rangle) \nabla G(|f_i\rangle)^* \right)^{\frac{1}{2}} . \tag{20}$$

However, the iteration (19) involves the inverse of S_k, which is computationally expensive. AdaGrad, therefore, replaces S_k by its diagonal part and results the following iterative scheme:

$$|f_{k+1}\rangle = |f_k\rangle - \alpha \mathrm{diag}\left(\Gamma_k\right)^{-1/2} \nabla G(|f_k\rangle), \tag{21}$$

where $\Gamma_k = \sum_{i=1}^{k} \left(\nabla G(|f_i\rangle)\right)^2$ and both the square and the square root are entry-wise operations, and α is a parameter since the computational costs for both the inverse of the square root of a diagonal matrix and the diagonal matrix vector multiplication are the same as generating the gradient vector. Therefore, such a choice of stepsize only increases the computational cost by a small constant, while the convergence could be accelerated for stochastic gradients.

3.4 *Adaptive moment estimation method*

The adaptive moment estimation (ADAM) method (Kingma & Ba, 2015) is a modified version of the AdaGrad method that incorporates momentum acceleration. Instead of using raw gradient $\nabla G(|f_k\rangle)$ and Γ_k as in (21), the ADAM method adds momentum parts for both and corrects the biases. The calculation of its stepsize is more complicated than all prementioned methods. We summarize the calculation as follows:

$$\begin{aligned}
&\left|g_{k+1}\right\rangle = \nabla G(\left|f_k\right\rangle)\\
&\left|m_{k+1}\right\rangle = \beta_1 \left|m_k\right\rangle + (1-\beta_1)\left|g_{k+1}\right\rangle \text{[Update biased first moment]}\\
&\left|v_{k+1}\right\rangle = \beta_2 \left|v_k\right\rangle + (1-\beta_2)\left|g_{k+1}\right\rangle^2 \text{ [Update biased second moment]}\\
&\left|\widehat{m}_{k+1}\right\rangle = \left|m_{k+1}\right\rangle/(1-\beta_1^{k+1})\text{[Correct biased first moment]}\\
&\left|\widehat{v}_{k+1}\right\rangle = \left|v_{k+1}\right\rangle/(1-\beta_2^{k+1})\text{[Correct biased second moment]}\\
&\left|f_{k+1}\right\rangle = \left|f_k\right\rangle - \alpha\left|\widehat{m}_{k+1}\right\rangle/\sqrt{\widehat{v}_{k+1}}
\end{aligned} \tag{22}$$

with initial first moment vector $\left|m_1\right\rangle = |0\rangle$ and second moment vector $\left|v_1\right\rangle = |0\rangle$, where β_1 and β_2 are weights for the first and second moment, respectively, and α is a parameter. These initial first and second moments are crucial for the bias correction parts. The gradient part in ADAM keeps an exponentially decaying average of past gradients and behaves like the heavy ball method. At the same time, unlike AdaGrad keeps the summation over all past squared gradients, ADAM keeps, again, an exponentially decaying average of that. Such a modification reduces the aggressive decaying rate of the stepsize in AdaGrad.

4. Numerical results

This section focuses on testing the performances of different gradient descent methods mentioned in Sections 2.2 and 3. The power of the preconditioning technique has been illustrated in Fig. 3, and we would not retest it here. The algorithms of different gradient descent methods are implemented in MATLAB 2017b and the results reported here are obtained on a MacBook Pro with 2.3 GHz Intel Core i7 and 8 GB memory.

For simplicity, we will use the short name as in Table 1 instead of the original full name. BDMC2 and BDMC3 use the same expression of the stepsize with different setting of γ. The accuracy of all iterative methods is measured against the underlying true solution $|f^*\rangle$ as

$$e_k^{rel} = \frac{\left\| \left|f_k\right\rangle - |f^*\rangle \right\|}{\|f^*\|}, \tag{23}$$

where $\left|f_k\right\rangle$ is the solution at kth step and e_k^{rel} is called the relative error at kth step.

One example is to simulate the DMC by noisy symmetric positive definite matrices M of size 100 by 100. We first generate the simulating system as follows:

$$M = Q \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & 100 \end{bmatrix} Q^*, \tag{24}$$

TABLE 1 *Name convention for iterative methods. BDMC2 and BDMC3 use the same expression of the stepsize with different setting of γ*

Short name	Full name	Scheme
GD	Gradient descent method	
BDMC	Bold Diagrammatic Monte Carlo method	Section 2.2 (7)
BDMC2	Bold Diagrammatic Monte Carlo method	Section 2.2 (10)
BDMC3	Bold Diagrammatic Monte Carlo method	Section 2.2 (10)
HB	Heavy ball method	Section 3.1
sBB	Stochastic Barzilai–Borwein method	Section 3.2
AdaGrad	Adaptive gradient method	Section 3.3
ADAM	Adaptive moment estimation method	Section 3.4

TABLE 2 *'Close-to-optimal' parameters of iterative methods obtained empirically for the test example*

	GD	BDMC	BDMC2/3	HB		sBB		AdaGrad	ADAM		
ϵ	α	t	β	α	β	m	β	α	α	β_1	β_2
0.01	0.005	−0.6	0.4	0.01	0.6	50	0.9	0.9	0.1	0.99	0.999
0.1	0.001	−0.6	0.4	0.001	0.6	50	0.4	0.4	0.01	0.99	0.999
1	0.0005	−0.5	0.5	0.0005	0.6	30	0.5	0.2	0.001	0.99	0.999
10	0.0003	−0.5	0.5	0.0001	0.6	40	0.5	0.2	0.002	0.99	0.999
100	0.00008	−0.4	0.5	0.00005	0.6	50	0.4	0.2	0.002	0.99	0.999

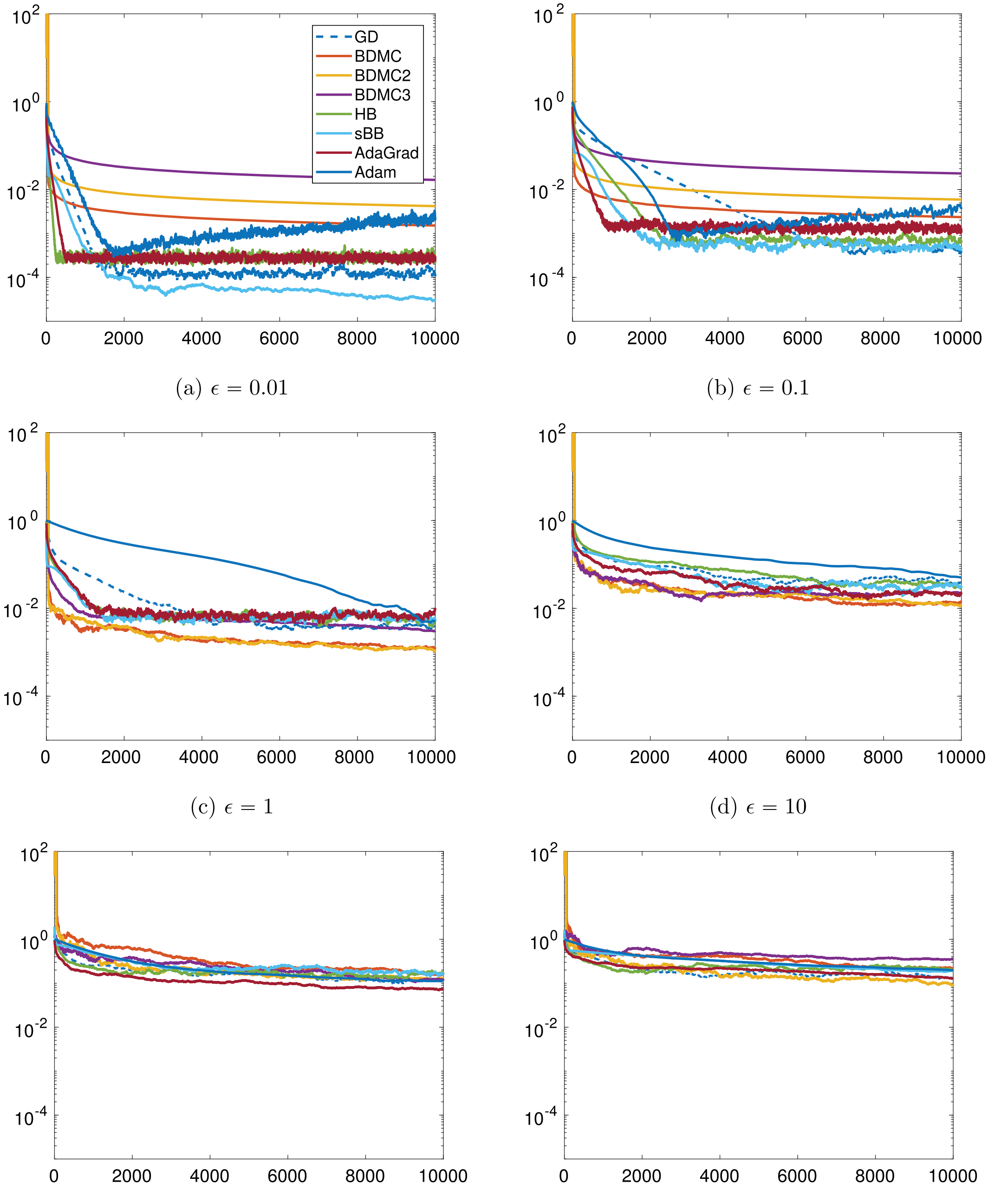
where Q is a random unitary matrix. Then an underlying true solution $|f^*\rangle$ is generated from normal distribution. The vector $|b\rangle$, therefore, is the multiplication of M and $|f^*\rangle$, i.e., $|b\rangle = M\,|f^*\rangle$. In order to simulate the uncertainty of the DMC, we added noise to each entry of $M\,|f\rangle$, i.e.,

$$g(|f\rangle\,,\xi) = M\,|f\rangle - |b\rangle + \epsilon\,|\xi\rangle\,, \tag{25}$$

where $|\xi\rangle$ is a random vector with each entry drawn from standard normal distribution and ϵ is the noise level.

Each different gradient descent method in Sections 2.2 and 3 has some parameters in common, maximum number of iteration is 10000, convergence tolerance is 10^{-6} and initial guess $\left|f_1\right\rangle$ is a random vector with entry drawn from standard normal distribution (except that ADAM always starts from $|0\rangle$). Besides these common parameters, these methods have their own parameters requiring tuning. For each method, we tried different settings and summarize the close-to-optimal parameter in Table 2 up to one significant digits. 'Close-to-optimal' is in the sense that the averaged relative error $\frac{\||f_{10000}\rangle - |f^*\rangle\|}{\||f^*\rangle\|}$ of 10 runs is minimized. BDMC2 sets $\gamma = 1$, which is asymptotically equal to the stepsize of BDMC, whereas BDMC3 sets γ to be the smallest integer satisfying the assumption in Theorem 2.4.

Figure 4 illustrates the performance of different stochastic gradient descent methods with different levels of noise in the gradient. According to Fig. 4 (a) and (b), which correspond to low noise cases, the sBB method outperforms other methods. Although it is not the best method in the first 2000 iterations, it achieves the best relative error when the iteration number is getting larger. In these low noise cases, BDMC and BDMC2 do not perform well comparing to other methods. While in the medium noise cases (see Fig. 4 (c) and (d)), BDMC and BDMC2 are the best methods among all. They outperform other methods from the very beginning of the iterations. Based on our numerical tests, Fig. 4 (a), (b),



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.1$

(c) $\epsilon = 1$

(d) $\epsilon = 10$

(e) $\epsilon = 100$

(f) $\epsilon = 100$

FIG. 4. Convergence results for different methods and different noise level. For all subfigures, y-axis denotes the relative error as in (23) and x-axis denotes the iteration number. The legends for (b)–(f) are the same as that in (a).

(c) and (d) are relatively robust with respect to different runs of the algorithms. However, as the noise level being close to the largest eigenvalue of M, the test results are no longer robust. The ranking of the methods shifts randomly. For example, Fig. 4 (e) and (f) are two runs of the methods at the same noise level $\epsilon = 100$. In (e), BDMC is the worst method, whereas in (f) it is one of the best. Therefore, the conclusion of the ranking of the methods cannot be made here for the high noise level case.

We also would like to raise one concern about the BDMC and BDMC2 methods. For both of them, the relative errors ‘blow up’ in the first few iterations and quickly drop down to a reasonable level. Moreover, the peaks of the ‘blow-ups’ could be as large as 10^{10} for the example here. These peaks are also related to the condition number of the original matrix M. The worse the condition number of M, the higher the peak. At the same time, the parameter t must be tuned to close to -1 (β in BDMC2 be tuned to close to 0) to enable the drop-down behavior and achieve convergence. Relating this phenomena to the assumptions in Theorem 2.4, we notice that the parameters in both BDMC and BDMC2 violate the assumption of α_1. When the assumptions is fully filled, as in BDMC3, such a ‘blow up’ behavior can be avoided. Unfortunately, the performance of BDMC3 in many cases is not as good as either BDMC or BDMC2.

The last point is about the sensitivity of the parameters. The parameters of gradient descent method and heavy ball method are the most sensitive ones, i.e., small change in the parameters would result huge performance difference. On the other side, the parameters in the sBB method and AdaGrad method are the least sensitive ones. Although in Table 2, their parameters vary a lot, but many other choices of their parameters actually show similar convergence behavior. Therefore, these two methods are easier to use in practice.

5. Conclusion

This note provides mathematical understanding of the original BDMC method in Prokof’ev & Svistunov (2007). The two parts in the BDMC method are interpreted as the preconditioning part and stochastic iterative part. In the preconditioning part, a quadratic polynomial of the matrix, $p_{\widehat{\lambda}}(M)$, turns an indefinite matrix into a positive definite matrix and the corresponding condition number could potentially be orders of magnitudes smaller than the traditional preconditioning technique M^2. In addition to $p_{\widehat{\lambda}}(M)$, we propose another quadratic polynomial $q_{\widehat{\delta}}(M)$ that has the same performance on some matrices and achieves better performance on another big group of matrices. For the second part, the stochastic iterative part, we rewrite the original multi-step BDMC method as a gradient descent method with diminishing stepsize (8). Asymptotically, the complicated stepsize can be replaced by (10). The choices of both stepsizes behave similar on all numerical examples we have tested. Due to the DMC procedure involved in the evaluation of the matrix, the BDMC iterative method is actually a stochastic gradient descent method on quadratic objective function. Naturally, we introduce a few stochastic gradient descent methods from machine learning and deep learning, which are originally designed for non-convex objective functions. All these stochastic gradient descent methods are tested on a simulated toy example with different level of noises. In the small noise levels,sBB shows great power over other method. For medium noise level close to the smallest eigenvalue of the matrix, BDMC methods (with two choices of stepsize) outperform other methods. When the noise level is as large as the largest eigenvalue of the matrix, the conclusion for the performance of methods is unclear.

At this point, the new preconditioning technique and variant stochastic gradient descent methods are only tested on simulated matrices with noise. In the future, we would like to apply all these techniques and methods to the actual physical systems of interest using BDMC method such that a more conclusive statement for the performance of all suggested alternatives can be drawn.

Acknowledgements

We thank Lexing Ying for interesting discussions regardings the BDMC method.

Funding

This work is partially supported by the National Science Foundation (OAC-1450280 and DMS-1454939).

References

Austin, B. M., Zubarev, D. Y. & Lester Jr, W. A. (2012) Quantum Monte Carlo and related approaches. *Chem. Rev.*, **112**, 263–288.

Barzilai, J. & Borwein, J. M. (1988) Two-point step size gradient methods. *IMA J. Numer. Anal.*, **8**, 141–148.

Bottou, L., Curtis, F. E. & Nocedal, J. (2018) Optimization methods for large-scale machine learning, *SIAM Rev.*, **60**, 223–311.

Ceperley, D. M. (2010) An overview of quantum Monte Carlo methods. *Rev. Mineral. Geochem.*, **71**, 129–135.

Duchi, J., Hazan, E. & Singer, Y. (2011) Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, **12**, 2121–2159.

Fetter, A. L. & Walecka, J. D. (2003) *Quantum Theory of Many-Particle Systems*. Mineola, NY: Dover.

Foulkes, W. M. C., Mitas, L., Needs, R. J. & Rajagopal, G. (2001) Quantum Monte Carlo simulations of solids. *Rev. Mod. Phys.*, **73**, 33–83.

Kingma, D. P. & Ba, J. (2015) Adam: a method for stochastic optimization. *3rd International Conference for Learning Representations*. San Diego, Ithaca, NY: arXiv.org.

Kolorenc, J. & Mitas, L. (2011) Applications of quantum Monte Carlo methods in condensed systems. *Rep. Prog. Phys.*, **4**, 026502.

Kulagin, S. A., Prokof'ev, N. V., Starykh, O. A., Svistunov, B. V. & Varney, C. N. (2013) Bold diagrammatic Monte Carlo method applied to Fermionized frustrated spins. *Phys. Rev. Lett.*, **110**, 070601.

LeBlanc, J. P. F., Antipov, A. E., Becca, F., Bulik, I. W., Chan, G. K.-L., Chung, C.-M., Deng, Y., Ferrero, M., Henderson, T. M., Jiménez-Hoyos, C. A., Kozik, E., Liu, X.-W., Millis, A. J., Prokof'ev, N. V., Qin, M., Scuseria, G. E., Shi, H., Svistunov, B. V., Tocchio, L. F., Tupitsyn, I. S., White, S. R., Zhang, S., Zheng, B.-X., Zhu, Z. & Gull, E. (2015) Solutions of the two-dimensional Hubbard model: benchmarks and results from a wide range of numerical algorithms. *Phys. Rev. X*, **5**, 041041-1–041041-28.

Mattuck, R. D. (1992) *A Guide to Feynman Diagrams in the Many-Body Problem: Second Edition*. New York: Dover.

Paige, C. C. & Saunders, M. A. (1975) Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, **12**, 617–629.

Polyak, B. T. (1964) Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.*, **4**, 1–17.

Prokof'ev, N. V. & Svistunov, B. V. (2007) Bold diagrammatic Monte Carlo technique: when the sign problem is welcome. *Phys. Rev. Lett.*, **99**, 250201.

Prokof'ev, N. V. & Svistunov, B. V. (2008a) Bold diagrammatic Monte Carlo: a generic sign-problem tolerant technique for polaron models and possibly interacting many-body problems. *Phys. Rev. B*, **77**, 125101.

Prokof'ev, N. V. & Svistunov, B. V. (2008b) Fermi-polaron problem: diagrammatic Monte Carlo method for divergent sign-alternating series. *Phys. Rev. B*, **77**, 020408.

Saad, Y. & Schultz, M. H. (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**, 856–869.

Tan, C., Ma, S., Dai, Y.-H. & Qian, Y. (2016) Barzilai–Borwein step size for stochastic gradient descent. *Advances in Neural Information Processing* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon& R. Garnett eds). Red Hook, NY: Curran Associates, Inc., pp. 685–693.

Van Houcke, K., Kozik, E., Prokof'ev, N. & Svistunov, B. (2008) Diagrammatic Monte Carlo. *Computer Simulation Studies in Condensed Matter Physics XXI* (D. P. Laudan, S. P. Lewis & H. B. Schuttler eds). Berlin Heidelberg: Springer.

Van Houcke, K., Werner, F., Kozik, E., Prokof'ev, N. V., Svistunov, B. V., Ku, M. J. H., Sommer, A. T., Cheuk, L. W., Schirotzek, A. & Zwierlein, M. W. (2012) Feynman diagrams versus Fermi-gas Feynman emulator. *Nat. Phys.*, **8**, 366–370.