Fast randomized iteration: diffusion Monte Carlo through the lens of numerical linear algebra

Lek-Heng Lim[1] and Jonathan Weare[1,2]

[1]Department of Statistics, University of Chicago
[2]James Franck Institute, University of Chicago

Abstract

We review the basic outline of the highly successful diffusion Monte Carlo technique commonly used in contexts ranging from electronic structure calculations to rare event simulation and data assimilation, and show that aspects of the scheme can be extended to address a variety of common tasks in numerical linear algebra. From the point of view of numerical linear algebra, the main novelty of the new algorithms is that they work in either linear or constant cost per iteration (and in total, under appropriate conditions) and are rather versatile: In this article, we will show how they apply to solution of linear systems, eigenvalue problems, and matrix exponentiation, in dimensions far beyond the present limits of numerical linear algebra. In fact, the schemes that we propose are inspired by recent DMC based quantum Monte Carlo schemes that have been applied to matrices as large as $10^{108} \times 10^{108}$. We will also provide convergence results and discuss the dependence of these results on the dimension of the system. For many problems one can expect the total cost of the schemes to be sub-linear in the dimension of the problem. So while traditional iterative methods in numerical linear algebra were created in part to deal with instances where a matrix (of size $\mathcal{O}(n^2)$) is too big to store, the adaptations of DMC that we propose are intended for instances in which even the solution vector itself (of size $\mathcal{O}(n)$) may be too big to store or manipulate.

1 Introduction

Randomized algorithms have become immensely popular in numerical linear algebra over the last decade. Most of the developments to date have focused on low-rank approximations of large matrices (see e.g. [15, 19, 20, 21, 23, 37, 39, 46, 47, 56]). For earlier uses of randomization in numerical linear algebra see, for example, [1] (in the context of matrix inversion) and [32] (for estimates of the trace of a matrix), and for an interesting description of the relationships between Markov Chain Monte Carlo schemes and common iterative techniques in numerical linear algebra see [27]. The goal of this article is to, after providing a brief introduction to the highly successful *diffusion Monte Carlo* (DMC) algorithm, suggest a new class of algorithms inspired by DMC for problems in numerical linear algebra. DMC is used in applications including electronic structure calculations, rare event simulation, and data assimilation, to efficiently approximate expectations of the type appearing in Feynman–Kac formulae, i.e., for weighted expectations of Markov processes typically associated with parabolic partial differential equations (see e.g. [17]). While based on principles underlying DMC, the fast randomized iteration (FRI) schemes that we study in this article are designed to address arguably the most classical and common tasks in matrix computations: linear systems, eigenvector problems, and matrix exponentiation, i.e., solving for v in

$$Av = b, \qquad Av = \lambda v, \qquad v = \exp(A)b \tag{1}$$

for matrices A that might not have any natural association with a Markov process.

FRI schemes rely on basic principles similar to those at the core of other methods that have appeared recently in the numerical linear algebra literature but they differ substantially in detail and in the problems they address. These differences will be remarked on again later, but roughly, while most of the randomized linear algebra techniques in the references above rely on a single sampling step, FRI methods randomize repeatedly and, as a consequence, require more carefully constructed randomizations. FRI schemes are not, however, the first to employ randomization within iterative schemes. In fact the strategy of replacing expensive integrals or sums (without immediate stochastic interpretation) appearing in iterative protocols has a long history. For example, it was used in schemes for the numerical solution of hyperbolic systems of partial differential equations in [12]. That strategy is represented today in applications ranging from density functional calculations in physics and chemistry (see e.g. [3]) to maximum likelihood estimation in statistics and machine learning (see e.g. [9]). Though related in that they rely on repeated randomization within an iterative scheme, these schemes differ from the methods we consider in that they do not use a stochastic representation of the solution vector itself. In contrast to these and other randomized methods that have been used in linear algebra applications, our focus is on problems for which the solution vector is extremely large so they can only be treated by linear or constant cost algorithms. In fact, the scheme that is our primary focus is ideally suited to problems so large that the solution vector itself is too large to store so that no traditional iterative method (even for sparse matrices) is appropriate. This is possible because the scheme computes only low dimensional projections of the solution vector and not the solution vector itself. The full solution is replaced by a sequence of sparse random vectors whose expectations are close the true solution and whose variances are small.

Diffusion Monte Carlo (see e.g. [2, 8, 11, 24, 29, 30, 33, 35, 38]) is a central component in the quantum Monte Carlo (QMC) approach to computing the electronic ground state energy of the Schrödinger–Hamiltonian operator [1]

$$\mathcal{H}v = -\frac{1}{2}\Delta v + U\,v. \tag{2}$$

We are motivated in particular by the work in [8] in which the authors apply a version of the DMC procedure to a finite (but huge) dimensional projection of $\mathcal{H}$ onto a discrete basis respecting an anti-symmetry property of the desired eigenfunction. The approach in [8] and subsequent papers have yielded remarkable results in situations where the projected Hamiltonian is an extremely large matrix (e.g. $10^{108} \times 10^{108}$, see [51]) and standard approaches to finite dimensional eigenproblems are far from reasonable (see [5, 6, 7, 13, 14, 51]).

The basic DMC approach is also at the core of schemes developed for a number of applications beyond electronic structure. In fact, early incarnations of DMC were used in the simulation of small probability events [31, 48] for statistical physics models. Also in statistical physics, the transfer matrix Monte Carlo (TMMC) method was developed to compute the partition functions of certain lattice models by exploiting the observation that the partition function can be represented as the dominant eigenvalue of the so-called transfer matrix, a real, positive, and sparse matrix (see [41]). TMMC may be regarded as an application of DMC to discrete eigenproblems. DMC has also become a popular method for many data assimilation problems and the notion of a "compression" operation introduced below is very closely related to the "resampling" strategies developed for those problems (see e.g. [28, 34, 16]).

[1] The symbol Δ is used here and below to denote the usual Laplacian operator on functions of $\mathbb{R}^d$, $\Delta u = \sum_{i=1}^{d} \partial_{x_i}^2 u$. U is a potential function that acts on v by pointwise multiplication. Though this operator is symmetric, the FRI schemes we introduce below are not restricted to symmetric eigenproblems.

One can view the basic DMC (or Markov chain Monte Carlo for that matter) procedure as a combination of two steps: In one step an integral operator (a Green's function) is applied to an approximate solution consisting of a weighted finite sum of delta functions, in another step the resulting function (which is no longer a mixture of delta functions) is again approximated by a finite mixture of delta functions. The more delta functions allowed in the mixture, the higher the accuracy and cost of the method. A key to understanding the success of these methods is the observation that not *all* delta functions (i.e., at all positions in space) need appear in the mixture. A similar observation holds for the methods we introduce: FRI schemes need not access all entries in the matrix of interest to yield an accurate solution. In fact, we prove that the cost to achieve a fixed level of accuracy with our methods can be bounded independently of the size of the matrix, though in many applications one should expect some dependence on dimension. As with other randomized schemes, when an effective deterministic method is available it will very likely outperform the methods we propose; our focus is on problems for which satisfactory deterministic alternatives are not available (e.g. when the size of the intermediate iterates or final result are so large as to prohibit any conceivable deterministic methods). Moreover, the schemes that we propose are a supplement and not a replacement for traditional dimensional reduction strategies (e.g. intelligent choice of basis). Indeed, successful application of DMC within practical QMC applications relies heavily on a change of variables based on approximations extracted by other methods (see the discussion of importance sampling in [24]).

The theoretical results that we provide are somewhat atypical of results commonly presented in the numerical linear algebra literature. In the context of linear algebra applications, both DMC and FRI schemes are most naturally viewed as randomizations of standard iterative procedures and their performance is largely determined by the structure of the particular deterministic iteration being randomized. For this reason, as well as to avoid obscuring the essential issues with the details of individual cases, we choose to frame our results in terms of the difference between the iterates v_t produced by a general iterative scheme and the iterates generated by the corresponding randomized scheme V_t (rather than considering the difference between V_t and $\lim_{t\to\infty} v_t$). In ideal situations our bounds are of the form

$$|||V_t - v_t||| = \sup_{f\in\mathbb{C}^n,\, \|f\|_1\leq 1} \sqrt{\mathbf{E}\left[|f\cdot V_t - f\cdot v_t|^2\right]} \leq \frac{C}{\sqrt{m}}$$

where the constant C is independent of the dimension n of the problem and $m \leq n$ controls the cost per iteration of the randomized scheme (one iteration of the randomized scheme is roughly a factor of n/m less costly than its deterministic counterpart and the two schemes are identical when $m = n$). The norm in the above display measures the root mean squared deviation in low dimensional projections of the iterates. This choice is important as described in more detail in Sections 3 and 4. For more general applications, one can expect the constant C, which incorporates the stability properties of the randomized iteration, to depend on dimension. In the worst case scenario, the randomized scheme is no more efficient than its deterministic counterpart (in other words, reasonable performance may require $m \sim n$). Our numerical simulations, in which n/m ranges roughly between 10^7 and 10^{14}, strongly suggest that this scenario may be rare.

We will begin our development in Section 2 with a description of the basic diffusion Monte Carlo procedure. Next, in Section 3 we describe how ideas borrowed from DMC can be applied to general iterative procedures in numerical linear algebra. As a prototypical example, we describe how randomization can be used to dramatically decrease the cost of finding the dominant eigenvalue and (projections of) the dominant eigenvector, first to $\mathcal{O}(n)$ in the dimension of the problem and then to $\mathcal{O}(1)$. Also in that section, we consider the specific case in which the iteration mapping is an ε-perturbation of the identity, relevant to a wide range of applications involving evolutionary differential equations. In this case a poorly chosen randomization scheme can result in an unstable

algorithm while a well chosen randomization can result in an error that decreases with ε (over ε^{-1} iterations). Next, in Sections 4 and 5, we establish several simple bounds regarding the stability and error of our schemes. Finally, in Section 6 we provide three computational examples to demonstrate the performance of our approach. A simple, educational implementation of Fast Randomized Iteration applied to our first computational example can be found here [55].

Remark 1. In several places we have included remarks that clarify or emphasize concepts that may otherwise be unclear to readers more familiar with classical, deterministic, numerical linear algebra methods. We anticipate that some of these remarks will be useful to this article's broader audience as well.

2 Diffusion Monte Carlo within quantum Monte Carlo

The ground state energy, λ_*, of a quantum mechanical system governed by the Hamiltonian in (2) is the smallest eigenvalue (with corresponding eigenfunction v_*) of the Hermitian operator $\mathcal{H}$. The starting point for a DMC calculation is the imaginary-time Schrödinger equation[2] [3]

$$\partial_t v = -\mathcal{H} v \tag{3}$$

(for a review of QMC see [24]). One can, in principle, use power iteration to find λ_*: beginning from an initial guess v_0 (and assuming a gap between λ_* and the rest of the spectrum of $\mathcal{H}$), the iteration

$$\lambda_t = -\frac{1}{\varepsilon} \log \int e^{-\varepsilon \mathcal{H}} v_{t-1}(x)\, dx \quad \text{and} \quad v_t = \frac{e^{-\varepsilon \mathcal{H}} v_{t-1}}{\int e^{-\varepsilon \mathcal{H}} v_{t-1}(x)\, dx} \tag{4}$$

will converge to the pair (λ_*, v_*) where v_* is the eigenfunction corresponding to λ_*.

Remark 2. For readers who are more familiar with the power method in numerical linear algebra, this may seem a bit odd but a discrete analogue of (4) is just $v_t = Av_{t-1}/\|Av_{t-1}\|_1$ applied to a positive definite matrix $A = \exp(-\varepsilon H) \in \mathbb{C}^{n\times n}$ where H is Hermitian[4]. The slight departure from the usual power method is only in (i) normalizing by a 1-norm (or rather, by the sum of entries $\mathbb{1}^{\mathrm{T}} A v_{t-1}$ since both v and A are non-negative), and (ii) iterating on $\exp(-\varepsilon H)$ instead of on H directly (the goal in this context is to find the smallest eigenvalue of H not the magnitude-dominant eigenvalue of H). The iteration on the eigenvalue is then $\lambda_t = -\epsilon^{-1} \log\|Av_{t-1}\|_1$ since $\lambda_*(H) = -\epsilon^{-1} \log \lambda_*(A)$.

The first step in any (deterministic or stochastic) practical implementation of (4) is discretization of the operator $e^{-\varepsilon \mathcal{H}}$. Diffusion Monte Carlo often uses the second order time discretization

$$e^{-\varepsilon \mathcal{H}} \approx K_\varepsilon = e^{-\frac{\varepsilon}{2} U} e^{\frac{\varepsilon}{2} \Delta} e^{-\frac{\varepsilon}{2} U}.$$

A standard deterministic approach would then proceed by discretizing the operator K_ε in space and replacing $e^{-\varepsilon \mathcal{H}}$ in (4) with the space and time discretized approximate operator. The number of spatial discretization points required by such a scheme to achieve a fixed accuracy will, in general, grow exponentially in the dimension d of the argument of $\mathcal{H}$.

[2] The reader familiar with quantum mechanics but unfamiliar with QMC may wonder why we begin with the imaginary time Schrödinger equation and not the usual Schrödinger equation $i\partial_t v = -\mathcal{H} v$. The reason is that while the solutions to both equations can be expanded in terms of the eigenfunction of $\mathcal{H}$, for the usual Schrödinger equation the contributions to the solution from eigenfunctions with larger eigenvalues do not decay relative to the ground state. By approximating the solution to the real time equation for large times we can approximate the ground state eigenfunction of $\mathcal{H}$.

[3] In practical QMC applications one solves for the $\rho = v_* \tilde{v}$ where $\tilde{v}$ is an approximate solution found in advance by other methods. The new function ρ is the ground state eigenfunction of a Hamiltonian of the form $\tilde{\mathcal{H}} v = -\frac{1}{2}\Delta v + \mathrm{div}(bv) + \tilde{U} v$. The implications for the discussion in this section are minor.

[4] Note that H, which is a matrix is not the same as $\mathcal{H}$ which is an operator on a function space and is only introduced for the purposes of relating the expression in (4) to the usual power method for matrices

Diffusion Monte Carlo uses two randomization steps to avoid this explosion in cost as d increases. These randomizations have the effect of ensuring that the random approximations V_t^m of the iterates v_t are always of the form

$$V_t^m(x) = \sum\nolimits_{j=1}^{N_t} W_t^{(j)} \delta_{X_t^{(j)}}(x)$$

where $\delta_y(x)$ is the Dirac delta function centered at $y \in \mathbb{R}^d$, the $W_t^{(j)}$ are real, non-negative numbers with $\sum_{j=1}^{N_t} W_t^{(j)} = 1$, and, for each $j \leq N_t$, $X_t^{(j)} \in \mathbb{R}^d$. As will be made clear in a moment, the integer m superscripted in our notation controls the number of delta functions, N_t, included in the above expression for V_t^m.

The fact that the function V_t^m is non-zero at only N_t values is crucial to the efficiency of diffusion Monte Carlo. Starting with $N_0 = m$ and from an initial condition of the form

$$V_0^m = \frac{1}{m} \sum\nolimits_{j=1}^{m} \delta_{X_0^{(j)}}$$

the first factor of $e^{-\frac{\varepsilon}{2}U}$ applied to V_0^m results in

$$\frac{1}{m} \sum\nolimits_{j=1}^{m} e^{-\frac{\varepsilon}{2}U(X_0^{(j)})} \delta_{X_0^{(j)}}$$

which can be assembled in $\mathcal{O}(m)$ operations. The first of the randomization steps used in DMC relies on the well known relationship

$$\int f(x)[e^{\frac{\varepsilon}{2}\Delta}\delta_y](x)\,dx = \mathbf{E}_y\left[f(B_\varepsilon)\right] \tag{5}$$

where f is a test function, B_s is a standard Brownian motion evaluated at time $s \geq 0$, and the subscript on the expectation is meant to indicate that $B_0 = y$ (i.e. in the expectation in (5) B_ε is a Gaussian random variable with mean y and variance ε). In fact, this representation is a special case of the Feynman–Kac formula $\int f(x)[e^{-\varepsilon\mathcal{H}}\delta_y](x)\,dx = \mathbf{E}_y\big[f(B_\varepsilon)e^{-\int_0^\varepsilon U(B_s)\,ds}\big]$. Representation (5) suggests the approximation

$$K_\varepsilon V_0^m \approx \tilde{V}_1^m = \frac{1}{m} \sum\nolimits_{j=1}^{m} e^{-\frac{\varepsilon}{2}\left(U(\xi_1^{(j)})+U(X_0^{(j)})\right)} \delta_{\xi_1^{(j)}}$$

where, conditioned on the $X_0^{(j)}$, the $\xi_1^{(j)}$ are independent and $\xi_1^{(j)}$ is normally distributed with mean $X_0^{(j)}$ and covariance εI (here I is the $d \times d$ identity matrix). This first randomization has allowed an approximation of $K_\varepsilon V_0^m$ by a distribution, $\tilde{V}_1^m$, that is again supported on only m points in $\mathbb{R}^d$. One might, therefore, attempt to define a sequence V_t^m iteratively by the recursion

$$V_{t+1}^m = \frac{\tilde{V}_{t+1}^m}{\int V_{t+1}^m(x)\,dx} = \sum\nolimits_{j=1}^{m} W_{t+1}^{(j)}\, \delta_{\xi_{t+1}^{(j)}}$$

where we have recursively defined the weights

$$W_{t+1}^{(j)} = \frac{e^{-\frac{\varepsilon}{2}\left(U(\xi_{t+1}^{(j)})+U(X_t^{(j)})\right)} W_t^{(j)}}{\sum_{\ell=1}^{m} e^{-\frac{\varepsilon}{2}\left(U(\xi_{t+1}^{(\ell)})+U(X_t^{(\ell)})\right)} W_t^{(\ell)}}$$

with $W_0^{(j)} = 1/m$ for each j. The cost of this randomization procedure is $\mathcal{O}(dm)$ so that the total cost of a single iteration is $\mathcal{O}(dm)$.

At each step the weights in the expression for the iterates V_t^m are multiplied by additional random factors. These factors are determined by the potential U and the positions of the $\xi_t^{(j)}$. On the other hand, the $\xi_t^{(j)}$, evolve without reference to the potential function U (they are discretely sampled points from m independent Brownian motions). As a consequence, over many iterations one can expect extreme relative variations in the $W_t^{(j)}$ and, therefore, poor accuracy in V_t^m as an approximation of the functions v_t produced by (4).

The second randomization employed by the DMC algorithm is the key to controlling the growth in variance and generalizations of the idea will be key to designing fast randomized iteration schemes in the next section. In order to control the variation in weights, at step t, DMC randomly removes points $\xi_t^{(j)}$ corresponding to small weights $W_t^{(j)}$ and duplicates points corresponding to large weights. A new distribution Y_t^m is generated from V_t^m so that

$$\mathbf{E}\left[Y_t^m \,|\, V_t^m\right] = V_t^m$$

by "resampling" a new collection of m points from the $\xi_t^{(j)}$ with associated probabilities $W_t^{(j)}$. The resulting points are labeled $X_t^{(j)}$ and the new distribution Y_t^m takes the form

$$Y_t^m = \frac{1}{N_t} \sum\nolimits_{j=1}^{N_{t+1}} \delta_{X_t^{(j)}}.$$

The next iterate V_{t+1}^m is then built exactly as before but with V_t^m replaced by Y_t^m. All methods to select the $X_t^{(j)}$ generate, for each j, a non-negative integer $N_t^{(j)}$ with

$$\mathbf{E}\big[N^{(j)} \;\big|\; \{W_t^{(\ell)}\}_{\ell=1}^m\big] = mW_t^{(j)}$$

and then sets $N_t^{(j)}$ of the elements in the collection $\{X_t^{(j)}\}_{j=1}^{N_{t+1}}$ equal to $\xi_t^{(j)}$ so that $N_{t+1} = \sum_{j=1}^{N_t} N_t^{(j)}$. For example, one popular strategy in DMC generates the $N_t^{(j)}$ independently with

$$\mathbf{P}\big[N_t^{(j)} = \lfloor mW_t^{(j)} \rfloor\big] = \lceil mW_t^{(j)} \rceil - mW_t^{(j)}, \qquad \mathbf{P}\big[N_t^{(j)} = \lceil mW_t^{(j)} \rceil\big] = mW_t^{(j)} - \lfloor mW_t^{(j)} \rfloor. \quad (6)$$

The above steps define a randomized iterative algorithm to generate approximations V_t^m of v_t. The second randomization procedure (generating Y_t^m from V_t^m) will typically require $\mathcal{O}(m)$ operations, preserving the overall $\mathcal{O}(dm)$ per iteration cost of DMC (as we have described it). The memory requirements of the scheme are also $\mathcal{O}(dm)$. The eigenvalue λ_* can be approximated, for example, by

$$-\frac{1}{\varepsilon} \log\left(\frac{1}{T}\sum\nolimits_{t=1}^{T} \frac{1}{N_t}\sum\nolimits_{j=1}^{N_t} e^{-\frac{\varepsilon}{2}\left(U(\xi_{t+1}^{(j)}) + U(X_t^{(j)})\right)}\right)$$

for T large.

Before moving on to more general problems notice that the scheme just outlined applies just as easily to space-time discretizations of $e^{-\varepsilon\mathcal{H}}$. For example, if we set $h = \sqrt{(1+2d)\varepsilon}$ and denote by $\mathcal{E}_h^d \subset \mathbb{R}^d$ the uniform rectangular grid with resolution h in each direction, the operator $e^{-\varepsilon\mathcal{H}}$ can be discretized using

$$e^{-\varepsilon\mathcal{H}} \approx K_{\varepsilon,h} = e^{-\frac{\varepsilon}{2}U} e^{\frac{\varepsilon}{2}\Delta_h} e^{-\frac{\varepsilon}{2}U}$$

where, for any vector $g \in \mathcal{E}_h^d$,

$$\Delta_h g(x) = \frac{1}{\varepsilon}\left(-g(x) + \frac{1}{1+2d}\sum\nolimits_{y\in\mathcal{E}_h^d,\, \|y-x\|_2 \le h} g(y)\right)$$

(here we find it convenient to identify functions $g : \mathcal{E}_h^d \to \mathbb{R}$ and vectors in $\mathbb{R}^{\mathcal{E}_h^d}$). The operator $e^{-\varepsilon\Delta_h}$ again has a stochastic representation; now the representation is in terms of a jump Markov process with jumps from a point in $\mathcal{E}_h^d$ to one of its nearest neighbors on the grid (for an interesting approach to discretizing stochastic differential equations taking advantage of a similar observation see [10]).

Remark 3. The reader should notice that not only will we be unable to store the matrix $K_{\varepsilon,h}$ (which is exponentially large in d) or afford to compute $K_{\varepsilon,h}v$ for a general vector v, but we will not even be able to store the iterates v_t generated by the power method. Even the sparse matrix routines developed in numerical linear algebra to deal with large matrices are not reasonable for this problem.

In this discrete context, a direct application of the DMC approach (as in [41]) would represent the solution vector as a superposition of standard basis elements[5] and replace calculation of $K_{\varepsilon,h}v$ by a random approximation whose cost is (for this particular problem) free of any direct dependence on the size of $K_{\varepsilon,h}$ (though its cost can depend on structural properties of $K_{\varepsilon,h}$ which may be related to its size), whose expectation is exactly $K_{\varepsilon,h}v$, and whose variance is small. The approach in [8] is also an application of these same basic DMC steps to a discrete problem, though in that case the desired eigenvector has entries of *a priori* unkown sign, requiring that the solution vector be represented by a superposition of signed standard basis elements.

In this article we take a different approach to adapting DMC to discrete problems. Instead of reproducing in the discrete setting exactly the steps comprising DMC, consider a slightly modified scheme that omits direct randomization of an approximation to $e^{\varepsilon\Delta_h}$, and instead relies solely on a general random mapping Φ_t^m very similar to the map from V_t^m to Y_t^m but which takes a vector $V_t^m \in \mathbb{R}^{\mathcal{E}_h^d}$ and produces a new random vector Y_t^m with m, or nearly m, non-zero components and satisfying $\mathbf{E}\left[Y_t^m \,|\, V_t^m\right]$ as above. Starting from a non-negative initial vector $V_0^m \in \mathcal{E}_h^d$ with $\|V_0^m\|_1 = 1$ and with at most $\mathcal{O}(md)$ non-zero entries, V_{t+1}^m is generated from V_t^m as follows:

Step 1. Generate $Y_t^m = \Phi_t^m\left(V_t^m\right)$ with at most m non-zero entries.

Step 2. Set $V_{t+1}^m = \dfrac{K_{\varepsilon,h}Y_t^m}{\|K_{\varepsilon,h}Y_t^m\|_1}$.

For example, adapting the popular resampling choice mentioned above, we might choose the entries of Y_t^m independently with

$$\begin{aligned}\mathbf{P}\left[(Y_t^m)_j = \left\lfloor m(V_t^m)_j \right\rfloor / m\right] &= \left\lceil m(V_t^m)_j \right\rceil - m(V_t^m)_j,\\ \mathbf{P}\left[(Y_t^m)_j = \left\lceil m(V_t^m)_j \right\rceil / m\right] &= m(V_t^m)_j - \left\lfloor m(V_t^m)_j \right\rfloor.\end{aligned} \tag{7}$$

Note that the iterates V_t^m generated by this scheme have at most $\mathcal{O}(dm)$ non-zero entries. The details of the mappings Φ_t^m (which we call compression mappings) will be described later in Section 5 where, for example, we will find that the cost of applying the mapping Φ_t^m will typically be $\mathcal{O}(dm)$ when its argument has $\mathcal{O}(dm)$ non-zero entries. And while the cost of applying $K_{\varepsilon,h}$ to an arbitrary vector in $\mathbb{R}^{\mathcal{E}_h^d}$ is $|\mathcal{E}_h^d|$, the cost of applying $K_{\varepsilon,h}$ to a vector with m non-zero entries is only $\mathcal{O}(md)$. The total cost of the scheme is therefore $\mathcal{O}(md)$ in storage and operations per iteration. These cost requirements are dependent on the particular spatial discretization of $e^{\varepsilon\Delta}$; if we had chosen a discretization corresponding to a dense matrix $K_{\varepsilon,h}$ then the cost reduction would be much less extreme. Nonetheless, as we will see in Section 3, the scheme just described can be easily generalized and, as we will see in Sections 4 and 6, will often result in methods that are significantly faster than their deterministic counterparts.

Even restricting ourselves to the quantum Monte Carlo context, there is ample motivation to generalize the DMC scheme. Often one wishes to approximate not the smallest eigenvalue of $\mathcal{H}$ but instead the smallest eigenvalue corresponding to an antisymmetric (in exchange of particle positions) eigenfunction. DMC as described in this section, cannot be applied directly to computing this value, a difficulty commonly referred to as the Fermion sign problem. Several authors have attempted to address this issue with various modifications of DMC. In particular, Alavi and coworkers recently developed a version of DMC for a particular spatial disretization of the Hamiltonian (in the configuration interaction basis) that exactly preserves antisymmetry (unlike the finite difference discretization we just described). Run to convergence, their method provides the same approximation as the so called full CI method but can be applied with a much larger basis (e.g. in

[5]The delta functions represent the indices of v_t that we are keeping track of; δ_ξ is more commonly denoted e_ξ in numerical linear algebra — the standard basis vector with 1 in the ξth coordinate and zero elsewhere.

experiments by Alavi and coworkers reported in [51] up to 10^{108} total functions in the expansion of the solution). Though it differs substantially in its details from the scheme proposed in [8], the generalizations of DMC represented by the two step procedure in the last paragraph and by the developments in the next section can also be applied to a wider range of discretizations of $\mathcal{H}$. Finally we remark that, while we have considered DMC in the particular context of computing the ground state energy of a Hamiltonian, the method is used for a much wider variety of tasks with only minor modification to its basic structure. For example, particle filters (see e.g. [16]) are an application DMC to on-line data assimilation and substantive differences are mostly in the interpretation of the operator to which DMC is applied (and the fact that one is typically interested in the solution after finitely many iterations).

3 A general framework

Consider the general iterative procedure,

$$v_{t+1} = \mathcal{M}(v_t) \tag{8}$$

for $v_t \in \mathbb{C}^n$. Eigenproblems, linear systems, and matrix exponentiation can each be accomplished by versions of this iteration. In each of those settings the cost of evaluating $\mathcal{M}(v)$ is dominated by a matrix-vector multiplication. We assume that the cost (in terms of floating point operations and storage) of performing the required matrix-vector multiplication makes it impossible to carry out recursion (8) to the desired precision. As in the steps described at the end of the last section, we will consider the error resulting from replacement of (8) by

$$V_{t+1}^m = \mathcal{M}\left(\Phi_t^m(V_t^m)\right). \tag{9}$$

where the compression maps $\Phi_t^m : \mathbb{C}^n \to \mathbb{C}^n$ are independent, inexpensive to evaluate, and enforce sparsity in the V_t^m iterates (the number of non-zero entries in V_t^m will be $\mathcal{O}(m)$) so that $\mathcal{M}$ can be evaluated at much less expense. When $\mathcal{M}$ is a perturbation of identity *and* an $\mathcal{O}(n)$ scheme is appropriate (see Sections 3.2 and 4 below) we will also consider the scheme,

$$V_{t+1}^m = V_t^m + \mathcal{M}\left(\Phi_t^m(V_t^m)\right) - \Phi_t^m(V_t^m). \tag{10}$$

The compressions Φ_t^m will satisfy (or very nearly satisfy) the statistical consistency criterion

$$\mathbf{E}\left[\Phi_t^m(v)\right] = v \tag{11}$$

and will have to be carefully constructed to avoid instabilities and yield effective methods. For definiteness one can imagine that Φ_t^m is defined by a natural extension of (7)

$$\begin{aligned}
\mathbf{P}\left[\left(\Phi_t^m(v)\right)_j = \frac{v_j}{m|v_j|}\left\lfloor\frac{m|v_j|}{\|v\|_1}\right\rfloor\right] &= \left\lceil\frac{m|v_j|}{\|v\|_1}\right\rceil - \frac{m|v_j|}{\|v\|_1}, \\
\mathbf{P}\left[\left(\Phi_t^m(v)\right)_j = \frac{v_j}{m|v_j|}\left\lceil\frac{m|v_j|}{\|v\|_1}\right\rceil\right] &= \frac{m|v_j|}{\|v\|_1} - \left\lfloor\frac{m|v_j|}{\|v\|_1}\right\rfloor
\end{aligned} \tag{12}$$

to accept arguments $v \in \mathbb{C}^n$ (V_t^m is no longer a non-negative real number). This choice has several drawbacks, not least of which is its cost, and *we do not use it in our numerical experiments.* Alternative compression schemes, including the one used in our numerical simulations, are considered in detail in Section 5. There we will learn that one can expect that, for any pair $f, v \in \mathbb{C}^n$,

$$\sqrt{\mathbf{E}\left[|f^{\mathrm{H}}\Phi_t^m(v) - f^{\mathrm{H}}v|^2\right]} \leq \frac{2}{\sqrt{m}}\|f\|_\infty\|v\|_1 \tag{13}$$

(the superscript $^{\mathrm{H}}$ is used throughout this article to denote the conjugate transpose of a vector with complex entries). These errors are introduced at each iteration and need to be removed to obtain an accurate estimate. Depending on the setting, we may rely on averaging over long trajectories, averaging over parallel simulations (replicas), or dynamical self-averaging (see Sections 3.2 and 4), to remove the noise introduced by our randomization procedure. Because the specific choice of $\mathcal{M}$ and the form of averaging used to remove noise can differ substantially by setting, we will describe the schemes within the context of specific (and common) iterative procedures.

3.1. The eigenproblem revisited Consider, for example, a more general eigenproblem than the one we considered in Section 2. Given $K \in \mathbb{C}^{n\times n}$ the goal is to determine $\lambda_* \in \mathbb{C}$ and $v_* \in \mathbb{C}^n$ such that

$$Kv_* = \lambda_* v_* \tag{14}$$

and such that, for any other solution pair (μ, u), $|\mu| < |\lambda_*|$. In what follows in this section we will assume that this problem has a unique solution. The standard methods of approximate solution of (14) are variants of the power method, a simple version of which performs

$$v_{t+1} = \frac{Kv_t}{\|Kv_t\|_1}, \qquad \lambda_{t+1} = \frac{u^{\mathrm{H}} K v_t}{u^{\mathrm{H}} v_t}. \tag{15}$$

Under generic conditions, these converge to the correct (λ_*, v_*) starting from an appropriate initial vector v_0 (see e.g. [18]). The scheme in (15) requires $\mathcal{O}(n^2)$ work per iteration and at least $\mathcal{O}(n)$ storage. In this article, we are interested in situations in which these cost and storage requirements are unreasonable.

From $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ For the iteration in (15) the randomized scheme (9) (along with an approximation of λ_*) becomes

$$\begin{aligned} V^m_{t+1} &= \frac{K\Phi^m_t(V^m_t)}{\|K\Phi^m_t(V^m_t)\|_1}, & \Lambda^m_{t+1} &= \frac{u^{\mathrm{H}} K\Phi^m_t(V^m_t)}{u^{\mathrm{H}} V^m_t}, \\ \overline{V}_{t+1} &= (1-\varepsilon_t)\overline{V}^m_t + \varepsilon_t V^m_{t+1}, & \overline{\Lambda}^m_{t+1}, &= (1-\varepsilon_t)\overline{\Lambda}^m_t + \varepsilon_t \Lambda^m_{t+1}, \end{aligned} \tag{16}$$

where ε_t is a sequence of positive numbers with $\sum_{t=0}^{\infty} \varepsilon_t = \infty$ and $\sum_{t=0}^{\infty} \varepsilon_t^2 < \infty$ (if we choose $\varepsilon_t = \frac{1}{t+1}$ then V_t and Λ_t are trajectory averages). In (16), the compressions Φ^m_t are independent of one another. Using the rules defining Φ^m_t in Section 5, construction of $\Phi^m_t(V^m_t)$ at each step will require $\mathcal{O}(n)$ operations. Since multiplication of the vector $\Phi^m_t(V^m_t)$ by a dense matrix K requires $\mathcal{O}(nm)$ operations, this scheme has an $\mathcal{O}(n)$ storage and cost per iteration requirement. For exactly the same reasons, iterations based on (10) will also have cost and storage requirements of $\mathcal{O}(n)$. As we will see in Section 4, when the mapping $\mathcal{M}$ is of the form $v + \varepsilon b(v)$ for some small parameter ε, methods of the form (10) can be expected to have smaller error than iteration (9).

From $\mathcal{O}(n)$ to $\mathcal{O}(1)$ For many problems even $\mathcal{O}(n)$ cost and storage requirements are unacceptable. But now suppose that K is sparse with q non-zero entries per column. In this case not only would the cost of computing the matrix vector product $K\Phi^m_t(v)$ reduce to $\mathcal{O}(qm)$, but, according to (16), the vectors V^m_t would now have at most $\mathcal{O}(qm)$ non zero entries and the cost of computing $\Phi^m_t(V^m_t)$ would reduce to $\mathcal{O}(qm)$ operations as well. Thus when K is sparse the total number of floating point operations required by (16) reduces to $\mathcal{O}(qm)$ per iteration. This observation does not hold for methods of the form (10) which will typically result in dense iterates V^m_t and a cost of $\mathcal{O}(n)$ even when K is sparse.

As we have mentioned (and as was true in Section 2), in many settings even storing the full solution vector is impossible. Overcoming this impediment requires a departure from the usual perspective of numerical linear algebra. Instead of trying to approximate all entries of v_*, our goal becomes to compute

$$f_* = f^{\mathrm{H}} v_*$$

for some matrix f with n columns and many fewer ($\mathcal{O}(1)$ in n) rows. This change in perspective is reflected in the form of our compression rule error estimate in (13) and in the form of our convergence results in Section 4 that measure error in terms of dot products with test vectors. Indeed, were we to estimate a more standard quantity such as

$$\mathbf{E}\left[\|\Phi^m_t(v) - v\|_1\right]$$

we would find that the error decreased proportional to $(n-m)/n$ requiring that m increase with n to achieve fixed accuracy. The consequence in terms of the algorithm is simply the removal in (16) of the equation defining $\overline{V}_t^m$ and insertion of

$$F_{t+1}^m = f^{\mathrm{H}} V_{t+1}^m \quad \text{and} \quad \overline{F}_{t+1}^m = (1-\varepsilon_t)\overline{F}_t^m + \varepsilon_t F_{t+1}^m \tag{17}$$

which produces an estimate F_t^m of f_*.

Remark 4. While estimation of f_* may seem an unusual goal in the context of classical iterative schemes it is completely in line with the goals of any Markov chain Monte Carlo scheme which typically seek only to compute averages with respect to the invariant measure of a Markov chain and not to completely characterize that measure.

Schemes with $\mathcal{O}(1)$ storage and operations requirements per-iteration can easily be designed for any general matrix. Accomplishing this for a dense matrix requires an additional randomization in which columns of K (or of some factor of K) are randomly set to zero independently at each iteration, e.g. again in the context of power iteration, assuming that V_{t-1}^m has at most $\mathcal{O}(m)$ non-zero entries, one can use

$$V_{t+1}^m = \frac{\sum_{j=1}^n \Phi_t^{m_t^j, j} \left(K_j\right) \left(V_t^m\right)_j}{\left\| \sum_{j=1}^n \Phi_t^{m_t^j, j} \left(K_j\right) \left(V_t^m\right)_j \right\|_1} \tag{18}$$

in place of (16), where here K_j is used to denote the jth column of K and each $\Phi_t^{m_t^j, j}$ is an independent copy of $\Phi_t^{m_t^j}$. The number of entries retained in each column is controlled by m_t^j which, if the norms of the columns of K are known in advance (otherwise one can set $m_t^j = 1$), can be set randomly at each iteration by applying, e.g. (6) or some other resampling scheme with $W_t^{(j)}$ replaced by the vector with jth entry equal to $\|K_j\|_1 |(V_t^m)_j|$. This alternative approach also leaves the next iterate, V_{t+1}^m, with at most $\mathcal{O}(m)$ non-zero entries. Use of (18) in place of (16) will result in a scheme whose cost per iteration is independent of n if the compressions of the columns have cost independent of n. This may be possible without introducing significant error, for example, when the entries in the columns of K can take only a small number of values. However, use of (18) does not reduce the number of entries of K that must be accessed at any iteration and it will always increase error. In our numerical examples we do not apply the approach in (18) and we expect that in many applications effective constant cost per iteration schemes will only be possible if K is sparse.

3.2. Peturbations of identity We now consider the case that $\mathcal{M}$ is a perturbation of the identity, i.e., that

$$\mathcal{M}(v) - v = \varepsilon\, b(v) + o(\varepsilon) \tag{19}$$

where ε is a small positive parameter. This case is of particular importance because, when the goal is to solve a differential equation initial value problem

$$\frac{d}{dt} y = b(y), \qquad y(0) = y_0, \tag{20}$$

discrete-in-time approximations take the form (8) with $\mathcal{M}$ of the form in (19). As is the case in several of our numerical examples, the solution to (20) may represent, for example, a semi-discretization (a discretization in space) of a partial differential equation (PDE).

Several common tasks in numerical linear algebra, not necessarily directly related to ODE or PDE can also be addressed by considering (20). For example, suppose that we solve the ordinary differential equation (ODE) (20) with $b(y) = Ay - r$ for some $r \in \mathbb{C}^n$ and any $n \times n$ complex valued matrix A. The solution to (20) in this case is

$$y(t) = e^{tA} y_0 + A^{-1} \left(I - e^{tA}\right) r.$$

Setting $r = 0$ in the last display we find that any method to approximate ODE (20) for $t = 1$ can be used to approximate the product of a given vector and the exponential of the matrix A. On the other hand, if $r \neq 0$ and all eigenvalues of A have negative real part then, for very large t, the solution to (20) converges to $A^{-1}r$. In fact, in this case we obtain the continuous time variant of Jacobi iteration for the equation $Ax = r$. Like Jacobi iteration, it can be extended to somewhat more general matrices. Discretizing (20) in time with $b(y) = Ay - r$ and a small time step allows treatment of matrices with a wider range of eigenvalues than would be possible with standard Jacobi iteration.

Some important eigenproblems are also solved using an $\mathcal{M}$ satisfying (19). For example, this is the case when the goal is to compute the eigenvalue/vector pair corresponding to the eigenvalue of largest real part (rather than of largest magnitude) of a differential operator, e.g. the Schrödinger operator discussed in Section 2. While the power method applied directly to a matrix A converges to the eigenvector of A corresponding to the eigenvalue of largest magnitude, the angle between the vector $\exp(tA)y_0$ and the eigenvector corresponding to the eigenvalue of A with largest real part converges to 0 (assuming y_0 is not orthogonal to that eigenvector). If we discretize (20) in time with $b(v) = Av$ and renormalize the solution at each time step (to have unit norm) then the iteration will converge to the desired eigenvector (or a ε dependent approximation of that eigenvector).

As we will learn in the next section, designing effective fast randomized iteration schemes for these problems requires that the error in the stochastic representation $\mathcal{M} \circ \Phi_t^m$ of $\mathcal{M}$ decrease sufficiently rapidly with ε. In particular, in order for our schemes to accurately approximate solutions to (20) over intervals of $\mathcal{O}(1)$ units of time (i.e., over $\mathcal{O}(1/\varepsilon)$ iterations of the discrete scheme), we will need, and will verify in certain cases, that

$$\sqrt{\mathbf{E}\left[|f^{\mathrm{H}}\mathcal{M}\left(\Phi_t^m(v)\right) - f^{\mathrm{H}}\mathcal{M}(v)|^2\right]} \sim \sqrt{\varepsilon}.$$

Obtaining a bound of this type will require that we use a carefully constructed random compression Φ_t^m such as those described in Section 5. In fact, when a scheme with $\mathcal{O}(n)$ cost per iteration is acceptable, iteration (9) can be replaced by (10), i.e., by

$$V_{t+1}^m = V_t^m + \varepsilon\, b\left(\Phi_t^m(V_t^m)\right) + o(\varepsilon) \tag{21}$$

in which case we can expect errors over $\mathcal{O}\left(\varepsilon^{-1}\right)$ iterations that vanish with ε (rather than merely remaining stable). As we will see in more detail in the next section, the principle of dynamic self-averaging is essential to the convergence of either (9) or (10) when $\mathcal{M}$ is a perturbation of identity. The same principle is invoked in the contexts of, for example, multi-scale simulation (see e.g. [45] and [22] and the many references therein) and stochastic approximation (see e.g. [36] and the many references therein).

4 Convergence

Many randomized schemes in numerical linear algebra (see e.g. [15, 19, 20, 21, 23, 37, 39, 46, 47, 56]) rely at their core on approximation of a product such as AB, where, for example, A and B are $n \times n$ matrices, by a product of the form $A\Theta B$ where Θ is an $n \times n$ random matrix with $\mathbf{E}[\Theta] = I$ and so that $A\Theta B$ can be assembled at much less expense than AB. For example, one might choose Θ to be a diagonal matrix with only $m \ll n$ non-zero entries on the diagonal so that ΘB has only m non-zero rows and $A\Theta B$ can be assembled in $\mathcal{O}(n^2 m)$ operations instead of $\mathcal{O}(n^3)$ operations. Alternatively one might choose $\Theta = \xi\xi^{\mathrm{T}}$ where ξ is a $n \times m$ random matrix with independent entries, each having mean 0 and variance $1/m$. With this choice one can again construct $A\Theta B$ in $\mathcal{O}(n^2 m)$ operations. Typically, this randomization is carried out once in the course of the algorithm. The error made in such an approximation can be expected to be of size $\mathcal{O}(1/\sqrt{m})$ where the prefactor depends (very roughly) on the size of the matrices (and other structural properties) but does not depend directly on n.

In the schemes that we consider, we apply a similar randomization to speed matrix vector multiplication at each iteration of the algorithm (though our compression rules depend on the vector appearing in the product). As explored below, the consequence is that any stability property of the original, deterministic iteration responsible for its convergence, will be weakened by the randomization and that effect may introduce an additional n dependence in the cost of the algorithm to achieve a fixed accuracy. The compression rule must therefore be carefully constructed to minimize error. This is discussed in detail in Section 5. In this section, we consider the error resulting from (9) and (10) for an unspecified compression rule satisfying the generic error properties established (with caveats) in Section 5. Both because it provides a dramatic illustration of the need to construct accurate compression rules and because of its importance in practical applications, we pay particular attention to the case in which $\mathcal{M}$ is an ε-perturbation of the identity. Our results rely on classical techniques in the numerical analysis of deterministic and stochastic dynamical systems and, in particular, are typical of basic results concerning the convergence of stochastic approximation (see e.g. [36] for a general introduction and [40] for results in the context of machine learning) and interacting particle methods (see e.g. [17] for a general introduction and [49] for results in the context of QMC). They concern the mean squared size of the difference between the output, V_t^m, of the randomized scheme and the output, v_t, of its deterministic counterpart and are chosen to efficiently highlight important issues such as the role of stability properties of the deterministic iteration (8), the dependence of the error on the size of the solution vector, n, and the role of dynamic self-averaging. More sophisticated results (such as Central Limit Theorems and asymptotic and non-asymptotic exponential bounds on deviation probabilities) are possible following developments in, for example, [36] and [17, 49].

Our notion of error will be important. It will not be possible to prove, for example that

$$\mathbf{E}\left[\|V_t^m - v_t\|_1\right]$$

remains small without a strong dependence on n. It is not even the case that

$$\mathbf{E}\left[\|\Phi_t^m(v) - v\|_1\right]$$

is small when n is large and $\|v\|_1 = 1$. Take, for example, the case that $v_i = 1/n$. In this case any scheme that sets $n - m$ entries to zero will result in an error $\|\Phi_t^m(v) - v\|_1 \geq (n-m)/n$. On the other hand, we need to choose a measure of error sufficiently stringent so that our eventual error bounds imply that our methods accurately approximate observables of the form $f^{\mathrm{H}} v_t$. For example, analogues of all of the results below using the error metric

$$\sqrt{\mathbf{E}\left[\|V_t^m - v_t\|_2^2\right]}$$

could be established. However, error bounds of this form are not, by themselves, enough to imply bounds on the error in $f^{\mathrm{H}} v_t$ because they ignore correlations between the components of V_t^m.

Remark 5. It is perhaps more typical in numerical linear algebra to state error bounds in terms of the quantity one ultimately hopes to approximate and not in terms of the distance to another approximation of that quantity. For example, one might wonder why our results aren't stated in terms of the total work required to achieve (say with high probability) an error of a specified size in an approximation of the dominate eigenvalue of a matrix. Our choice to consider the difference between V_t^m and v_t is motivated by the fact that the essential characteristics contributing to errors due to randomization are most naturally described in terms of the map defining the deterministic iteration. More traditional characterizations of the accuracy of the schemes can be inferred from the bounds provided below and error bounds for the corresponding deterministic iterative schemes.

Motivated by our stated goal, as described in Section 3, of estimating quantities of the form $f^{\mathrm{H}} v_t$ we measure the size of the (random) errors produced by our scheme using the norm

$$|||X||| = \sup_{\|f\|_\infty \leq 1} \sqrt{\mathbf{E}\left[|f^{\mathrm{H}} X|^2\right]} \tag{22}$$

where X is a random variable with values in $\mathbb{C}^n$ (all random variables referenced are assumed to be functions on a single probability space which will be left unspecified). This norm is the $(\infty, 2)$-norm of the square root of the second moment matrix of X, i.e.

$$|||X||| = \|R\|_{\infty,2} = \sup_{\|f\|_\infty \leq 1} \|Rf\|_2$$

where

$$R^{\mathrm{H}} R = \mathbf{E}\left[XX^{\mathrm{H}}\right].$$

It is not difficult to see that the particular square root chosen does not effect the value of the norm. It will become apparent that our choice of the norm in (22) is a natural one for our convergence results in this and the next section.

The following alternate characterization of $|||\cdot|||$ will be useful later.

Lemma 1. *The norm in* (22) *may also be expressed as*

$$|||X||| = \sup_{\|G\|_{\infty,*} \leq 1} \sqrt{\mathbf{E}\left[\|GX\|_1^2\right]}, \tag{23}$$

where

$$\|G\|_{\infty,*} := \sum\nolimits_{i=1}^{n} \max_{j=1,\ldots,n} |G_{ij}| \tag{24}$$

is the dual norm of the ∞-norm of $G \in \mathbb{C}^{n\times n}$,

$$\|G\|_\infty = \max_{\|f\|_\infty \leq 1} \|Gf\|_\infty = \max_{i=1,\ldots,n} \sum\nolimits_{j=1}^{n} |G_{ij}|.$$

Proof. Indeed,

$$\mathbf{E}\left[\|GX\|_1^2\right] = \mathbf{E}\Big[\Big(\sum\nolimits_{i=1}^{n} \|g_i\|_\infty \, |\hat{g}_i^{\mathrm{H}} X|\Big)^2\Big]$$

where g_i is the ith column of G and $\hat{g}_i = g_i/\|g_i\|_\infty$. Using the condition

$$\sum\nolimits_{i=1}^{n} \max_{j=1,\ldots,n} |G_{ij}| \leq 1 \tag{25}$$

and Jensen's inequality we find that

$$\mathbf{E}\left[\|GX\|_1^2\right] \leq \sum\nolimits_{i=1}^{n} \|g_i\|_\infty \, \mathbf{E}\left[|\hat{g}_i^{\mathrm{H}} X|^2\right] \leq |||X|||^2.$$

The other inequality follows by noting that for any $f \in \mathbb{C}^n$ with $\|f\|_\infty \leq 1$, the matrix G with first row equal to f^{H} and all other rows zero satisfies the constraint (25). That (24) is the dual norm of the ∞-norm follows from straightforward verification or see [25, Proposition 6.3]. □

Note that if the variable X is not random then one can choose $f_i = X_i/|X_i|$ in (22) and find that $|||X||| = \|X\|_1$. When X is random we have the upper bound $|||X|||^2 \leq \mathbf{E}\left[\|X\|_1^2\right]$. If, on the other hand, X is random but has mean zero and independent components then $|||X|||^2 = \mathbf{E}\left[\|X\|_2^2\right]$. Concerning the relationship between these two norms more generally, we rely on the following lemma.

Lemma 2. *Let A be any $n \times n$ Hermissian matrix with entries in $\mathbb{C}$. Then*

$$\sup_{\|f\|_\infty \leq 1} f^{\mathrm{H}} A f \geq \operatorname{trace} A.$$

Proof. The result holds in $n = 1$ dimensions. Suppose that the result holds in $n-1$ dimensions. We will show that it must also therefore hold in n dimensions and conclude, by induction, that the result holds in any dimension.

Let $\tilde{A}$ be the $(n-1)\times(n-1)$ principle sub-matrix of an $n\times n$ matrix A. For any vector $f \in \mathbb{C}^n$ we can write

$$\begin{aligned} f^{\mathrm{H}} A f &= \sum\nolimits_{i=1}^{n} |f_i|^2 A_{ii} + 2\Re\left[\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{i-1} \bar{f}_i A_{ij} f_j\right] \\ &= \tilde{f}^{\mathrm{H}} \tilde{A} \tilde{f} + |f_n|^2 A_{nn} + 2\Re\left[\bar{f}_n \sum\nolimits_{j=1}^{n-1} A_{nj} \tilde{f}_j\right] \end{aligned}$$

where $\tilde{f} \in \mathbb{C}^{n-1}$ has entries equal to the first $n-1$ entries of f.

By the induction hypothesis, we can choose the first $n-1$ entries of f (i.e., $\tilde{f}$) so that the right hand side of the last display is not less than

$$\sum\nolimits_{i=1}^{n-1} A_{ii} + |f_n|^2 A_{nn} + 2\Re\left[\bar{f}_n \sum\nolimits_{j=1}^{n-1} A_{nj} \tilde{f}_j\right].$$

If, for this choice of $\tilde{f}$, $\sum_{j=1}^{n-1} A_{nj}\tilde{f}_j$ is non-zero then choose f_n as

$$f_n = \frac{\sum_{j=1}^{n-1} A_{nj}\tilde{f}_j}{\left|\sum_{j=1}^{n-1} A_{nj}\tilde{f}_j\right|}.$$

Otherwise set $f_n = 1$. With the resulting choice of f_n,

$$|f_n|^2 A_{nn} + 2\Re\left[\bar{f}_n \sum\nolimits_{j=1}^{n-1} A_{nj}\tilde{f}_j\right] \geq A_{nn}.$$

We have therefore shown that

$$\sup_{\|f\|_\infty \leq 1} f^{\mathrm{H}} A f \geq \sum\nolimits_{i=1}^{n} A_{ii}. \qquad \square$$

Lemma 2, applied to the second moment matrix of X, implies that $|||X|||^2 \geq \mathbf{E}\left[\|X\|_2^2\right]$. Summarizing these relationships we have

$$\mathbf{E}\left[\|X\|_2^2\right] \leq |||X|||^2 \leq \mathbf{E}\left[\|X\|_1^2\right]. \tag{26}$$

Consistent with results in the next section we will assume that the typical error from our compression rule is

$$|||\Phi_t^m(v) - v||| \leq \frac{\gamma}{\sqrt{m}} \|v\|_1 \tag{27}$$

for $v \in \mathbb{C}^n$, where γ is a constant that is independent of m and n. We will also assume that

$$\mathbf{E}\left[\|\Phi_t^m(v)\|_1^2\right] \leq C\|v\|_1^2 \tag{28}$$

for some constant C independent of m and n. For all of the compression methods discussed in the next section, the bias $\|\mathbf{E}\left[\Phi_t^m(v)\right] - v\|_1$ is negligible compared to the statistical error in (27). To slightly simplify expressions we will omit contributions to our bounds from compression bias (we do consider the bias of the overall iterative procedure). In other words we will assume that (11) is satisfied exactly.

As a result of the appearance of $\|v\|_1$ in (27), in our eventual error bounds it will be impossible to avoid dependence on $\|V_t^m\|_1$. The growth of these quantities is controllable by increasing m, but the value of m required will often depend on the n. The next theorem concerns the size of $\mathbf{E}\left[\|V_t^m\|_1^2\right]$. After the statement and proof of the theorem we discuss how the various quantities appearing there can be be expected to depend on n. In this theorem and in the rest of our results it will be convenient to recognize that, in many applications, the iterates V_t^m and $Y_t^m = \Phi_t^m(V_t^m)$ are confined within some subset of $\mathbb{C}^n$. For example, the iterates may all have a fixed norm or may have all non-negative entries. We use the symbol $\mathcal{X}$ to identify this subset (which differs depending on the problem).

Theorem 6. *Suppose that for some constants γ and R,*

$$\mathcal{U}(\mathcal{M}(v)) \leq \alpha\,\mathcal{U}(v) + R$$

for all $v \in \mathcal{X}$, and that, for some constant β,

$$\|v\|_1^2 \leq \beta\mathcal{U}(v)$$

for all $v \in \mathcal{X}$. Assume further that there is a constant C and some matrix $G \in \mathbb{C}^{n\times n}$ satisfying (25), so that, for all $z \in \mathbb{C}^n$,

$$\sup_{v\in\mathbb{C}^n} z^{\mathrm{H}}\left(D^2\mathcal{U}(v)\right)z \leq C\|Gz\|_1^2$$

where $D^2\mathcal{U}$ is the matrix of second derivatives of $\mathcal{U}$. Then

$$\mathbf{E}\left[\|V_t^m\|_1^2\right] \leq \beta R\left[\frac{1-\alpha^t\left(1+\frac{\beta\gamma^2C}{2m}\right)^t}{1-\alpha\left(1+\frac{\beta\gamma^2C}{2m}\right)}\right] + \beta\alpha^t\left(1+\frac{\beta\gamma^2C}{2m}\right)^t\mathcal{U}(V_0^m).$$

Proof. Let $Y_t^m = \Phi_t^m(V_t^m)$ and notice that

$$\begin{aligned}\mathcal{U}(V_t^m) &= \mathcal{U}(\mathcal{M}(Y_{t-1}^m))\\ &\leq R + \alpha\,\mathcal{U}(V_{t-1}^m) + \alpha\left(\mathcal{U}(Y_{t-1}^m) - \mathcal{U}(V_{t-1}^m)\right).\end{aligned}$$

Using the fact that $\mathcal{U}$ is twice differentiable with bounded second derivative this last expression is bounded above by

$$\mathcal{U}(V_t^m) \leq R + \alpha\,\mathcal{U}(V_{t-1}^m) + \alpha\nabla\mathcal{U}(V_{t-1}^m)\left(Y_{t-1}^m - V_{t-1}^m\right) + \frac{\alpha C}{2}\|G\left(Y_{t-1}^m - V_{t-1}^m\right)\|_1^2.$$

An application of Lemma 1 along with the fact that $\mathbf{E}\left[Y_{t-1}^m \,|\, V_{t-1}^m\right] = V_{t-1}^m$, reveals the bound

$$\mathbf{E}\left[\|G\left(Y_{t-1}^m - V_{t-1}^m\right)\|_1^2\right] \leq \|\!|Y_{t-1}^m - V_{t-1}^m|\!\|^2.$$

As a consequence, noting (27), we arrive at the upper bound

$$\begin{aligned}\mathbf{E}\left[\mathcal{U}(V_t^m)\right] &\leq R + \alpha\,\mathcal{U}(V_{t-1}^m) + \frac{\alpha\gamma^2C}{2m}\mathbf{E}\left[\|V_{t-1}^m\|_1^2\right]\\ &\leq R + \alpha\left(1+\frac{\beta\gamma^2C}{2m}\right)\mathbf{E}\left[\mathcal{U}(V_{t-1}^m)\right]\end{aligned}$$

from which we can conclude that

$$\mathbf{E}\left[\|V_t^m\|_1^2\right] \leq \beta\mathbf{E}\left[\mathcal{U}(V_t^m)\right] \leq \beta R\left[\frac{1-\alpha^t\left(1+\frac{\beta\gamma^2C}{2m}\right)^t}{1-\alpha\left(1+\frac{\beta\gamma^2C}{2m}\right)}\right] + \beta\alpha^t\left(1+\frac{\beta\gamma^2C}{2m}\right)^t\mathcal{U}(V_0^m). \qquad \square$$

First, the reader should notice that setting $\gamma = 0$ in Theorem 6 shows that the deterministic iteration (8) is stable whenever $\alpha < 1$. However, even for an $\mathcal{M}$ corresponding to a stable iteration, the randomized iteration (9) may not be stable (and will, in general, be less stable). If the goal is to estimate, e.g. a fixed point of $\mathcal{M}$, the user will first have to choose m large enough that the randomized iteration is stable.

Though it is not explicit in the statement of Theorem 6, in general the requirements for stability will depend on n. Consider, for example, the case of a linear iteration, $\mathcal{M}(v) = Kv$. This iteration is stable if the largest eigenvalue (in magnitude) is less than 1. If we choose $\mathcal{U}(v) = \|v\|_2^2$ then we can take α to be the largest eigenvalue of $K^{\mathrm{H}}K$ and $R = 0$ in the statement of Theorem 6. The bound $\|\cdot\|_1 \leq \sqrt{n}\|\cdot\|_2$ (and the fact that it is sharp) suggests that we will have to take $\beta = n$ in Theorem 6 (note that we can take $C = 1$ in the eventual bound). This scaling suggests that, to guarantee stability we need to choose $m \sim n$.

Fortunately this prediction is often (but not always) pessimistic. For example, if K is a matrix with non-negative real entries and V_0^m has non-negative real entries then the iterates V_t^m will have real, non-negative entries (i.e., $v \in \mathcal{X}$ implies $v_i \geq 0$). We can therefore use $\mathcal{U}(v) = (\mathbb{1}^{\mathrm{T}}v)^2 = \|v\|_1^2$

for $v \in \mathcal{X}$ and find that we can take $\alpha = \|K\|_1^2$, $R = 0$, and $\beta = 1$, in the statement of Theorem 6. With this choice of $\mathcal{U}$ we can again choose $C = 1$ so that n does not appear directly in the stability bound. We anticipate that most applications will fall somewhere between these two extremes; maintaining stability will require increasing m as n is increased but not in proportion to the increase in n.

Having characterized the stability our schemes we now move on to bounding their error. We have crafted the theorem below to address both situations in which one is interested in the error after a finite number of iterations and situations that require error bounds independent of the number of iterations. In general, achieving error bounds independent of the number of iterations requires that $\mathcal{M}$ satisfy stronger stability properties than those implied by the conditions in Theorem 6. While the requirements in Theorem 6 could be modified to imply the appropriate properties for most applications, we opt instead for a more direct approach and modify our stability assumptions on $\mathcal{M}$ to (29) and (30) below. In this theorem and below we will make use the notation $\mathcal{M}_s^t$ to denote $\mathcal{M}$ composed with itself $t - s$ times. In our proof of the bound in Theorem 7 below we will divide the error into two terms, one of which is a sum of individual terms with vanishing conditional expectations. Those individual summands are Martingale differences, and the size of their sum can be expected to grow less than linearly with the number of iterations. This general observation is called dynamic self-averaging and results in an improved error bound. The improvement is essential in the context of perturbations of the identity and we will mention it again below when we focus on that case.

Theorem 7. *Suppose that the iterates V_t^m of (9) remain in $\mathcal{X} \subset \mathbb{C}^n$. Fix a positive integer T. Assume that, for some $\alpha \geq 0$ and some constants L_1 and L_2 and for every pair of integers $s \leq r \leq T$, for every vector $f \in \mathbb{C}^n$ with $\|f\|_\infty \leq 1$ there is a matrix $G \in \mathbb{C}^{n \times n}$ satisfying (25) and*

$$\sup_{v,\tilde{v} \in \mathcal{X}} \frac{|f^{\mathrm{H}} \mathcal{M}_s^r(v) - f^{\mathrm{H}} \mathcal{M}_s^r(\tilde{v})|}{\|Gv - G\tilde{v}\|_1} \leq L_1 \alpha^{r-s} \tag{29}$$

and for every $f \in \mathbb{C}^n$ with $\|f\|_\infty \leq 1$ and some matrix $G \in \mathbb{C}^{n \times n}$ satisfying (25), for all $v \in \mathcal{X}$ there is a matrix A (a measurable function of v) so that

$$\frac{|f^{\mathrm{H}} \mathcal{M}_s^r(v) - f^{\mathrm{H}} \mathcal{M}_s^r(\tilde{v}) - f^{\mathrm{H}} A(v - \tilde{v})|}{\|Gv - G\tilde{v}\|_1^2} \leq L_2 \alpha^{r-s} \tag{30}$$

for all $\tilde{v} \in \mathcal{X}$. Then the error at step $t \leq T$ satisfies the bound

$$|||V_t^m - v_t||| \leq \alpha \left[\gamma \frac{1}{\sqrt{m}} (L_1 + L_2) \left(\frac{1 - \alpha^{2t}}{1 - \alpha^2} \right)^{1/2} M_T + \gamma^2 \frac{1}{m} L_2 \frac{1 - \alpha^t}{1 - \alpha} M_T^2 \right]$$

where $M_T^2 = \sup_{r<T} \mathbf{E}\left[\|V_r^m\|_1^2\right]$.

Proof. We begin with a standard expansion of the scheme's error.

$$\begin{aligned} |||V_t^m - v_t||| &= \left|\left|\left| V_t^m - \mathcal{M}_0^t(v_0) \right|\right|\right| \\ &= \left|\left|\left| \sum_{r=0}^{t-1} \mathcal{M}_{r+1}^t(V_{r+1}^m) - \mathcal{M}_r^t(V_r^m) \right|\right|\right|. \end{aligned}$$

Now notice that if we define $Y_r^m = \Phi_r^m(V_r^m)$ then $V_{r+1}^m = \mathcal{M}_r(Y_r^m)$ and the last equation becomes

$$|||V_t^m - v_t||| = \left|\left|\left| \sum_{r=0}^{t-1} \mathcal{M}_r^t(Y_r) - \mathcal{M}_r^t(V_r^m) \right|\right|\right|.$$

The right hand side of the last equation is bounded above by

$$\left|\left|\left| \sum_{r=0}^{t-1} \mathcal{M}_r^t(Y_r) - \mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m] \right|\right|\right| + \sum_{r=0}^{t-1} |||\mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m] - \mathcal{M}_r^t(V_r^m)|||.$$

Considering the first term in the last display note that, for any fixed $f \in \mathbb{C}^n$,

$$\mathbf{E}\Big[\big|f^{\mathrm{H}}\sum_{r=0}^{t-1}(\mathcal{M}_r^t(Y_r) - \mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m])\big|^2\Big] = \sum_{r=0}^{t-1}\mathbf{E}\Big[\big|f^{\mathrm{H}}(\mathcal{M}_r^t(Y_r) - \mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m])\big|^2\Big] + 2\sum_{s=0}^{t-1}\sum_{r=s+1}^{t-1}\Re\Big\{\mathbf{E}\Big[\big(f^{\mathrm{H}}(\mathcal{M}_r^t(Y_r) - \mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m])\big) \times \overline{\big(f^{\mathrm{H}}(\mathcal{M}_s^t(Y_s) - \mathbf{E}[\mathcal{M}_s^t(Y_s) \mid V_s^m])\big)}\Big]\Big\}.$$

Letting $\mathcal{F}_r$ denote the σ-algebra generated by $\{V_s^m\}_{s=0}^r$ and $\{Y_r^m\}_{s=0}^{r-1}$, for $s < r$ we can write

$$\mathbf{E}\Big[\big(f^{\mathrm{H}}(\mathcal{M}_r^t(Y_r) - \mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m])\big) \times \overline{\big(f^{\mathrm{H}}(\mathcal{M}_s^t(Y_s) - \mathbf{E}[\mathcal{M}_s^t(Y_s) \mid V_s^m])\big)}\Big] = \mathbf{E}\Big[\mathbf{E}\big[f^{\mathrm{H}}(\mathcal{M}_r^t(Y_r) - \mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m]) \mid \mathcal{F}_r\big] \times \overline{\big(f^{\mathrm{H}}(\mathcal{M}_s^t(Y_s) - \mathbf{E}[\mathcal{M}_s^t(Y_s) \mid V_s^m])\big)}\Big].$$

Because, conditioned on V_r^m, Y_r^m is independent of $\mathcal{F}_r$, the expression above vanishes exactly.

Supremizing over the choice of f, we have shown that

$$|||V_t^m - v_t||| \leq \Big(\sum_{r=0}^{t-1}|||\mathcal{M}_r^t(Y_r) - \mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m]|||^2\Big)^{1/2} + \sum_{r=0}^{t-1}|||\mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m] - \mathcal{M}_r^t(V_r^m)|||.$$

Expanding the term inside of the square root we find that

$$\begin{aligned}|||V_t^m - v_t||| &\leq \Big(\sum_{r=0}^{t-1}\big(|||\mathcal{M}_r^t(Y_r) - \mathcal{M}_r^t(V_r^m)||| + |||\mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m] - \mathcal{M}_r^t(V_r^m)|||\big)^2\Big)^{1/2} \\ &\qquad + \sum_{r=0}^{t-1}|||\mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m] - \mathcal{M}_r^t(V_r^m)||| \\ &\leq \Big(\sum_{r=0}^{t-1}|||\mathcal{M}_r^t(Y_r) - \mathcal{M}_r^t(V_r^m)|||^2\Big)^{1/2} + \Big(\sum_{r=0}^{t-1}|||\mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m] - \mathcal{M}_r^t(V_r^m)|||^2\Big)^{1/2} \\ &\qquad + \sum_{r=0}^{t-1}|||\mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m] - \mathcal{M}_r^t(V_r^m)|||\end{aligned}$$

where, in the second inequality, we have used the triangle inequality for the ℓ^2 norm in $\mathbb{R}^t$. If $A_r \in \mathbb{C}^{n\times n}$ is any matrix value random variable, measurable with respect to the σ-algebra generated by V_r^m, then

$$\mathbf{E}[\mathcal{M}_r^t(Y_r) \mid V_r^m] - \mathcal{M}_r^t(V_r^m) = \mathbf{E}\big[\big(\mathcal{M}_r^t - A_r\big)(Y_r) \;\big|\; V_r^m\big] - \big(\mathcal{M}_r^t - A_r\big)(V_r^m).$$

As a consequence, applying our assumptions (29) and (30), we obtain the upper bound

$$|||V_t^m - v_t||| \leq (L_1 + L_2)\Big(\sum_{r=0}^{t-1}\alpha^{2(t-r)}|||\Phi_r^m(V_r^m) - V_r^m|||^2\Big)^{1/2} + L_2\sum_{r=0}^{t-1}\alpha^{t-r}|||\Phi_r^m(V_r^m) - V_r^m|||^2.$$

Bounding the error from the random compressions we arrive at the error bound

$$|||V_t^m - v_t||| \leq \frac{\gamma(L_1 + L_2)}{\sqrt{m}}\Big(\sum_{r=0}^{t-1}\alpha^{2(t-r)}\mathbf{E}\left[\|V_r^m\|_1^2\right]\Big)^{1/2} + \frac{\gamma^2 L_2}{m}\sum_{r=0}^{t-1}\alpha^{t-r}\mathbf{E}\left[\|V_r^m\|_1^2\right]. \qquad \square$$

Condition (29) is easily verified for linear maps $\mathcal{M} = K$ where the magnitude of the dominant eigenvalue of K is less than 1. The condition is more difficult to verify for the power iteration map

$$\mathcal{M}(v) = \frac{Kv}{\|Kv\|_1}.$$

A condition close to (29) holds for all $v, \tilde{v}$ with $\|v\|_1 = \|\tilde{v}\|_1 = 1$ but the parameter α in general depends on the proximity of v and $\tilde{v}$ to the space spanned by all of the non-dominant eigenvectors (see e.g. [54]). For large enough m we can ensure that all iterates V_t^m remain at least some fixed distance from the space spanned by the non-dominant eigenvectors but this will often require that m grow with n. Moreover, almost sure bounds separating the iterates from the non-dominant eigenspace are more than is required to prove convergence of the randomized schemes.

Again we can identify cases in which condition (29) holds for the power iteration map without direct dependence on n in the bound. When the matrix K is real and non-negative we can assume that the iterates V_t^m are also real and non-negative, i.e., the iterates remain in

$$\mathcal{X} = \{v \in \mathbb{C}^n : v_i \geq 0,\, \|v\|_1 = \mathbb{1}^{\mathrm{T}}v = 1\}.$$

On the other hand, if K is also irreducible, then the non-dominant eigenspace of K consists of vectors with both positive and negative entries and our iterates, by virtue of being non-negative, are bounded away from this space. For more general problems one can expect the speedup over the standard deterministic power method to be roughly between a factor of n and no speedup at all (it is clear that the randomized scheme can be worse than its deterministic counterpart when that method is a reasonable alternative). A complete description of the speedup in the particular case that $\mathcal{M}$ is the power iteration map would be very interesting but is not pursued here.

Bias. Even for a very general iteration the effect of randomization is evident when one considers the size of the expected error (rather than the expected size of the error). When $\mathcal{M}(v) = Kv$ and V_t^m is generated by (9), one can easily check that $z_t = \mathbf{E}[V_t^m]$ satisfies the iteration $z_{t+1} = Kz_t$, i.e., $z_t = v_t$. Even when the mapping $\mathcal{M}$ is non-linear, the expected error, $\mathbf{E}[V_t^m] - v_t$, is often much smaller than the error, $V_t^m - v_t$, itself. The following is just one simple result in this direction and demonstrates that one can often expect the bias to be $\mathcal{O}(m^{-1})$ (which should be contrasted to an expected error of $\mathcal{O}(m^{-1/2})$). The proof is very similar to the proof of Theorem 7 and is omitted.

Theorem 8. *Under the same assumptions as in Theorem 7, the bias at step $t \leq T$ satisfies the bound*

$$\|\mathbf{E}[V_t^m] - v_t\|_1 \leq \frac{\gamma^2 L_2}{m} \frac{\alpha(1-\alpha^t)}{1-\alpha} \sup_{r<T} \mathbf{E}\left[\|V_r^m\|_1^2\right].$$

Perturbations of identity. When the goal is to solve ordinary or partial differential equations, stronger assumptions on the structure of $\mathcal{M}$ are appropriate. We now consider the case in which $\mathcal{M}$ is a perturbation of the identity. More precisely we will assume that

$$\mathcal{M}(v) = v + \varepsilon\, b(v). \tag{31}$$

Though we will not write it explicitly, further dependence of b on ε is allowed as long as the assumptions on b below hold uniformly in ε. In the differential equations setting ε can be thought of as a time discretization parameter as in Section 2.

When $\mathcal{M}$ is a perturbation of identity, it is reasonable to strengthen our assumptions on the error made at each compression step. The improvement stems from the fact that the mapping $\mathcal{M}$ nearly preserves the sparsity of its argument. As we will explain in detail in the next section, if $v \in \mathcal{S}_m$ where

$$\mathcal{S}_m = \{z \in \mathbb{C}^n : \#\{j : z_j \neq 0\} \leq m\}$$

and $w \in \mathbb{C}^n$, then it is reasonable to assume that, for example,

$$|||\Phi_t^m(v+w) - v - w||| \leq \frac{C}{\sqrt{m}} \|w\|_1^{\frac{1}{2}} \|v+w\|_1^{\frac{1}{2}} \tag{32}$$

for some constant C independent of m and n. The following Lemma illustrates how such a bound on the compression rule can translate into small compression errors when $\mathcal{M}$ is a perturbation of the identity.

Lemma 3. *Suppose that (32) and (28) hold. If $\mathcal{M}(v) = v + \varepsilon b(v)$ and, for some constant L, $\|b(v)\|_1 \leq L(1 + \|v\|_1)$ for all $v \in \mathcal{X}$, then, for V_t^m generated by (9), and for some constant $\tilde{\gamma}$,*

$$|||\Phi_t^m(V_t^m) - V_t^m|||^2 \leq \tilde{\gamma}^2 \frac{\varepsilon}{m} \sqrt{\mathbf{E}\left[\|V_t^m\|_1^2\right]} \sqrt{1 + \mathbf{E}\left[\|V_{t-1}^m\|_1^2\right]} \tag{33}$$

Proof. If $Y_t^m = \Phi_t^m(V_t^m)$ then

$$\begin{aligned}\mathbf{E}\left[|f^{\mathrm{H}}\Phi_t^m(V_t^m) - f^{\mathrm{H}}V_t^m|^2 \,|\, Y_{t-1}^m\right] &= \mathbf{E}\left[|f^{\mathrm{H}}\Phi_t^m\left(Y_{t-1}^m + \varepsilon b(Y_{t-1}^m)\right) - f^{\mathrm{H}}\left(Y_{t-1}^m + \varepsilon b(Y_{t-1}^m)\right)|^2 \,|\, Y_{t-1}^m\right] \\ &\leq C\frac{\varepsilon}{m}\|b(Y_{t-1}^m)\|_1 \|V_t^m\|_1\end{aligned}$$

for some constant C. Our assumed bound on the growth of b along with (28) implies that

$$\mathbf{E}\left[\|b(Y_{t-1}^m)\|_1^2\right] \leq C'\left(1+\mathbf{E}\left[\|V_{t-1}^m\|_1^2\right]\right)$$

for some constant C'. From these bounds it follows that for some constant $\tilde{\gamma}$,

$$\|\|\Phi_t^m(V_t^m)-V_t^m\|\|^2 \leq \tilde{\gamma}^2\frac{\varepsilon}{m}\sqrt{\mathbf{E}\left[\|V_t^m\|_1^2\right]}\sqrt{1+\mathbf{E}\left[\|V_{t-1}^m\|_1^2\right]}. \qquad \square$$

We now provide versions of Theorems 6 and 7 appropriate when $\mathcal{M}$ is a perturbation of identity. The proofs of both of these theorems are very similar to the proofs of Theorems 6 and 7 and are, at least in part, omitted. First we address stability in the perturbation of identity case.

Theorem 9. *Suppose that the iterates V_t^m of (9) remain in $\mathcal{X}\subset\mathbb{C}^n$ and that (33) holds. Suppose further that $\mathcal{U}$ satisfies the conditions in the statement of Theorem 6 with the exception of the following: Now*

$$U(\mathcal{M}(v)) \leq e^{\varepsilon\alpha}\mathcal{U}(v)+\varepsilon R.$$

Then

$$\sup_{t<T/\varepsilon}\mathbf{E}\left[\|V_t^m\|_1^2\right] \leq \beta R\left[\varepsilon+\frac{\exp\left[T\left(\alpha+\frac{\beta\gamma^2C}{2m}\right)\right]-1}{\alpha+\frac{\beta\gamma^2C}{2m}}\right]+\beta\exp\left[T\left(\alpha+\frac{\beta\gamma^2C}{2m}\right)\right]\mathcal{U}(V_0^m).$$

where the various constants appearing in this expression were defined in and before the statement of Theorem 6.

What is important about the statement of Theorem 9 is that the bound remains stable as ε decreases despite the fact that set being supremized over is increasing. Under the assumptions in the theorem (which are only reasonable when $\mathcal{M}$ is a perturbation of the identity) one can expect that the iterates can be bounded over $\mathcal{O}\left(\varepsilon^{-1}\right)$ iterations uniformly in ε.

The following theorem interprets the result of Theorem 7 when $\mathcal{M}$ is a perturbation of identity. One might expect that, over $\mathcal{O}(\varepsilon^{-1})$ iterations, $\mathcal{O}\left(\sqrt{\varepsilon}\right)$ errors made during the compression step would accumulate and lead to an error of $\mathcal{O}(\varepsilon^{-1/2})$. Indeed, this is exactly what would happen if the errors made in the compression step were systematic (i.e., if the compression bias was $\mathcal{O}\left(\sqrt{\varepsilon}\right)$). Fortunately, when the compression rule satisfies the consistency criterion (11) the errors self average and their effect on the overall error of the scheme is reduced. As mentioned above, this phenomenon played a role in the structure of the result in Theorem 7 and its proof, but its role is more crucial in Theorem 10 which provides uniform in ε bounds on the error of (9) over $\mathcal{O}(\varepsilon^{-1})$ iterations. Without the reduction in the growth of the error with t provided by self-averaging it would not be possible to achieve an error bound over $\mathcal{O}\left(\varepsilon^{-1}\right)$ iterations that is stable as ε decreases.

Theorem 10. *Suppose that the iterates V_t^m of (9) remain in $\mathcal{X}\subset\mathbb{C}^n$ and that (33) holds. Fix a real number $T>0$. Assume that*

$$\mathcal{M}(v) = v+\varepsilon b(v).$$

Assume further that, for some real number β and some constants L_1 and L_2 and for every pair of integers $s\leq r\leq T/\varepsilon$, for every vector $f\in\mathbb{C}^n$ with $\|f\|_\infty\leq 1$ there is a matrix $G\in\mathbb{C}^{n\times n}$ satisfying (25) and

$$\sup_{v,\tilde{v}\in\mathcal{X}}\frac{|f^{\mathrm{H}}\mathcal{M}_s^r(v)-f^{\mathrm{H}}\mathcal{M}_s^r(\tilde{v})|}{\|Gv-G\tilde{v}\|_1} \leq L_1e^{-\varepsilon\beta(r-s)} \tag{34}$$

and for every $f\in\mathbb{C}^n$ with $\|f\|_\infty\leq 1$ and some matrix $G\in\mathbb{C}^{n\times n}$ satisfying (25), for all $v\in\mathcal{X}$ there is a matrix A (a measurable function of v) so that

$$\frac{|f^{\mathrm{H}}\mathcal{M}_s^r(v)-f^{\mathrm{H}}\mathcal{M}_s^r(\tilde{v})-f^{\mathrm{H}}A(v-\tilde{v})|}{\|Gv-G\tilde{v}\|_1^2} \leq L_2e^{-\varepsilon\beta(r-s)} \tag{35}$$

for all $\tilde{v} \in \mathcal{X}$. Then the error at step $t \leq T/\varepsilon$ satisfies the bound

$$|||V_t^m - v_t||| \leq \frac{\tilde{\gamma}(L_1+L_2)}{\sqrt{m}} \left(e^{-2\beta T} \mathbf{E}\left[\|V_0^m\|_1^2\right] + \frac{1-e^{-2\beta T}}{2\beta} M_T \sqrt{1+M_T^2} \right)^{1/2} + \frac{\tilde{\gamma}^2 L_2}{m} \left(e^{-2\beta T} \mathbf{E}\left[\|V_0^m\|_1^2\right] + \frac{1-e^{-\beta T}}{\beta} M_T \sqrt{1+M_T^2} \right).$$

where $M_T^2 = \sup_{r<T/\varepsilon} \mathbf{E}\left[\|V_r^m\|_1^2\right]$.

Proof. By exactly the same arguments used in the proof of Theorem 7 we arrive at the bound

$$|||V_t^m - v_t||| \leq (L_1+L_2) \left(\sum\nolimits_{r=0}^{t-1} e^{-2\beta(t-r)\varepsilon} |||\Phi_r^m(V_r^m) - V_r^m|||^2 \right)^{1/2} + L_2 \sum\nolimits_{r=0}^{t-1} e^{-\beta(t-r)\varepsilon} |||\Phi_r^m(V_r^m) - V_r^m|||^2.$$

Bounding the error from the random compressions we arrive at the error bound

$$|||V_t^m - v_t||| \leq \frac{\tilde{\gamma}(L_1+L_2)}{\sqrt{m}} \left(e^{-2\beta t\varepsilon} \mathbf{E}\left[\|V_0^m\|_1^2\right] + \varepsilon \sum\nolimits_{r=1}^{t-1} e^{-2\beta(t-r)\varepsilon} \sqrt{\mathbf{E}\left[\|V_t^m\|_1^2\right]} \sqrt{1+\mathbf{E}\left[\|V_{t-1}^m\|_1^2\right]} \right)^{\frac{1}{2}} + \frac{\tilde{\gamma}^2 L_2}{m} \sum\nolimits_{r=0}^{t-1} e^{-\beta(t-r)\varepsilon} \sqrt{\mathbf{E}\left[\|V_t^m\|_1^2\right]} \sqrt{1+\mathbf{E}\left[\|V_{t-1}^m\|_1^2\right]}. \quad \square$$

Though the error established in the last claim is stable as ε decreases, we have mentioned in Section 3.2 that when $\mathcal{M}$ is a perturbation of identity, by using iteration (10) instead of (9), one might be able to obtain errors that vanish as ε decreases (keeping m fixed). This is the subject of Theorem 11 below which, like Theorem 10 relies crucially on self-averaging of the compression errors. Note that iteration (10) typically requires $\mathcal{O}(n)$ operations per iteration and storage of length n vectors. We have the following theorem demonstrating the decrease in error with ε in this setting.

Theorem 11. *Suppose that the iterates V_t^m of (10) remain in $\mathcal{X} \subset \mathbb{C}^n$ and that (27) holds. Under the same assumptions on $\mathcal{M}$ as in Theorem 10, the error at step $t \leq T/\varepsilon$ satisfies the bound*

$$\sup_{t \leq T/\varepsilon} |||V_t^m - v_t||| \leq \frac{\sqrt{\varepsilon}\gamma}{\sqrt{m}} (L_1+L_2) L_1 \left(\frac{1-e^{-2\beta T}}{2\beta} \right)^{\frac{1}{2}} \sup_{r<T/\varepsilon} \sqrt{\mathbf{E}\left[\|V_r^m\|_1^2\right]} + \frac{\varepsilon\gamma^2 L_2 L_1^2}{m} \left(\frac{1-e^{-\beta T}}{\beta} \right) \sup_{r<T/\varepsilon} \mathbf{E}\left[\|V_r^m\|_1^2\right].$$

Proof. By a very similar argument as in the proof of Theorem 7 arrive at the bound

$$|||V_t^m - v_t||| \leq \left(\sum\nolimits_{r=0}^{t-1} |||\mathcal{M}_{r+1}^t(V_r^m + \varepsilon b(Y_r^m)) - \mathcal{M}_{r+1}^t(V_r^m + \varepsilon b(V_r^m))|||^2 \right)^{1/2} + \left(\sum\nolimits_{r=0}^{t-1} |||\mathbf{E}\left[\mathcal{M}_{r+1}^t(V_r^m + \varepsilon b(Y_r^m)) \,|\, V_r^m\right] - \mathcal{M}_r^t(V_r^m)|||^2 \right)^{1/2} + \sum\nolimits_{r=0}^{t-1} |||\mathbf{E}\left[\mathcal{M}_{r+1}^t(V_r^m + \varepsilon b(Y_r^m)) \,|\, V_r^m\right] - \mathcal{M}_{r+1}^t(V_r^m + \varepsilon b(V_r^m))|||$$

which, also as in that proof, is bounded above by

$$|||V_t^m - v_t||| \leq (L_1+L_2) \left(\varepsilon^2 \sum\nolimits_{r=0}^{t-1} \alpha^{2(t-r-1)} |||b(Y_r^m) - b(V_r^m)|||^2 \right)^{1/2} + L_2 \varepsilon^2 \sum\nolimits_{r=0}^{t-1} \alpha^{t-r} |||b(Y_r^m) - b(V_r^m)|||^2.$$

From (34) and Lemma 1 we find that

$$|||b(Y_r^m) - b(V_r^m)||| \leq L_1 |||Y_r^m - V_r^m|||.$$

The rest of the argument proceeds exactly as proof of Theorem 7. $\square$

5 Compression rules

In this section we give a detailed description of the compression rule used in our numerical simulations as well as several others and analysis of the accuracy of those schemes. Our compression schemes have the general form in Algorithm 1. Programmed efficiently, and assuming that v has

Data: $v \in \mathbb{C}^n$ with all nonzero entries, $m \in \mathbb{N}$
Result: $V = \Phi^m(v) \in \mathbb{C}^n$ with at most m nonzero entries
$\ell = 0$;
$V = 0$;
$r = \|v\|_1/m$;
$I = \arg\max_i\{|v_i|\}$;
while $|v_I| \geq r$ **do**
  $\ell = \ell + 1$;
  $V_I = v_I$;
  $v_I = 0$;
  $r = \|v\|_1/(m-\ell)$;
  $I = \arg\max_i\{|v_i|\}$;
end
For each j let N_j be a non-negative random integer with $\mathbf{E}[N_j] = (m-\ell)|v_j|/\|v\|_1$;
finally, for $i = 1, 2, \ldots, n$ set

$$V_i = N_i \frac{v_i \|v\|_1}{|v_i|(m-\ell)};$$

Algorithm 1: A simple compression rule.

exactly n nonzero entries, all of the schemes we discuss in this section will require at most $\mathcal{O}(n)$ floating point operations including the generation of as few as one uniform random variate and $\mathcal{O}(n + m\log n)$ floating point comparisons. It is likely that better compression schemes are possible, for example by incorporation of ideas from [30]. The reader should note that in this section n represents the number of non-zero entries in the input vector v of the compression rule and not the dimension associated with a particular problem (which may be much larger).

What is left unspecified by the steps in Algorithm 1 is the choice of the joint distribution of the N_j. For many applications, the details of this choice are crucial. The simplest possibilities, such as choosing the N_j to be distributed according to a multinomial distribution with probabilities $p_j = |v_j|/\|v\|_1$, can easily result in unstable schemes. Before detailing better choices, we remark on the deterministic procedure preceding the generation of the N_j. Let σ be a perturbation of $\{1, 2, \ldots, n\}$ so that the elements of v_σ have decreasing magnitude and let

$$\tau_v^m = \min\left\{0 \leq \ell \leq m : \sum\nolimits_{j=\ell+1}^{n} |v_{\sigma_j}| \geq (m-\ell)|v_{\sigma_{\ell+1}}|\right\}. \tag{36}$$

τ_v^m is the number of entries of v that are exactly preserved (in V) during the **while** loop in Algorithm 1. Note that if the number of nonzero entries in v exceeds m then $\tau_v^m < m$.

Lemma 4. *τ_v^m satisfies*

$$\sum\nolimits_{j=\tau_v^m+1}^{n} |v_{\sigma_j}| \leq \frac{m-\tau_v^m}{m}\|v\|_1.$$

Proof. Observe that if $\tau_v^m > 0$, then condition

$$\sum\nolimits_{j=\ell+1}^{n} |v_{\sigma_j}| \leq \frac{m-\ell}{m}\|v\|_1$$

holds for $\ell = 0$. Assume that

$$\sum\nolimits_{j=\ell}^{n} |v_{\sigma_j}| \leq \frac{m-\ell+1}{m}\|v\|_1.$$

for some $\ell \leq \tau_v^m$. From the definition of τ_v^m and the fact that $\ell \leq \tau_v^m$, we must also have that

$$\frac{1}{m-\ell}\sum\nolimits_{j=\ell+1}^{n} |v_{\sigma_j}| < |v_{\sigma_{\ell+1}}|.$$

Combining the last two inequalities yields

$$\sum\nolimits_{j=\ell+1}^{n} |v_{\sigma_j}| \leq \frac{m-\ell}{m}\|v\|_1. \qquad \square$$

As we will find below, this Lemma implies that our compression rules will have errors that decrease as τ_v^m increases, justifying the presence of this additional deterministic step. In cases where the $|v_{\sigma_j}|$ decrease rapidly with j, the reduction in error can be dramatic.

Returning now to a more detailed description of the joint distribution of the N_j we begin with a particularly simple choice (and one that will have to be modified later) suggested by common practice in diffusion Monte Carlo applications. For each j generate an independent uniform random variable $U^{(j)}$ in $(0,1)$, and set

$$N_j = \left\lfloor \frac{(m-\ell)|v_j|}{\|v\|_1} + U^{(j)} \right\rfloor. \tag{37}$$

One can easily check that $\mathbf{E}[N_j] = (m-\ell)|v_j|/\|v\|_1$. The next lemma characterizes the error resulting from this choice. Because we have a particular interest in cases in which $\mathcal{M}$ is a perturbation of identity and because, in those cases

$$|||\Phi_t(V_t^m) - V_t^m||| = |||\Phi_t(Y_{t-1}^m + \varepsilon b(Y_{t-1}^m)) - Y_{t-1}^m - \varepsilon b(Y_{t-1}^m)|||$$

where $Y_{t-1}^m = \Phi_{t-1}^m(V_t^m)$ has at most m non-zero entries, we will pay close attention to the effects on our error bounds of a perturbation $w \in \mathbb{C}^n$ of a vector v with only m nonzero entries, i.e., $v \in \mathcal{S}_m$. The lemma includes an estimate of $\mathbf{E}\left[\|\Phi_t^m(v)\|_1^2\right]$ and of the probability of the event $\{\Phi_t(v) = 0\}$ which will both play a role in the discussion immediately following its proof.

Lemma 5. *For Φ_t defined by Algorithm 1 with (37) and for any vectors $v \in \mathcal{S}_m$ and $w \in \mathbb{C}^n$,*

$$|||\Phi_t(v+w) - v - w||| \leq \sqrt{2}\frac{\left(\sum_{j=\tau_{v+w}^m+1}^{n} |w_j|\right)^{\frac{1}{2}}\left(\sum_{j=\tau_{v+w}^m+1}^{n} |v_j+w_j|\right)^{\frac{1}{2}}}{\sqrt{m-\tau_{v+w}^m}}. \tag{38}$$

The righthand side of the (38) is itself bounded above by

$$\sqrt{2}\frac{\|w\|_1^{\frac{1}{2}}\|v+w\|_1^{\frac{1}{2}}}{\sqrt{m}}. \tag{39}$$

Concerning the size of the resampled vector we have the bound

$$\mathbf{E}\left[\|\Phi_t^m(v+w)\|_1^2\right] = \|v+w\|_1^2 + 2\frac{\|v+w\|_1\|w\|_1}{m}.$$

Finally, if $\tau_{v+w}^m > 0$ then $\mathbf{P}[\Phi_t(v+w) = 0] = 0$. If $\tau_{v+w}^m = 0$ then

$$\mathbf{P}[\Phi_t(v) = 0] \leq \left(\min\left\{\frac{\|w\|_1}{\|v+w\|_1}, \frac{1}{e}\right\}\right)^m.$$

Proof. First we assume that, for all j, $|v_j + w_j| \leq \|v+w\|/m$. We will remove this assumption later. With this assumption in place, $N_j \in \{0,1\}$ and the **while** loop in Algorithm 1 is inactive so that

$$f^{\mathrm{H}}\Phi_t(v+w) = \sum\nolimits_{j=1}^{n} \bar{f}_j \frac{\|v+w\|}{m} N_j,$$

$$\mathbf{E}\left[|f^{\mathrm{H}}\Phi_t(v+w) - v - w|^2\right] = \mathbf{E}\left[\left|\sum\nolimits_{j=1}^{n} \bar{f}_j\left(\frac{\|v+w\|}{m}N_j - v - w\right)\right|^2\right].$$

The random variables in the sum are independent so the last expression becomes

$$\mathbf{E}\big[|f^{\mathrm{H}}\Phi_t(v+w)-f^{\mathrm{H}}(v+w)|^2\big]=\sum_{j=1}^n\mathbf{E}\left[\left|\frac{\|v+w\|}{m}N_j-v-w\right|^2\right]=\frac{\|v+w\|_1^2}{m^2}\sum_{j=1}^n\mathbf{var}\left[N_j\right].$$

Since $N_j\in\{0,1\}$, the expression for the variance of N_j becomes

$$\mathbf{var}\left[N_j\right]=\mathbf{E}\left[N_j\right](1-\mathbf{E}\left[N_j\right])=\frac{m(v_j+w_j)}{\|v+w\|_1}\left(1-\frac{m(v_j+w_j)}{\|v+w\|_1}\right)$$

so that

$$\mathbf{E}\left[|f^{\mathrm{H}}\Phi_t(v+w)-v-w|^2\right]=\frac{\|v+w\|_1^2}{m^2}\left[m-\left(\frac{m}{\|v+w\|_1}\right)^2\|v+w\|_2^2\right].$$

Because this scheme does not depend on the ordering of the entries of $v+w$ we can assume that the entries have been ordered so that $v_j=0$ for $j>m$. In this case we can write

$$\|v+w\|_2^2=\sum_{j=1}^m|v_j+w_j|^2+\sum_{j=m+1}^n|w_j|^2\geq\frac{1}{m}\left(\sum_{j=1}^m|v_j+w_j|\right)^2\geq\frac{1}{m}\left(\|v+w\|_1-\|w\|_1\right)^2,$$

which then implies that

$$\begin{aligned}\mathbf{E}\left[|f^{\mathrm{H}}\Phi_t(v+w)-f^{\mathrm{H}}(v+w)|^2\right]&\leq\frac{\|v+w\|_1^2}{m}\left(1-\frac{1}{\|v+w\|_1^2}\left(\|v+w\|_1-\|w\|_1\right)^2\right)\\&\leq\frac{\|w\|_1}{m}\left(2\|v+w\|_1-\|w\|_1\right)\leq\frac{2\|w\|_1\|v+w\|_1}{m}.\end{aligned}$$

Expression (38) follows from the definition of τ_{v+w}^m and the fact that v_{σ_j} for $j\leq\tau_{v+w}^m$ are preserved exactly by Φ_t. The upper bound in (39) follows after an application of Lemma 4.

In bounding the size of $\Phi_t^m(v+w)$ we will again assume that $\tau_{v+w}^m=0$ and that the entries have been ordered so that $v_j=0$ for $j>m$. The size of the resampled vector can be bounded by first noting that, since the N_j are independent and are in $\{0,1\}$,

$$\begin{aligned}\mathbf{E}\left[\left(\sum_{j=1}^n N_j\right)^2\right]&=\sum_{j=1}^n\frac{m|v_j+w_j|}{\|v+w\|_1}+2\sum_{i=1}^n\sum_{j=i+1}^n\frac{m|v_i+w_i|}{\|v+w\|_1}\frac{m|v_j+w_j|}{\|v+w\|_1}\\&=\sum_{j=1}^n\left(\frac{m|v_j+w_j|}{\|v+w\|_1}\right)^2+2\sum_{i=1}^n\sum_{j=i+1}^n\frac{m|v_i+w_i|}{\|v+w\|_1}\frac{m|v_j+w_j|}{\|v+w\|_1}\\&\qquad+\sum_{j=1}^n\frac{m|v_j+w_j|}{\|v+w\|_1}-\left(\frac{m|v_j+w_j|}{\|v+w\|_1}\right)^2\\&=m^2+\sum_{j=1}^n\frac{m|v_j+w_j|}{\|v+w\|_1}-\left(\frac{m|v_j+w_j|}{\|v+w\|_1}\right)^2.\end{aligned}$$

Breaking up the last sum in this expression we find that

$$\begin{aligned}\sum_{j=1}^m\frac{m|v_j+w_j|}{\|v+w\|_1}-\left(\frac{m|v_j+w_j|}{\|v+w\|_1}\right)^2&\leq m\sum_{j=1}^m\frac{|v_j+w_j|}{\|v+w\|_1}-m\left(\sum_{j=1}^m\frac{|v_j+w_j|}{\|v+w\|_1}\right)^2\\&\leq m\left(1-\sum_{j=1}\frac{|v_j+w_j|}{\|v+w\|_1}\right)\leq\frac{m\|w\|_1}{\|v+w\|_1}\end{aligned}$$

and that

$$\sum_{j=m+1}^n\frac{m|w_j|}{\|v+w\|_1}-\left(\frac{m|w_j|}{\|v+w\|_1}\right)^2\leq\frac{m\|w\|_1}{\|v+w\|_1}$$

so that

$$\mathbf{E}\Big[\Big(\sum_{j=1}^n N_j\Big)^2\Big]\leq m^2+2\frac{m\|w\|_1}{\|v+w\|_1}.$$

It follows then that (at least when $\tau_{v+w}^m=0$),

$$\mathbf{E}\left[\|\Phi_t^m(v+w)\|_1^2\right]\leq\|v+w\|_1+2\frac{\|v+w\|_1\|w\|_1}{m}.$$

Writing the corresponding formula for $\tau^m_{v+w} > 0$ and applying Lemma 4 gives the bound in the statement of the lemma.

Finally we consider the probability of the event $\{\Phi^m_t(v+w) = 0\}$. If $\tau^m_{v+w} = 0$ then $N_j \in \{0, 1\}$ so that $\mathbf{P}[N_j = 0] = 1 - m|v_j + w_j|/\|v + w\|_1$, and, since the N_j are independent,

$$\mathbf{P}[N_j = 0 \text{ for all } j] = \prod_{j=1}^{n} \left(1 - \frac{m|v_j + w_j|}{\|v + w\|_1}\right) \leq \prod_{j \leq n,\, v_j \neq 0} \left(1 - \frac{m|v_j + w_j|}{\|v + w\|_1}\right)$$

The first product in the last display is easily seen to be bounded above by e^{-m}. The second product is maximized subject to the constraint

$$\sum\nolimits_{j \leq n,\, v_j \neq 0} \left(1 - \frac{m|v_j + w_j|}{\|v + w\|_1}\right) \leq \frac{m\|w\|_1}{\|v + w\|_1}$$

when the terms in the product are all equal, in which case we get

$$\mathbf{P}[N_j = 0 \text{ for all } j] \leq \left(\frac{\|w\|_1}{\|v + w\|_1}\right)^m. \qquad \square$$

In practice one would need to modify the choice in (37) to avoid several potential drawbacks. First, With any application of the resulting compression rule (assuming that $\tau^m_v = 0$) there is a non-zero probability that $\Phi^m_t(v) = 0$ (i.e., that all the $N_j = 0$). As Lemma 5 demonstrates, the probability of this event is extremely small. The issue can be avoided by simply sampling $\Phi^m_t(v)$ until $\Phi^m_t(v) \neq 0$, i.e., sample $\Phi^m_t(v)$ conditioned on the event $\{\Phi^m_t(v) \neq 0\}$. The resulting bias and additional cost are negligible (as can be seen using the estimate of the probability of the event $\{\Phi^m_t(v) = 0\}$ given in Lemma 5). Another potential drawback is the possibility that $\Phi^m_t(v)$ has more than m non-zero entries. This can be easily avoided at insignificant additional cost by randomly setting entries of $\Phi^m_t(v)$ to zero (and renormalizing the result) using the systematic compression procedure described below. The resulting compression of v has the same ℓ^1 norm as $\Phi^m_t(v)$ and exactly the same expectation. The resulting statistical error is also insignificant (a fact that can be established using the estimate on the size of $\Phi^m_t(v)$ in Lemma 5).

A much more significant drawback associated with the choice (37) is the requirement that we generate n (assuming v has n non-zero components) independent uniform random variates. Depending on the cost of evaluating $\mathcal{M}(V^m_t)$, the generation of so many random variables could constitute a significant portion of the cost of the overall scheme. An unbiased method that is observed in practice to behave similarly to the scheme in (37) but which requires only a single random variate, exactly preserves the ℓ^1 norm of the input vector, and produces an output vector with exactly m non-zero entries, can be defined by, for $k = 1, 2, \ldots, m - \ell$, setting

$$I_k = \max\left\{i : \sum\nolimits_{j=1}^{i-1} |v_j| \leq \|v\|_1 U^{(k)} < \sum\nolimits_{j=1}^{i} |v_j|\right\} \tag{40}$$

and

$$N_j = \begin{cases} 1 & \text{if } j \in \{I_k\}, \\ 0 & \text{if } j \notin \{I_k\}, \end{cases} \tag{41}$$

where

$$U^{(k)} = \frac{1}{m - \ell}(k - 1 + U). \tag{42}$$

and $U \in (0, 1)$ is a single uniformly chosen random variable. The random sampling step in the resulting compression scheme corresponds to the so called systematic resampling scheme used frequently in the context of sequential Monte Carlo (see e.g. [16]). It is well known in that context, that systematic sampling can fail to converge for certain sequences of input vectors (they need to change as one increases m or the method will converge). Nonetheless, systematic resampling is

an extremely popular technique and in many applications results in lower error than competing schemes. Similarly, for certain choices of v, the compression scheme specified by (41) and (42) is guaranteed to converge only as m/n approaches 1. Supporting the use of the scheme for problems in which $\mathcal{M}$ is a perturbation of the identity, it is possible to prove that if, after exact preservation of all entries $v_{\sigma_j} + w_{\sigma_j}$ with $j \leq \tau^m_{v+w}$, the remaining entries are re-ordered so that entries with $v_j = 0$ appear at the end of the list of remaining entries in $v + w$, then (assuming $\tau^m_{v+w} = 0$ for simplicity)

$$\|\Phi^m(v + w) - v - w\|_1 \leq 2\|w\|_1$$

when $\|w\|_1 < \|v + w\|_1/m$. Though we have not used it in our numerical tests, the additional rearrangement step imposes no substantial additional cost and is essential to this bound.

Slight modifications of (41) and (42) do result in provably convergent schemes. For example, another commonly used resampling scheme in sequential Monte Carlo called stratified resampling (see e.g. [16]) corresponds to choosing each uniform random variable $U^{(k)} \in [(k-1)/(m-\ell), k/(m-\ell))$ in (40) independently. Again the resulting scheme is unbiased and preserves the ℓ^1 norm of the input vector. The following lemma provides a bound on the scheme's error.

Lemma 6. *Let σ be a perturbation of $\{1, 2, \ldots, n\}$ so that the elements of $v_\sigma + w_\sigma$ have decreasing magnitude and let τ^m_{v+w} be defined as in* (36) *with v replaced by $v + w$. For the compression rule defined in Algorithm 1 with $U^{(k)}$ chosen independently in* (40)*,*

$$\begin{aligned}\|\|\Phi^m_t(v + w) - v - w\|\| \leq \frac{1}{\sqrt{m - \tau^m_{v+w}}}\Big(\sum\nolimits_{j=\tau^m_{v+w}+1}^{n} |v_{\sigma_j} + w_{\sigma_j}|\Big) \\ \times \min\left\{1, \left[\frac{(m - \tau^m_{v+w})\big(\sum_{j=\tau^m_{v+w}}^{n} |w_{\sigma_j}|\big)}{\sum_{j=\tau^m_{v+w}}^{n} |v_{\sigma_j} + w_{\sigma_j}|}\right]^{1/2}\right\}. \quad (43)\end{aligned}$$

for any pair $v, w \in \mathbb{C}^n$ with $v \in \mathcal{S}^m$. The right hand side in (43) *is itself bounded above by*

$$\frac{\|v + w\|_1}{\sqrt{m}} \min\left\{1, \left[\frac{(m - \tau^m_{v+w})\|w\|_1}{\|v + w\|_1}\right]^{1/2}\right\}. \quad (44)$$

Proof. Because the scheme exactly preserves entries $v_{\sigma_\ell} + w_{\sigma_\ell}$ for $\ell \leq \tau^m_{v+w}$ and, for all $\ell > \tau^m_{v+w}$,

$$|v_{\sigma_\ell} + w_{\sigma_\ell}| < \frac{1}{m - \tau^m_{v+w}} \sum\nolimits_{j=\tau^m_{v+w}+1}^{n} |v_{\sigma_j} + w_{\sigma_j}|,$$

it suffices to consider the case in which, for all j, $|v_j + w_j| < \|v + w\|_1/m$. Expression (43) will follow by setting to 0 entries with index σ_j for $j \leq \tau(m)$ in both v and w and replacing m in the eventual bound by $m - \tau^m_{v+w}$.

Let $V = \Phi^m(v + w)$ and suppose $f \in \mathbb{C}^n$ with $\|f\|_\infty \leq 1$. Our goal is to establish an upper bound for

$$\mathbf{E}\left[|f^{\mathrm{H}}V - f^{\mathrm{H}}(v + w)|^2\right] = \mathbf{var}\left[f^{\mathrm{H}}V\right]$$

where **var** on the right hand side of the last display represents the variance of the random variable $f^{\mathrm{H}}V$ and we have used the easily established fact that $\mathbf{E}[V] = v + w$. Let $s_0 = 0$ and

$$s_j = \frac{1}{\|v + w\|_1} \sum\nolimits_{i=1}^{j} |v_i + w_i|$$

for $j = 1, 2, \ldots, n$, and define the function $I(u)$ for $u \in [0, 1]$ by

$$I(u) = \max\{j : s_{j-1} \leq u < s_j\}.$$

Then we can write

$$f^{\mathrm{H}}V = \frac{\|v + w\|_1}{m} \sum\nolimits_{k=1}^{m} \bar{f}_{I(U^{(k)})} \frac{v_{I(U^{(k)})}}{|v_{I(U^{(k)})}|}$$

so, since the $U^{(k)}$ are independent,

$$\mathbf{var}\left[f^{\mathrm{H}}V\right] = \frac{\|v+w\|_1^2}{m^2}\sum\nolimits_{k=1}^{m}\mathbf{var}\left[\bar{f}_{I(U^{(k)})}\frac{v_{I(U^{(k)})}}{|v_{I(U^{(k)})}|}\right].$$

Now let

$$p_{k,j} = m\left|[s_{j-1},s_j)\cap[(k-1)/m,k/m)\right|.$$

In terms of the $p_{k,j}$ we can write

$$\mathbf{E}\left[\bar{f}_{I(U^{(k)})}\frac{v_{I(U^{(k)})}}{|v_{I(U^{(k)})}|}\right] = \sum\nolimits_{j=1}^{n}\bar{f}_j p_{k,j}$$

and

$$\mathbf{var}\left[\bar{f}_{I(U^{(k)})}\frac{v_{I(U^{(k)})}}{|v_{I(U^{(k)})}|}\right] = \sum\nolimits_{j=1}^{n}|\bar{f}_j|^2 p_{k,j} - \left|\sum\nolimits_{i=1}^{n}\bar{f}_i\frac{v_i}{|v_i|}p_{k,i}\right|^2.$$

For $k=1,2,\ldots,m$, let

$$j_k = \arg\max_{j\le n} p_{k,j} \qquad \text{and} \qquad r_k = p_{k,j_k}.$$

We find that

$$\begin{aligned}\mathbf{var}\left[\bar{f}_{I(U^{(k)})}\frac{v_{I(U^{(k)})}}{|v_{I(U^{(k)})}|}\right] &= |\bar{f}_{j_k}|^2 r_k + \sum\nolimits_{j\neq j_k}|\bar{f}_j|^2 p_{k,j} - \left|\bar{f}_{j_k}\frac{v_{j_k}}{|v_{j_k}|}r_k + \sum\nolimits_{i\neq j_k}\bar{f}_i\frac{v_i}{|v_i|}p_{k,i}\right|^2\\ &\le |f_{j_k}|^2 r_k + \sum\nolimits_{j\neq j_k}|f_j|^2 p_{k,j} - |f_{j_k}|^2 r_k^2 + 2|f_{j_k}|r_k\left|\sum\nolimits_{i\neq j_k}\bar{f}_i\frac{v_i}{|v_i|}p_{k,i}\right|\\ &\le r_k(1-r_k) + (1-r_k) + 2r_k(1-r_k) \le 4(1-r_k).\end{aligned}$$

It remains to bound the sum $\sum_{k=1}^{m}(1-r_k)$ in terms of $\|w\|_1/\|v+w\|_1$. Define $\alpha_0 = 0$ and for $k=1,2,\ldots,m$. Let α_k be the index of the kth entry of $v+w$ with $v_k \neq 0$. Notice that, since $|v_j+w_j| < \|v+w\|_1/m$,

$$s_{\alpha_k} \le \frac{k}{m} + \frac{\|w\|_1}{\|v+w\|_1} < \frac{k+1}{m}. \tag{45}$$

Moreover, notice that

$$\sum\nolimits_{j=1}^{m}\frac{|v_{\alpha_j}+w_{\alpha_j}|}{\|v+w\|_1} \ge 1 - \frac{\|w\|_1}{\|v+w\|_1},$$

and that if

$$\sum\nolimits_{j=1}^{k+1}\frac{|v_{\alpha_j}+w_{\alpha_j}|}{\|v+w\|_1} \ge \frac{k+1}{m} - \frac{\|w\|_1}{\|v+w\|_1},$$

then

$$\begin{aligned}\sum\nolimits_{j=1}^{k}\frac{|v_{\alpha_j}+w_{\alpha_j}|}{\|v+w\|_1} &= -\frac{|v_{\alpha_{k+1}}+v_{\alpha_{k+1}}|}{\|v+w\|_1} + \sum\nolimits_{j=1}^{k+1}\frac{|v_{\alpha_j}+w_{\alpha_j}|}{\|v+w\|_1}\\ &\ge -\frac{1}{m} + \frac{k+1}{m} - \frac{\|w\|_1}{\|v+w\|_1} = \frac{k}{m} - \frac{\|w\|_1}{\|v+w\|_1},\end{aligned}$$

allowing us to conclude by induction that

$$s_{\alpha_k} \ge \sum\nolimits_{j=1}^{k}\frac{|v_{\alpha_j}+w_{\alpha_j}|}{\|v+w\|_1} \ge \frac{k}{m} - \frac{\|w\|_1}{\|v+w\|_1} > \frac{k-1}{m}. \tag{46}$$

On the other hand, these bounds, and the fact that $|v_j+w_j| < \|v+w\|_1/m$, also imply that

$$s_{\alpha_k-1} \le \frac{k-1}{m} + \frac{\|w\|_1}{\|v+w\|_1} < \frac{k}{m} \qquad \text{and} \qquad s_{\alpha_k-1} \ge \frac{k-1}{m} - \frac{\|w\|_1}{\|v+w\|_1} > \frac{k-2}{m}. \tag{47}$$

From the bounds (45) and (46) we conclude that within any interval $[(k-1)/m, k/m)$ we can find at most two points in the set $\{s_{\alpha_j}\}$ and those two points are $s_{\alpha_{k-1}}$ and s_{α_k}. Likewise, we can find at most the two points s_{α_k-1} and $s_{\alpha_{k+1}-1}$ from the set $\{s_{\alpha_j-1}\}$ in $[(k-1)/m, k/m)$.

If s_{α_k} is contained in the interval $[(k-1)/m, k/m)$ then so is the complete interval $(\max\{(k-1)/m, s_{\alpha_k-1}\}, s_{\alpha_k})$. If the entire interval $(s_{\alpha_k-1}, s_{\alpha_k})$ is contained in $[(k-1)/m, k/m)$ then $r_k \geq m|v_{\alpha_k} + w_{\alpha_k}|/\|v+w\|_1$. Since, for any i,

$$|v_{\alpha_j} + w_{\alpha_j}| + \sum\nolimits_{i \neq j} |v_{\alpha_j} + w_{\alpha_j}| \geq \|v+w\|_1 - \|w\|_1,$$

and $|v_j + w_j| < \|v+w\|_1/m$, we find that, for any j,

$$|v_{\alpha_j} + w_{\alpha_j}| \geq \|v+w\|_1/m - \|w\|_1$$

and therefore that

$$1 - r_k \leq \frac{m\|w\|_1}{\|v+w\|_1}.$$

If $s_{\alpha_k-1} \leq (k-1)/m$ then $r_k \geq m s_{\alpha_k} - (k-1)$ and, from (46),

$$m s_{\alpha_k} - (k-1) = m\left(s_{\alpha_k} - \frac{k}{m}\right) + 1,$$

so that

$$1 - r_k \leq \frac{m\|w\|_1}{\|v+w\|_1}.$$

On the other hand, if $s_{\alpha_k} \geq k/m$ then s_{α_k-1} is in the interval $[(k-1)/m, k/m)$ along with the complete interval $(s_{\alpha_k-1}, k/m)$. Therefore, $r_k \geq k - m s_{\alpha_k-1}$ and, from (47),

$$k - m s_{\alpha_k-1} \geq 1 + m\left(\frac{k-1}{m} - s_{\alpha_k-1}\right) \geq 1 + \frac{m\|w\|_1}{\|v+w\|_1},$$

so that again,

$$1 - r_k \leq \frac{m\|w\|_1}{\|v+w\|_1}.$$

Since $r_k \geq 0$ the above bounds on $1 - r_k$ allow us to conclude that

$$\sum\nolimits_{k=1}^{m} (1 - r_k) \leq m \min\left\{1, \frac{m\|w\|_1}{\|v+w\|_1}\right\},$$

so that

$$\mathbf{E}\left[|f^{\mathrm{H}}V - f^{\mathrm{H}}(v+w)|^2\right] \leq 4\frac{\|v+w\|_1}{m} \min\left\{1, \frac{m\|w\|_1}{\|v+w\|_1}\right\},$$

concluding the proof of (43). The upper bound (44) follows by an application of Lemma 4. □

Though the bounds in Lemma 6 are somewhat crude, they are sharp in one respect. One can find sequences of vectors achieving an error scaling as $\|v+w\|_1^{1/2}\|w\|_1^{1/2}$ as m is increased. As a consequence, when $\mathcal{M}$ is an ε perturbation of the identity, we can expect $\|\|\Phi_t(V_t^m) - V_t^m\|\|$ to be of order $\sqrt{\varepsilon}$ for this scheme, but we cannot guarantee that the error will remain controllable with m (e.g. $\mathcal{O}(m^{-1/2})$) as ε decreases. This point demonstrates that careful attention to the compression scheme is required in cases in which $\mathcal{M}$ is a perturbation of the identity. In that context, even schemes inspired by well regarded resampling strategies may not be effective.

6 Numerical tests

In this section we describe the application of the framework above to particular matrices arising in (i) the computation of the per-spin partition function of the 2D Ising model, (ii) the spectral gap of a diffusion process governing the evolution of a system of up to five, 2-dimensional particles (i.e., up to ten spatial dimensions), and (iii) a free energy landscape for that process. The corresponding numerical linear algebra problems are, respectively, (i) computing the dominant eigenvalue/eigenvector of matrices up to size $10^{15} \times 10^{15}$, (ii) computing the second largest eigenvalue/eigenvector of matrices up to size $10^{20} \times 10^{20}$, and (iii) solving a linear system involving exponentiation of matrices up to size $10^{20} \times 10^{20}$. Aside from sparsity, these matrices have no known readily exploitable structure for computations.

The reader may wonder why we consider random compressions instead of simple thresholding, i.e., a compression rule in which, if σ_j is the index of the jth largest (in magnitude) of v, v_{σ_j} is simply set to zero for all $j > m$ (the resulting vector can be normalized to preserve ℓ^1-norm or not). In the rest of this paper we will refer to methods using such a compression rule as truncation-by-size (TbS) schemes. TbS schemes have been considered by many authors (see e.g. [26, 50, 42, 43]) and are a natural approach. Note however that the error (if the compression is not normalized),

$$\left| \sum_{j>m} \bar{f}_{\sigma_j} v_{\sigma_j} \right|,$$

for the thresholding compression can be as large as $\|f\|_\infty \|v\|_1 (1 - m/n)$ which only vanishes if m is increased faster than n. In contrast, the random compressions above can have vanishing error even when n is infinite. This observation is key to understanding the substantial reduction in error we find for our fast randomized scheme over TbS in numerical results presented in this section.

In order for the FRI approach to yield significant performance improvements, one must use an efficient implementation of matrix by sparse vector multiplication. In the examples below we list the i, j pairs for which the product $K_{ij} v_j$ is nonzero, then sort the products according to the i-index and finally, add the products with common i. This is a simple and sub-optimal solution. In particular, if the entries of the matrix K can be efficiently accessed column by column then the sorting step can be avoided. More details can be found in the example code available here [55].

6.1. A transfer matrix eigenproblem In this example we find the dominant eigenvalue λ_* of the transfer matrix K of the 2-dimensional ℓ-spin Ising model. This eigenvalue is the per-spin partition function of the infinite spin Ising model, i.e.,

$$\lambda_*(T, B) = \lim_{\ell \to \infty} \left(\sum_\sigma e^{\frac{1}{T} \sum_{|(i,j)-(i',j')|=1} \sigma_{ij} \sigma_{i'j'} + B\sigma_{ij}} \right)^{1/\ell}$$

where $\sigma_{ij} \in \{-1, 1\}$ and the sum in the exponent is over pairs of indices on a square 2-dimensional, periodic lattice with ℓ sites. The outer sum is over all 2^ℓ possible values of σ, and for larger ℓ, one cannot possibly compute it directly. The matrix K is $2^\ell \times 2^\ell$. For example, in the case $\ell = 3$,

$$K = \begin{bmatrix} a & & & a^{-1} & & & & \\ b & & & b^{-1} & & & & \\ & a & & & a^{-1} & & & \\ & b & & & b^{-1} & & & \\ & & b & & & b^{-1} & & \\ & & c & & & c^{-1} & & \\ & & & b & & & b^{-1} & \\ & & & c & & & c^{-1} & \end{bmatrix} \tag{48}$$

where

$$a = e^{(2-B)/T}, \quad b = e^{-B/T}, \quad c = e^{-(2+B)/T}.$$

We therefore also cannot hope to apply the power method (or its relatives) directly to K when ℓ is large. In our experiments we set $T = 2.2$, $B = 0.01$, and $\ell = 50$ so that $n = 2^\ell > 10^{15}$, We apply both the FRI and TbS, $\mathcal{O}(1)$ schemes to computing λ_* as well as to computing the sum of all components of the corresponding eigenvector, v_* (normalized to have sum equal to 1), with index greater than or equal to 2^{49}, i.e.,

$$f_* = \sum\nolimits_{j \geq n/2} (v_*)_j.$$

Knowledge of the partition function λ_* as a function of temperature T and field strength B allows one to determine useful quantities such as the average magnetization (sum of spin values) and to diagnose phase transitions [26]. Our choice to estimate λ_* and f_* is motivated in part by the fact that these quantities can be approximated accurately by the corresponding values for smaller Ising systems. We will compare our results to those for the 24-spin Ising model which we can solve by standard power iteration. For an effective specialized method for this problem see [44]. A simple, educational implementation of Fast Randomized Iteration applied to this problem can be found here [55].

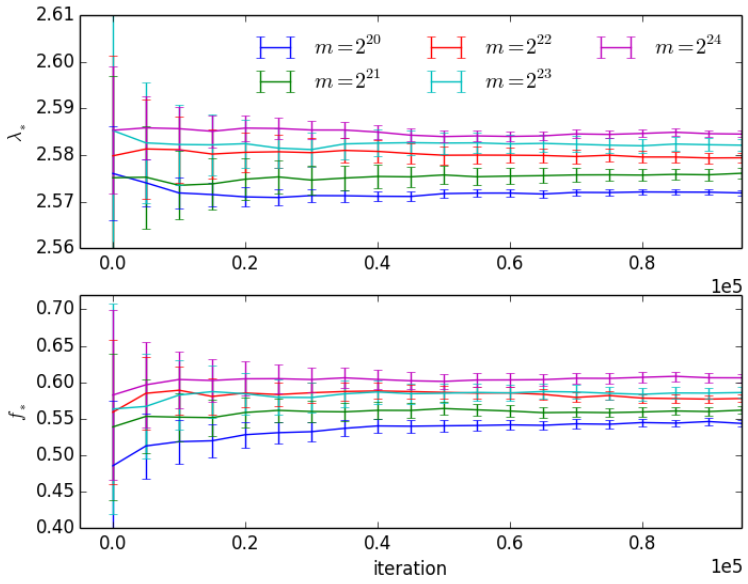

Figure 1: **Top.** Trajectory averages of the approximation, Λ_t^m, of the partition function for the 50-spin Ising model with $B = 0.01$ and $T = 2.2$, with 95% confidence intervals[6] as computed by the FRI method with $m = 2^k$ for $k = 20$, 21, 22, 23, and 24. The best (highest m) estimate for λ_* is 2.584 a difference of roughly $1B$ from the value for the 24-spin Ising model. **Bottom.** Corresponding trajectory averages of the approximation, F_t^m, of the total weight of all components of v_* with index greater than or equal to 2^{49} with 95% confidence intervals for the FRI method. The best (highest m) estimate for f_* is 0.606, a difference of roughly $5B$ from the value for the 24-spin model.

In Figure 1 we report the trajectory averages of the approximations Λ_t^m and F_t^m generated by the FRI scheme (iteration (16) using Algorithm 1) with $m = 2^{20}$, 2^{21}, 2^{22}, 2^{23}, and 2^{24} and 10^5 total iterations. The best (highest m) approximation of λ_* is 2.584 and the best approximation of f_* is 0.606. The results for the 24-spin Ising problem are $\lambda_* \approx 2.596$ and $f_* \approx 0.658$ a difference of roughly $1B$ and $5B$ from the respective approximations generated by the FRI method. In

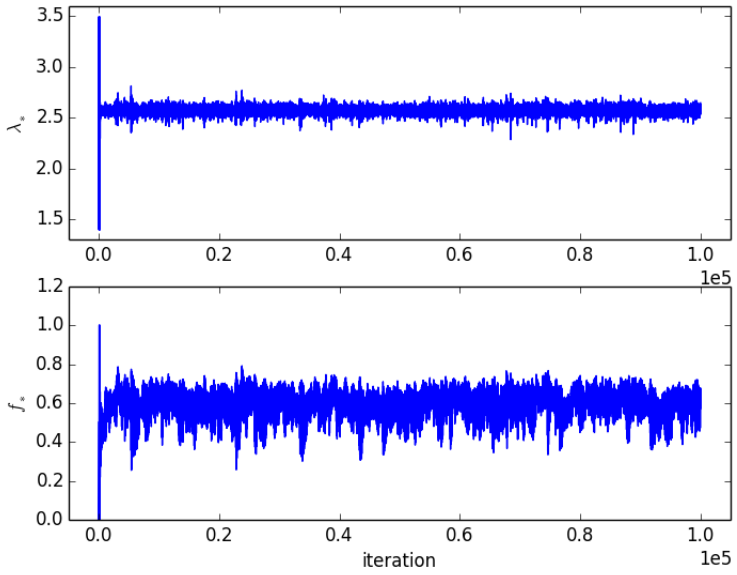

Figure 2: **Top.** Trajectory of the approximation, Λ_t^m, of the partition function for the 50-spin Ising model, for the FRI method with $m = 2^{24}$. The approximate integrated autocorrelation time for Λ_t^m is 20.5 iterations. **Bottom.** Corresponding trajectory of F_t^m as computed by the FRI method. The approximate integrated autocorrelation time for F_t^m is 274 iterations.

Figure 2 we plot the corresponding trajectories of Λ_t^m and F_t^m. These plots strongly suggest that the iteration equilibrates rapidly (relative to the total number of iterations). Indeed, we estimate the integrated autocorrelation times[6] of Λ_t^m and F_t^m to be 20.5 and 274 respectively. This in turn suggests that one could achieve a dramatic speedup by running many parallel and independent copies (replicas) of the simulation and averaging the resulting estimates of λ_* and f_* though we have not taken advantage of this here.

Figure 3 reports the analogous trajectories (see Equation (49) below) of Λ_t^m and F_t^m as generated by iteration (16) with the TbS scheme (iteration (16) using truncation-by-size) and the same values of m. The best (highest m) TbS approximation of λ_* is 2.545 and the best TbS approximation of f_* is 0.014, a difference of more than $5B$ and $64B$ respectively. In Figure 4 we plot the sums of the values of the approximation, V_t^m, of the dominant eigenvector of the Ising transfer matrix at $t = 10^5$ over 256 intervals of equal size out of the 2^{50} total indices. The top plot represents V_t^m as generated by the FRI method and the middle plot represents V_t^m as generated by the TbS approach. The TbS iteration has converged to a vector with nearly all of weight concentrated on very low indices. The bottom plot in Figure 4 represents the dominant eigenvector for the 24-spin Ising transfer matrix. The qualitative agreement with the realization of V_t^m as generated by the FRI method is much stronger than agreement with the result of the TbS method.

Remark 12. In this problem we compute the dominant eigenvalue λ_* and a projection f_* of the dominant eigenvector v_* of the matrix K defined in equation (48) using the FRI in conjunction

[6]According to the central limit theorem for Markov processes (assuming it holds), for large t the variance of the trajectory average of Λ_t^m should be $\sigma^2\tau/t$ where σ^2 is the infinite t limit of the variance of Λ_t and τ is the integrated autocorrelation time of Λ_t^m. Roughly, it measures the number of iterations between independent Λ_t^m.

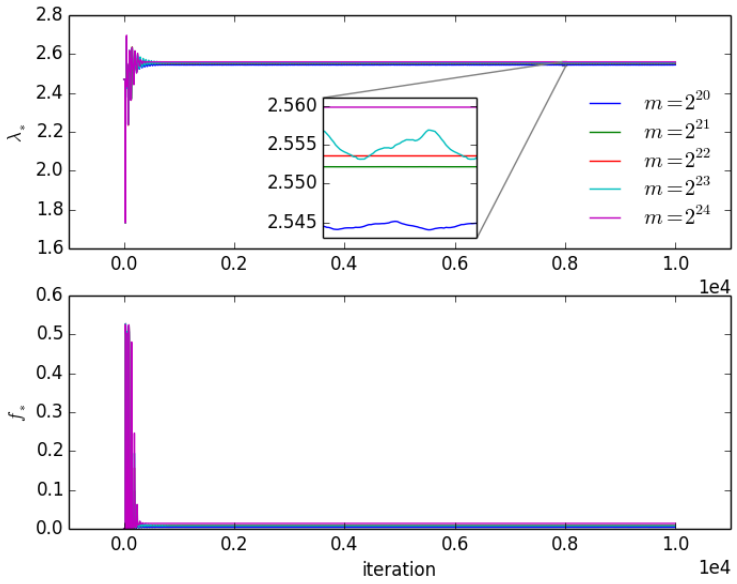

Figure 3: **Top.** Trajectory of the approximation, Λ_t^m, of the partition function for the 50-spin Ising model, for the TbS method with $m = 2^{24}$. The best (highest m) approximation is $\lambda_* \approx 2.545$, a difference of about $5B$ from the value for the 24-spin Ising model. **Bottom.** Corresponding trajectory of F_t^m as computed by the TbS method. The best (highest m) approximation is $f_* \approx 0.014$, a difference of almost $64B$ from the value for the 24-spin model.

with the power method. Using the trajectory averages

$$\bar{\Lambda}_t^m = \frac{1}{t}\sum_{s=1}^{t} \Lambda_t^m \qquad \text{and} \qquad \bar{\Lambda}_t^m = \frac{1}{t}\sum_{s=1}^{t} F_t^m \tag{49}$$

to estimate λ_* and f_* would seem strange had the iterates Λ_t^m and F_t^m been generated by the deterministic power method (we have not reported trajectory averages for the deterministic TbS approach). However, for finite m we do not expect Λ_t^m or F_t^m to converge to λ_* and f_* as t increases. Rather we expect that the distribution of Λ_t^m and F_t^m will converge to some distribution roughly centered around λ_* and f_* respectively. Though in our convergence results we have not addressed the ergodicity of the Markov process V_t^m, one would expect that reasonable functions of V_t^m such as Λ_t^m and F_t^m satisfy a law of large numbers so that, for very large t, the trajectory averages $\bar{\Lambda}_t^m$ and $\bar{F}_t^m$ differ from λ_* and f_* only by a systematic error (i.e., they converge to the limit of the expectations of Λ_t and F_t^m respectively).

6.2. A PDE eigenproblem For given functions $b(x)$ and $\sigma(x)$ with values $\mathbb{R}^n$ and $\mathbb{R}^n \times \mathbb{R}^r$, the backwards Kolmogorov operator

$$Lf = b^{\mathrm{T}} Df + \frac{1}{2}\operatorname{trace}\left(\sigma\sigma^{\mathrm{T}} D^2 f\right) \tag{50}$$

is the generator of the diffusion process

$$dX(t) = b(X(t))\,dt + \sigma(X(t))\,dW(t) \tag{51}$$

where $W(t)$ is an r-dimensional Brownian motion, Df is the vector of first order derivatives of f, and $D^2 f$ is the matrix of its second order derivatives. The operator L governs the evolution of moments of $X(t)$ in the sense that

$$\frac{d}{dt}\mathbf{E}_x\left[f(X(t))\right]\Big|_{t=0} = Lf(x)$$

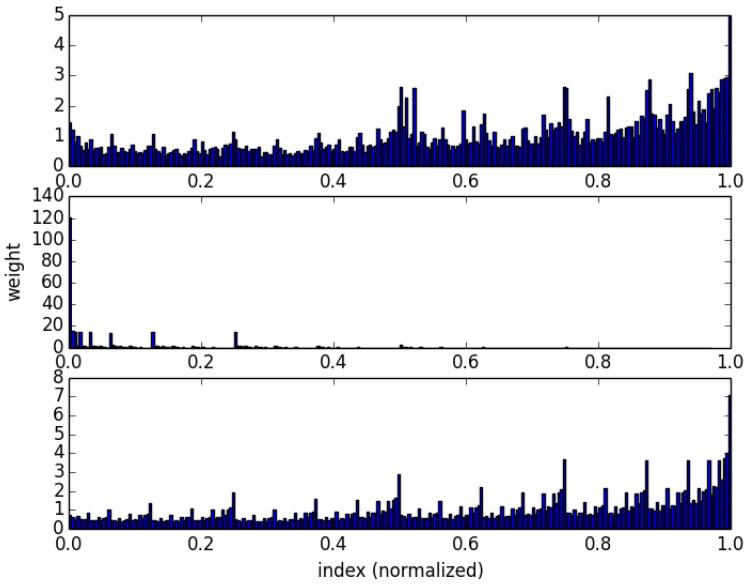

Figure 4: **Top.** Sums of the values of the approximation, V_t^m, of the dominant eigenvector of the 50-spin Ising transfer matrix with $B = 0.01$ and $T = 2.2$, at $t = 10^5$ over 256 intervals of equal size out of the 2^{50} total indices for the FRI method with $m = 2^{24}$. **Middle.** Corresponding sums for the TbS method. **Bottom.** Exact eigenvector for the 20-spin Ising model for qualitative comparison.

(the subscript on the expectation indicates that $X_0 = x$). Note that constant functions are in the kernel of L. The non-trivial eigenfunctions of L all correspond to negative eigenvalues. The magnitude of the greatest of these negative eigenvalues is the spectral gap and characterizes the rate of convergence of expectations such as $\mathbf{E}_x\big[f(X(t))\big]$ to their equilibrium (large t) values.

In this subsection we consider estimation of the largest negative eigenvalue of L for $b = -DV$ with

$$V(x^{(1)}, \ldots, x^{(\ell)}) = \frac{1}{2}\sum_{j=1}^{\ell} \cos(2\pi x_1^{(i)}) \cos(2\pi x_2^{(j)}) + 2\sum_{j=1}^{\ell}\sum_{k=j+1}^{\ell} \cos(\pi(x_1^{(j)} - x_1^{(k)})) \cos(\pi(x_2^{(j)} - x_2^{(k)}))$$

for $x^{(j)} = (x_1^{(j)}, x_2^{(j)}) \in [-1, 1) \times [-1, 1)$. The diffusion coefficient, $\sigma(x)$, is fixed as $\sqrt{2}$. The function V is the potential energy for a periodic system of ℓ, 2D-particles, each subject to both an external force as well as a nonlinear spring coupling the particles together. Equation (51) is a model of the dynamics of that system of particles (in a high friction limit).

The equation $Lg_* = \lambda_* g_*$ is first projected onto a Fourier spectral basis, i.e., we assume that

$$g_*(\vec{x}) = \sum\nolimits_{\vec{\alpha} \in \mathbb{Z}_N^{2\ell}} v(\vec{\alpha})\, e^{i\pi\vec{\alpha}^{\mathrm{H}}\vec{x}}$$

where $\vec{\alpha} = (\alpha^{(1)}, \ldots, \alpha^{(\ell)})$ with $\alpha^{(j)} = (\alpha_1^{(j)}, \alpha_1^{(j)})$, and the symbol $\mathbb{Z}_N^{2\ell}$ is used to indicate that both $\alpha_1^{(j)}$ and $\alpha_1^{(j)}$ are integers with magnitude less than N.

Suppose that $\hat{L}$ is the corresponding spectral projection of L (which, in this case, is real). The matrix $\hat{L}$ can be decomposed into a sum of diagonal (corresponding to the second order term in L) and non-diagonal term (corresponding to first order term in L), i.e.,

$$\hat{L} = A + D.$$

[6] We use the term "confidence intervals" loosely. Our results clearly indicate that m is not so large that systematic errors and deviations from asymptotic normality can be safely ignored.

In this problem we are trying to find not the eigenvalue of $\hat{L}$ of largest magnitude but the largest (they are real) eigenvalue and a transformation is required. As we mentioned in Section 3.2 this can be accomplished by applying power iteration to a matrix obtained from a discrete-in-time approximation of the ODE

$$\frac{d}{dt}y = \hat{L}y$$

i.e., by exponentiating the matrix $t\hat{L}$ for very large t. For example, for sufficiently small $\varepsilon > 0$, the eigenvalue, μ, of largest magnitude, of the matrix

$$K = e^{\frac{1}{2}\varepsilon D}(I + \varepsilon\hat{L})e^{\frac{1}{2}\varepsilon D} \tag{52}$$

is, to within an error of oder ε^2, $1 + \varepsilon\lambda_*$ where λ_* is the eigenvalue of $\hat{L}$ of largest real part (in our case the eigenvalues are real and non-positive). We will apply our iteration schemes to K. By fixing $v_t(\vec{0}) = 0$ we can guarantee that the approximate solutions all have vanishing integral over $[-1, 1)^{2\ell}$, ensuring that the iteration converges to an approximation of the desired eigenvector/value pair (instead of to $v(\vec{0}) = 1$, $v(\vec{\alpha}) = 0$ if $\vec{\alpha} \neq \vec{0}$).

Remark 13. In this problem, rather than estimating the dominant eigenvector of K, our goal is estimate the second largest (in magnitude) eigenvalue of K. Given that we know the largest eigenvalue of K is 1 with an eigenvector that has value one in the component corresponding to $\vec{\alpha} = \vec{0}$ and zeros in all other components, we can therefore exactly orthogonalize the iterates V_t^m with respect to the dominant eigenvalue at each iteration (by fixing $V_t^m(\vec{0}) = 0$). We may view this as using FRI in conjunction with a simple case of orthogonal iteration.

We compare the FRI and TbS approaches with $N = 51$ for the four- and five-particle systems ($\ell = 4, 5$). The corresponding total count of real numbers needed to represent the solution in the five-particle case is $101^{10} \approx 10^{20}$ so only the $\mathcal{O}(1)$ scheme is reasonable. For h we choose a value of 10^{-3}. Our potential V is chosen so that the resulting matrix $\hat{L}$ (and therefore also K) is sparse and its entries are computed by hand. For a more complicated V, the entries of $\hat{L}$ might have to be computed numerically on-the-fly or might not be computable at all. Our ability to efficiently compute the entries of $\hat{L}$ will be strongly effected by the choice of basis. For example, if we use a finite difference approximation of L then the entries of $\hat{L}$ can be computed easily. On the other hand, if the solution is reasonably smooth, the finite difference approximation will converge much more slowly than an approximation (like the spectral approximation) that more directly incorporates properties of the solution (regularity in this case).

Figure 5 plots the trajectory averages over 10^5 iterations for Λ_t^m in the $\ell = 4$ case generated by the FRI method (iteration (16) using Algorithm 1) along with corresponding trajectories of Λ_t^m as generated by the TbS approach (iteration (16) using truncation-by-size). We present results for both methods with $m = 1$, 2, 3, and 4×10^4. Observe that the results from the FRI method appear to have converged on one another while the results generated by the TbS approach show no signs of convergence. The best (highest m) estimate of the eigenvalue generated by FRI is -2.31 and the best estimate generated by TbS is -2.49. Figure 6 plots the trajectory of Λ_t^m in the five particle ($\ell = 5$) case as generated by the FRI method with $m = 10^6$ along with its trajectory average (neglecting the first 500 iterations) of about -1.3. Note that Λ_t^m appears to reach its statistical equilibrium rapidly relative to the 2×10^3 total iterations. Again, the rapid equilibration suggests that statistical error could be removed by averaging over many shorter trajectories evolved in parallel.

6.3. A PDE steady state problem The adjoint L^* of the operator defined in (50) with respect to the standard inner product is called the Fokker–Planck operator. The operator determines the

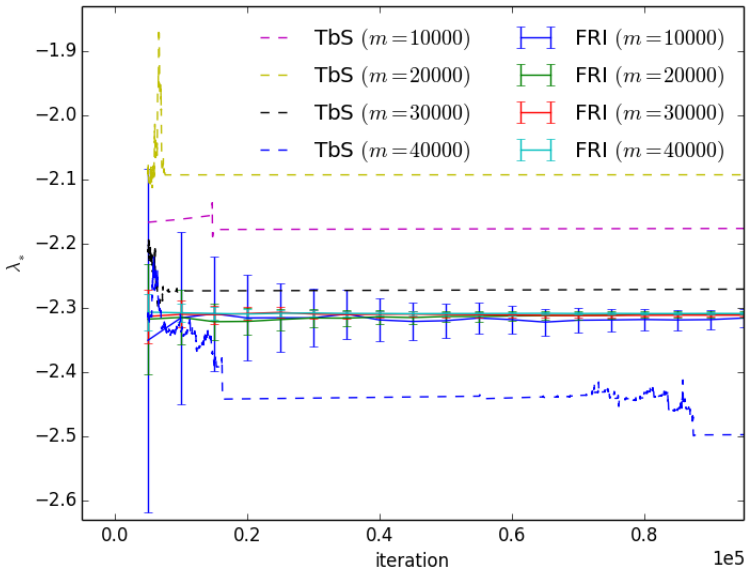

Figure 5: Trajectory averages of the approximation, Λ_t^m, of the largest negative eigenvalue of a backwards Kolmogorov operator for a four 2D-particle (8-dimensional) system, with 95% confidence intervals for the FRI method with $m = 1$, 2, 3, and 4×10^4. The operator is discretized using a Fourier basis with 101 modes per dimension for a total of more than 10^{16} basis elements (half that after taking advantage of the fact that the desired eigenvector is real). The step-size parameter h is set to 10^{-3}. Also on this graph, trajectories of Λ_t^m for the TbS method for the same values of m.

evolution of the density of the process $X(t)$ defined in (51) in the sense that if μ is that density then

$$\partial_t \mu = L^* \mu.$$

An element in the kernel of L^* (a steady state solution of the Fokker–Planck equation) is a density left invariant by $X(t)$.

For the choice of b and σ given in the previous subsection, the steady state solutions are easily seen to be constant multiples of the function

$$\mu_*(\vec{x}) = \frac{e^{-V(\vec{x})}}{\int e^{-V(\vec{x})}}.$$

In most applications, the goal is to compute averages of observables with respect to μ_*. For example, one might hope to find (up to an additive constant) the effective potential (or free-energy) experienced by particle 1,

$$F_1(x^{(1)}) = -\log \int \mu_*(\vec{x})\, dx^{(2)} \cdots dx^{(\ell)}.$$

For that purpose, explicit knowledge of μ_* is of little value since one cannot hope to compute integrals with respect to a function of so many variables (up to 101^8 in our tests). One instead hopes to find a more digestible expression for μ_*. Notice that if a Fourier expansion

$$\mu_*(\vec{x}) = \sum\nolimits_{\vec{\alpha} \in \mathbb{Z}_N^{2\ell}} v(\vec{j})\, e^{i\pi \vec{\alpha}^{\mathrm{H}} \vec{x}}$$

was available then we could compute

$$\mathcal{F}_1(x^{(1)}) = -\log \sum\nolimits_{\alpha^{(1)} \in \mathbb{Z}_N^2} v(\alpha^{(1)}, 0, \ldots, 0)\, e^{i\pi \alpha^{(1)T} x^{(1)}}.$$

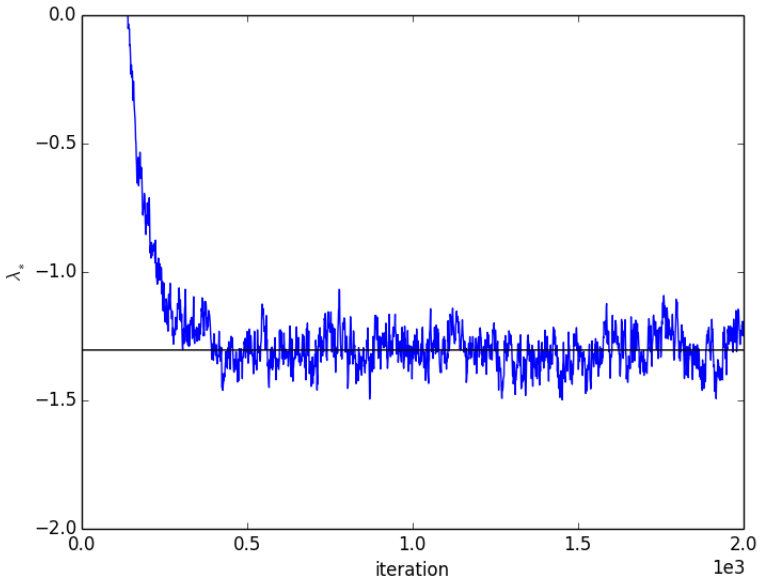

Figure 6: Trajectory of the approximation, Λ_t^m (solid blue line), of the largest negative eigenvalue of a backwards Kolmogorov operator for the five-particle system as computed by the FRI method with $m = 10^6$ over 2×10^3 iterations. The total dimension of the discretized system is more than 10^{20}. The average value of Λ_t^m (ignoring the first 500 iterations) is -1.3 and is shown by a solid black line.

As in the previous section we discretize the Fokker–Planck operator in a Fourier basis resulting in a finite-dimensional linear root finding problem

$$(K - I)\, v_* = 0$$

where K is now defined just as in (52) but with $A = \hat{L}^{\mathrm{H}} + D$. We choose to normalize the solution so that $v(\vec{0}) = 1$ which then results in a linear system

$$\left(\bar{K} - I\right) \bar{v}_* = r$$

where $r(\vec{\alpha}) = -K_{\vec{\alpha}\vec{0}}$, $\bar{K}$ has the row and column corresponding to the index $\vec{0}$ removed, and $\bar{v}_*$ has the component corresponding to index $\vec{0}$ removed.

Remark 14. Note that the linear system $(\bar{K} - I)\bar{v}_* = r$ is solved for v_* here using FRI in conjunction with Jacobi iteration. With the normalization $v_t(\vec{0}) = 1$, this is equivalent to using the power iteration to find the eigenvector corresponding to the largest eigenvalue of K (which is 1). Recalling that here $K \approx I + \varepsilon(\hat{L}^{\mathrm{H}} + D)$, observe that we are (when ε is small) approximately computing $\lim_{t\to\infty} \exp(At)v_0$ with $A = \hat{L}^{\mathrm{H}} + D$, which, since the largest eigenvalue of A is 0, is the desired eigenvector. We repeat that though we know that the dominant eigenvalue of K and we have a formula for μ_*, the dominant eigenvector of L^*, our goal is to compute projections of μ_* that cannot be computed by deterministic means.

In Figure 7 we present approximations of the function F_1 generated by the $\mathcal{O}(1)$ scheme

$$\begin{aligned} V_{t+1}^m &= \Phi_t^m\left(KV_t^m\right), \\ F_{t+1}^m(\alpha^{(1)}) &= \left(KV_t^m\right)(\alpha^{(1)}, 0, \ldots, 0) \\ \overline{F}_{t+1}^m(\alpha^{(1)}) &= (1 - \varepsilon_t)\overline{F}_t(\alpha^{(1)}) + \varepsilon_t F_{t+1}^m(\alpha^{(1)}), \end{aligned}$$

for all $\alpha^{(1)} \in \mathbb{Z}_N^2$ with $\varepsilon_t = (t+1)^{-1}$ and where the independent mappings Φ_t^m are generated according to Algorithm 1. The single particle free-energy[7] as generated by the FRI approach is plotted for $\ell = 2$, 3, 4, and 5 with $m = 10$, 200, 10^4, and 10^6 respectively. In the two, three, and four particle simulations we use 10^5 iterations. Again we choose $N = 51$ and $h = 10^{-3}$. The high cost per iteration in the five particle case restricts our simulations to 2×10^3 iterations. In the four particle case, for which we have validated the FRI solution by simulations with higher values of m ($m = 4 \times 10^4$), the free energy profile produced by the TbS approach differs from the FRI result by as much as 100%. We take slight advantage of the particle exchange symmetry and, at each iteration, replace $(KV_t^m)(\alpha^{(1)}, 0, \ldots, 0)$ in the above equation for F_{t+1}^m by the average of all ℓ components of the form $(KV_t^m)(0, \ldots, \alpha^{(k)}, 0, \ldots, 0)$. Note that in the expansion of μ_*, we know that $v(\vec{\alpha})$ is unchanged when we swap the indices $\alpha^{(j)}$ and $\alpha^{(k)}$ corresponding to any two particles. This fact could be leveraged to greatly reduce the number of basis functions required to accurately represent the solution. We have not exploited this possibility.

Though it is not accurate, the TbS scheme is substantially more stable on this problem. We assume that the relative stability of the TbS scheme is a manifestation of the fact that TbS is not actually representing the high wave number modes that are responsible for stability constraints. Nonetheless, simulating higher dimensional systems would require modifications in our approach. In particular it might be necessary to identify a small number of components of the solution that should always be resolved (never set to zero). For example, for this problem one might choose to resolve some number of basis functions for each particle that are independent of the positions of the other particles.

7 Discussion

We have introduced a family of fast, randomized iteration schemes for eigenproblems, linear systems, and matrix exponentiation. Traditional iterative methods for numerical linear algebra were created in part to deal with instances where the coefficient matrix A (of size $\mathcal{O}(n^2)$) is too big to store but where the operation $x \mapsto Ax$ can nonetheless be carried out. The $\mathcal{O}(1)$ iterative methods in this paper are intended for instances where even the solution vector x (of size $\mathcal{O}(n)$) is too big to store but where the ultimate goal is to compute Fx where F may be dense but has many fewer rows than columns. In addition to the contexts addressed explicitly above, such problems have become increasingly common in other areas as well; for example, in the analysis of large data sets and in the solution of large-scale Lyapunov and Sylvester equations [4, 52, 53] arising in control theory.

A completely deterministic approach to iterative problems related to the methods proposed in this article is the simple thresholding by size (TbS) in which, at each iteration, the smallest entries in the approximation are set to zero. An adaptive version of the TbS approach has recently been advocated for a wide range of applications (see [50, 42, 43]) including variants of those considered in this paper. Like TbS our randomized schemes also rely on the enforcement of sparsity and also tend to set small entries in the approximate solution to zero. While TbS schemes can be effective on some problems with sparse solutions its error, in general, will be strongly dependent on system size.

The core concept behind the schemes is the notion that, by randomly setting entries in vectors to zero, while maintaining a statistical consistency property, we can dramatically reduce the cost and storage of standard iterative schemes. One can view our FRI schemes as an attempt to blur the line separating Markov chain Monte Carlo (MCMC), which is effective in extremely high-dimensional

[7] Note that we only approximate $\mathcal{F}_1$ up to the additive constant $-\log \int e^{-V(\vec{x})}$. In fact, the free energy is typically only defined up to that constant because it is not uniquely specified (one can add a constant to V without changing μ_*.

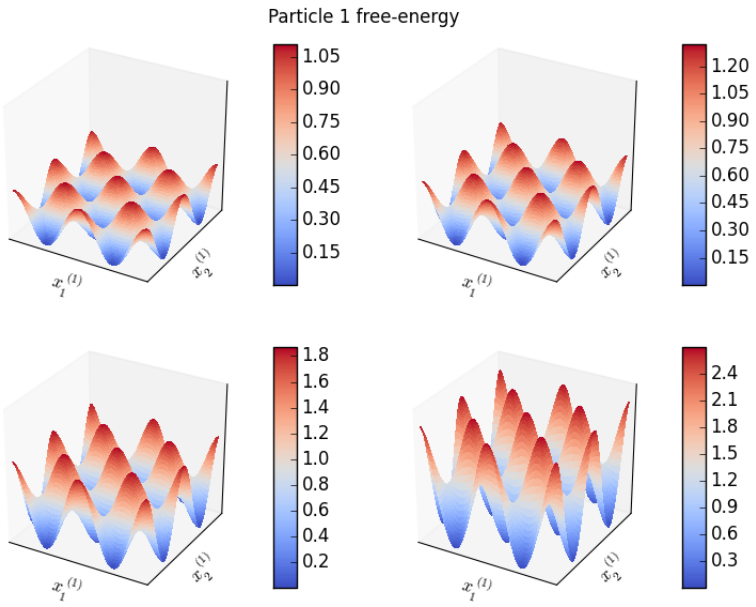

Figure 7: Free energy landscape experienced by a single particle for the (clockwise from top left) two, three, four, and five 2D-particle systems. The surfaces were generated using the FRI method with $m = 10$, 200, 10^4, and 10^6 respectively and $h = 10^{-3}$. The two-, three-, and four-particle simulations were run for 10^5 iterations. The five particle simulation was substantially more expensive per iteration and was run for only 2×10^3 iterations. The number of Fourier modes used to represent the solution in all simulations is 101 per dimension for a total of more than 10^8, 10^{12}, 10^{16}, and 10^{20} basis functions (half that after taking advantage of the fact that the solution is real). As expected, the free energy basins deepen as the number of particles grows. In the four particle case (for which we have high confidence in the estimate produced by FRI) the error from the TbS approach (the results of which we do not plot) is roughly 100% at peaks of the free energy landscape.

settings but is limited to a relatively narrow class of problems and often does not allow the user to take full advantage of known properties of the solution (e.g. smoothness or symmetry properties as in [8]), and traditional deterministic schemes, which are effective on a very general set of relatively low dimensional problems. As for MCMC approaches, if one settles for computing low dimensional projections of the full solution then not every element of the state space need be visited and effective FRI schemes with per iteration cost and storage requirements independent of system size can be derived (as for MCMC the validity of this statement depends on the particular sequence of problems considered). Also as for MCMC we expect that, when deterministic alternatives are available, they will outperform our randomized schemes. For matrices of the size considered in all of our numerical tests, deterministic alternatives are *not* available.

Experience with diffusion Monte Carlo in the context of quantum Monte Carlo simulations suggests that our randomized schemes will be most useful if applied after considerable effort has been expended on finding changes of variables that either make the desired solution as sparse as possible (reducing both bias and variance) or reduce bias by some other means. In many cases this will mean applying our randomized schemes only after one has obtained an estimate of the solution by some deterministic method applied to a reduced dimensional version of the target problem.

Acknowledgments

JQW would like to thank Eric Cances, Tony Lelievre, and Matthias Rousset, for their hospitality and helpful discussions during a visit to ENPC that coincided with the early stages of this work. He would also like to thank Omiros Papaspiliopoulos for many helpful conversations that informed the compression strategies adopted in this article. Both authors would like to thank Alexandre Chorin, Petros Drineas, Risi Kondor, Jianfeng Lu, Omiros Papaspiliopoulos, and Panos Stinis, who made comments that strongly affected this article's structure and content. LHL's work is generously supported by NSF IIS-1546413 and DMS-1209136. JQW's effort on this project was supported by the Advance Scientific Computing Research program within the DOE Office of Science through award de-sc0014205 as well as through a contract from Argonne, a U.S. Department of Energy Office of Science laboratory.

References

[1] V.N. Alexandrov and S. Lakka. Comparison of three monte carlo methods for matrix inversion. In Luc Bougé, Pierre Fraigniaud, Anne Mignotte, and Yves Robert, editors, *Euro-Par'96 Parallel Processing*, volume 1124 of *Lecture Notes in Computer Science*, pages 72–80. Springer Berlin Heidelberg, 1996.

[2] J. Anderson. A random-walk simulation of the Schrödinger equation: H_3^+. *J. Chem. Phys.*, 63(4):1499–1503, 1975.

[3] Roi Baer, Daniel Neuhauser, and Eran Rabani. Self-averaging stochastic kohn-sham density-functional theory. *Phys. Rev. Lett.*, 111:106402, Sep 2013.

[4] P. Benner, J.R. Li, and T. Penzl. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra and Applications*, 15(9):755–777, 2008.

[5] George H. Booth and Ali Alavi. Approaching chemical accuracy using full configuration interaction quantum monte carlo: a study of ionization potentials. *J. Chem. Phys.*, 132:174104, 2010.

[6] George H. Booth, D. Cleland, Alex J. W. Thom, and Ali Alavi. Breaking the carbon dimer: the challenges of multiple bond dissociation with full configuration interaction quantum Monte Carlo methods. *J. Chem. Phys.*, 135:084104, 2011.

[7] George H. Booth, Andreas Grüneis, Georg Kresse, and Ali Alavi. Towards an exact description of electronic wavefunctions in real solids. *Nature*, 493(7432):365–370, 01 2013.

[8] George H. Booth, Alex J. W. Thom, and Ali Alavi. Fermion monte carlo without fixed nodes: A game of life, death, and annihilation in slater determinant space. *The Journal of Chemical Physics*, 131(5), 2009.

[9] Léon Bottou. Stochastic learning. In Olivier Bousquet and Ulrike von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004.

[10] N. Bou-Rabee and E. Vanden-Eijnden. Continuous-time random walks for the numerical solution of stochastic differential equations. arXiv:1502.05034 [math.PR], 2015.

[11] D. Ceperley and B. Alder. Ground state of electron gas by a stochastic method. *Phys. Rev. Lett.*, 45(7):566–569, 1980.

[12] Alexandre Chorin. Random choice solution of hyperbolic systems. *Journal of Computational Physics*, 22:517–536, 1976.

[13] D. Cleland, George H. Booth, and Ali Alavi. Survival of the fittest: accelerating convergence in full configuration-interaction quantum Monte Carlo. *J. Chem. Phys.*, 132:041103, 2010.

[14] D. Cleland, George H. Booth, and Ali Alavi. A study of electron affinities using the initiator approach to full configuration interaction quantum Monte Carlo. *J. Chem. Phys.*, 134:024112, 2011.

[15] E. Coakley, V. Rokhlin, and M. Tygert. A fast randomized algorithm for orthogonal projection. *SIAM Journal on Scientific Computing*, 33(2):849–868, 2011.

[16] N. de Freitas, A. Doucet, and N. Gordon (Eds). *Sequential Monte Carlo Methods in Practice*. Springer, 2005.

[17] Pierre Del Moral. *Feynman-Kac formulae*. Probability and its Applications (New York). Springer-Verlag, New York, 2004. Genealogical and interacting particle systems with applications.

[18] James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[19] P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[20] P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[21] P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006.

[22] Weinan E, Bjorn Engquist, Xiantao Li, Weiqing Ren, and Eric Vanden-Eijnden. The heterognous multiscale methods: A review. *Communications in Computational Physics*, 2(3):367–450, 06 2007.

[23] Sylvester Eriksson-Bique, Mary Solbrig, Michael Stefanelli, Sarah Warkentin, Ralph Abbey, and Ilse C. F. Ipsen. Importance sampling for a Monte Carlo matrix multiplication algorithm, with application to information retrieval. *SIAM J. Sci. Comput.*, 33(4):1689–1706, July 2011.

[24] W. Foulkes, L. Mitas, R. Needs, and G. Rajagopal. Quantum Monte Carlo simulations of solids. *Rev. Mod. Phys.*, 73(1):33–79, 2001.

[25] S. Friedland and L.-H. Lim. The computational complexity of duality. `http://www.stat.uchicago.edu/~lekheng/work/dual.pdf`, 2015.

[26] Norman H. Fuchs. Approximate solutions for large transfer matrix problems. *Journal of Computational Physics*, 83(1):201 – 211, 1989.

[27] J. Goodman and N. Madras. Random-walk interpretations of classical iteration methods. *Linear Algebra Appl.*, 216:61–79, 1995.

[28] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear non-Gaussian Bayesian state estimation. *IEE Proceedings F*, 51:107 – 113, 1993.

[29] R. Grimm and R. Storer. Monte-Carlo solution of Schrödinger's equation. *J. Comp. Phys.*, 7(1):134–156, 1971.

[30] Martin Hairer and Jonathan Weare. Improved diffusion Monte Carlo. *Commun. Pure Appl. Math.*, 67:1995—2021, 2014.

[31] J.M. Hammersley and K.W. Morton. Poor man's Monte Carlo. *J. R. Stat. Soc. B*, 16(1):23–38, 1954.

[32] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, 1989.

[33] M. Kalos. Monte Carlo calculations of the ground state of three- and four-body nuclei. *Phys. Rev.*, 128(4):1791–1795, 1962.

[34] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.

[35] J. Kolorenč and L. Mitas. Applications of quantum Monte Carlo methods in condensed systems. *Rep. Prog. Phys.*, 74:1–28, 2011.

[36] H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*, volume 35 of *Stochastic Modelling and Applied Probability*. Springer, 2nd edition, 2003.

[37] Edo Liberty, Franco Woolfe, Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007.

[38] P. López Ríos, A. Ma, N. D. Drummond, M. D. Towler, and R. J. Needs. Inhomogeneous backflow transformations in quantum monte carlo calculations. *Phys. Rev. E*, 74:066701, Dec 2006.

[39] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30(1):47 – 68, 2011.

[40] Eric Moulines and Francis R. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 451–459. Curran Associates, Inc., 2011.

[41] M. P. Nightingale and H. W. J. Blöte. Gap of the linear spin-1 heisenberg antiferromagnet: A monte carlo calculation. *Phys. Rev. B*, 33:659–661, Jan 1986.

[42] Vidvuds Ozolins, Rongjie Lai, Russel Caflisch, and Stanley Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.

[43] Vidvuds Ozolins, Rongjie Lai, Russel Caflisch, and Stanley Osher. Compressed plane waves yield a compactly supported multiresolution basis for the laplace operator. *Proceedings of the National Academy of Sciences*, 111(5):1691–1696, 2014.

[44] B. Parlett and Wee-Liang Heng. The method of minimal representations in 2D Ising model calculations. *Journal of Computational Physics*, 114:257–264, 1994.

[45] G.A. Pavliotis and A.M. Stuart. *Multiscale Methods: Averaging and Homogenization*, volume 53 of *Texts in Applied Mathematics*. Springer, 2008.

[46] Vladimir Rokhlin, Arthur Szlam, and Mark Tygert. A randomized algorithm for principal component analysis. *SIAM J. Matrix Anal. Appl.*, 31(3):1100–1124, August 2009.

[47] Vladimir Rokhlin and Mark Tygert. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences*, 105(36):13212–13217, 2008.

[48] M.N. Rosenbluth and A.W. Rosenbluth. Monte Carlo calculation of the average extension of molecular chains. *J. Chem. Phys.*, 23(2):356–359, 1955.

[49] Mathias Rousset. On the control of an interacting particle approximation of schrödinger ground states. *SIAM J. Math. Anal*, 38(3):824–844, 2006.

[50] Hayden Schaeffer, Russel Caflisch, Cory D. Hauck, and Stanley Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences*, 110(17):6634–6639, 2013.

[51] J.J. Sheperd, George H. Booth, A. Grüneis, and Ali Alavi. Full configuration interaction perspective on the homogeneous electron gas. *Phys. Rev. B*, 85, 2012.

[52] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.*, 29(3):1268–1288, 2007.

[53] V. Simoncini, D.B. Szyld, and M. Monsalve. On two numerical methods for the solution of large-scale algebraic Riccati equations. *IMA J. Numer. Anal.*, 34(3):904–920, 2014.

[54] G. W. Stewart. *Matrix Algorithms II: Eigensystems*. SIAM, 2001.

[55] Jonathan Weare. A simple example in C++ of FRI applied to computing the dominant eigenvalue of a matrix. `http://dx.doi.org/10.5281/zenodo.31208`, 2015.

[56] Franco Woolfe, Edo Liberty, Vladimir Rokhlin, and Mark Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335 – 366, 2008.