SIAM J. SCI. COMPUT.
Vol. 36, No. 1, pp. A124–A147

CONVERGENCE IN VARIANCE OF CHEBYSHEV ACCELERATED GIBBS SAMPLERS*

COLIN FOX† AND ALBERT PARKER‡

Abstract. A stochastic version of a stationary linear iterative solver may be designed to converge in distribution to a probability distribution with a specified mean μ and covariance matrix A^{-1}. A common example is Gibbs sampling applied to a multivariate Gaussian distribution which is a stochastic version of the Gauss–Seidel linear solver. The iteration operator that acts on the error in mean and covariance in the stochastic iteration is the same iteration operator that acts on the solution error in the linear solver, and thus both the stationary sampler and the stationary solver have the same error polynomial and geometric convergence rate. The polynomial acceleration techniques that are well known in numerical analysis for accelerating the linear solver may also be used to accelerate the stochastic iteration. We derive first-order and second-order Chebyshev polynomial acceleration for the stochastic iteration to accelerate convergence in the mean and covariance by mimicking the derivation for the linear solver. In particular, we show that the error polynomials are identical and hence so are the convergence rates. Thus, optimality of the Chebyshev accelerated solver implies optimality of the Chebyshev accelerated sampler. We give an algorithm for the stochastic version of the second-order Chebyshev accelerated SSOR (symmetric successive overrelaxation) iteration and provide numerical examples of sampling from multivariate Gaussian distributions to confirm that the desired convergence properties are achieved in finite precision.

Key words. Chebyshev polynomial acceleration, Gauss–Seidel, Gibbs sampling, geometric convergence, linear solver, stochastic iteration, SSOR

AMS subject classifications. 65F10, 65B99, 62E17, 60G15, 60G60

DOI. 10.1137/120900940

1. Introduction.

Iterations of the form

$$x^{l+1} = Gx^l + g, \qquad l = 1, 2, \ldots, \tag{1.1}$$

where G is a fixed iteration operator and g is a fixed vector, are commonplace in numerical computation. For example, they occur in the stationary linear iterative methods used to solve systems of linear equations [1, 10, 17, 23]. We often refer to the associated algorithm as a *solver*. We consider these iterations, and also the related stochastic iteration

$$y^{l+1} = Gy^l + g_l, \qquad l = 1, 2, \ldots, \tag{1.2}$$

where now g_l is a "noise" vector given by an independent draw from some fixed probability distribution with finite variance. Just as the deterministic iteration (1.1) can be designed to converge to the solution of a linear system that is too large or complex to solve directly, the stochastic iteration (1.2) may be designed to converge in distribution to a target distribution that is too high dimensional, or complex, to sample from directly. Since the stochastic iteration may be used to generate samples

*Submitted to the journal's Methods and Algorithms for Scientific Computing section December 3, 2012; accepted for publication (in revised form) November 14, 2013; published electronically February 4, 2014. This work was supported by the New Zealand Institute for Mathematics and its Applications thematic programme on PDEs and Marsden contract UOO1015.
http://www.siam.org/journals/sisc/36-1/90094.html
†Department of Physics, University of Otago, Dunedin, New Zealand (fox@physics.otago.ac.nz).
‡Center for Biofilm Engineering and Department of Mathematical Sciences, Montana State University, Bozeman, MT 59715 (parker@math.montana.edu).

from a desired target distribution, we often refer to the associated algorithm as a *sampler*. An example is the conventional Gibbs sampling algorithm [21] applied to sampling from a high-dimensional Gaussian distribution. In that case the iteration operator G is identical to the iteration operator in the Gauss–Seidel iterative method [5, 7].

Novel Gibbs samplers may be designed by considering matrix splittings other than the Gauss–Seidel splitting [5]. Matrix splittings are considered further in section 2. Interestingly, the deterministic and stochastic iterations converge under exactly the same conditions, with a necessary and sufficient condition being that the spectral radius of G be strictly less than 1, that is, $\rho(G) < 1$ [4, 26]. Convergence in both cases is geometric, with the asymptotic average reduction factor given by $\rho(G)$ (though this is called the "convergence rate" in the statistics literature [19]).

A standard method of reducing the asymptotic average reduction factor is by polynomial acceleration, particularly using Chebyshev polynomials [1, 6, 10, 23]. The original formulation used a modified first-order iteration, as above, though the resulting algorithm is impractical due to numerical difficulties [1]. Practical implementations use a nonstationary second-order iteration that can give optimal reduction of error at each iteration.

In this paper, we develop polynomial acceleration for the stochastic iteration. In particular, we develop nonstationary first- and second-order iterations that give optimal convergence in mean and variance to a desired target distribution. Since convergence in mean is achieved by using exactly the linear iteration for solving a linear system, polynomial acceleration of the mean is exactly as in the existing treatments. Hence we focus on optimal convergence in variance that requires modification to the noise term. Correspondingly, we focus throughout the development on sampling from a target distribution that has zero mean and some finite covariance matrix, and hence the noise distribution always has zero mean. Extension to target distributions with nonzero mean is achieved simply by adding the deterministic iteration or, equivalently, adding a fixed vector to the noise term.

We develop the sampling algorithms and demonstrate the equivalence to linear solvers by investigating a sequence of linear iterative solvers, essentially following the historical development in sophistication and speed, and show that exactly the same ideas used to establish properties of the solver can be used to establish the equivalent properties for a sampler. In particular, convergence of the solver implies convergence of the sampler, and the convergence factors are identical, because they are given by the same expression.

We follow the development and derivations of convergence, given in Axelsson [1], for stationary and nonstationary (Chebyshev) first-order and second-order methods, set out in sect. 5.2 (Stationary Iterative Methods) and sect. 5.3 (The Chebyshev Iterative Method). We could have equally followed the excellent presentations of the same methods in Golub and Van Loan [10] or Saad [23]. Our own work and computational implementation actually take a route that switches between the formalism used in these three texts. By following here the route of a single exposition, we hope to show how establishing convergence of the stochastic versions can be made very straightforward.

The most straightforward application of the methods we develop is to sample from a high-dimensional Gaussian distribution, defined by the mean vector μ and covariance matrix A^{-1}. We present an example which shows the convergence of the Chebyshev sampler in finite precision applied to a Gaussian Markov random field (GMRF) with a known sparse precision matrix corresponding to a Matérn-class covariance function

[12, 15]. This example allows efficient numerical calculation since operation by A has reduced numerical cost.

Although we focus on the Gaussian in our numerical example, the accelerated algorithms we give are more generally applicable to any distribution where the focus is on the mean as a "best" estimate and the covariance as a measure of uncertainties, with higher moments not of primary concern. This is typical in inferential methods applied to solving inverse problems or in the growing field of uncertainty quantification, where the mean and variance of the distribution over parameters or predicted quantities are the primary summary statistics of interest.

1.1. Some links between sampling from distributions and solving systems of equations. Consider a probability distribution with probability density function $\pi(x)$ and the two tasks of drawing $x \sim \pi$ (x distributed as π) and computing $x = \arg\max \pi$ (or solving $-\nabla \log \pi = 0$). We use the notation $x_{-i} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ to denote all $n-1$ components of x other than x_i, and $\pi(x_i|x_{-i})$ to denote the univariate conditional distribution over x_i conditioned on the (fixed) value of all other components.

The classical Gibbs sampler or "stochastic relaxation" (also known as Glauber dynamics and the local heat bath algorithm) for generating a sample from π is an iterative algorithm in which one *sweep* consists of updating each component in sequence by drawing from the conditional distribution for the component with all other components fixed at the most recent value, as in Algorithm 1. Repeating this sweep indefinitely produces distributions over iterates that are guaranteed to converge (geometrically) to π under mild conditions [11, ref. 84], [19], though distributions with nonconnected support for which Algorithm 1 fails are easy to find [19].

Algorithm 1: One sweep of the componentwise Gibbs sampler targeting $\pi(x)$

for $i = 1, \ldots, n$ **do**
  sample $z \sim \pi(x_i|x_{-i})$;
  $x_i = z$;
end

It is not hard to see a connection between the Gibbs sampler in Algorithm 1 and the traditional Gauss–Seidel algorithm for maximizing π which consists of repeatedly applying the sweep over componentwise solvers with all other components fixed at the most recent value, as in Algorithm 2: Whereas the Gibbs sampler performs a componentwise conditional sampling, Gauss–Seidel performs componentwise optimization.

Algorithm 2: One sweep of Gauss–Seidel relaxation for maximizing $\pi(x)$

for $i = 1, \ldots, n$ **do**
  set $z = \arg\max_{x_i} \pi(x_i|x_{-i})$;
  $x_i = z$;
end

In statistical physics, distributions often arise with the form

$$\pi(x) = k(\beta)\mathrm{e}^{-\beta H(x)},$$

where $H(x)$ is an energy function (the Hamiltonian), β is inversely proportional to temperature, and k is a normalizing constant. It is often noted that a sampling algorithm may be used to minimize $H(x)$ in the zero temperature limit, i.e., by taking the limit $\beta \to \infty$. Then sampling degenerates to optimization since the distribution is localized at the mode. In particular, Algorithm 1 reduces to Algorithm 2.

In this paper, we exploit an equivalence that operates at finite β to show how the minimizer (or solver) may be adapted to become a sampling algorithm. For example, in the simplest case that $\beta = 1$ and H is quadratic, i.e.,

$$H(x) = \frac{1}{2}x^{\mathsf{T}}Ax - b^{\mathsf{T}}x$$

for some symmetric positive definite (precision matrix) A, π is Gaussian and the Gauss–Seidel minimizer of H becomes the Gibbs sampler for π when coordinatewise minimization is replaced by coordinatewise conditional sampling. One sweep of the Gibbs sampler may be written in the matrix form (1.2) with

$$G = M^{-1}N \quad \text{and} \quad g_l = M^{-1}c_l, \quad \text{where} \quad c_l \overset{\text{iid}}{\sim} N(0, D).$$

Here $M = L + D$ and $N = -L^{\mathsf{T}}$ is a splitting of the (symmetric) precision matrix A in which L is the strictly lower triangular part of A and D is the diagonal of A [11]. This is the same splitting used to write the Gauss–Seidel algorithm for solving $Ax = b$ in matrix form (1.1), with $g = M^{-1}b$. What makes this correspondence important is that the convergence properties of the solver are inherited by the sampler (and vice versa), which means that acceleration techniques developed for the solver may be applied to the sampler. The main purpose of this paper is to establish the equivalence of convergence in mean and covariance in the case of Chebyshev polynomial acceleration, without the assumption of the target distribution being Gaussian.

2. Matrix splitting and iteration operators. Consider the *splitting*

$$A = M - N, \tag{2.1}$$

where A is a symmetric positive definite (SPD) matrix and M is invertible. For example, for the Gauss–Seidel iteration, M is set to the lower triangular part of A (including the diagonal). We will often consider the case where the splitting is *symmetric*, which means that M is symmetric, and hence so is N. We will utilize the family of *iteration operators*

$$G_\tau = \left(I - \tau M^{-1}A\right) \tag{2.2}$$

parameterized by the relaxation parameter $\tau \neq 0$. The natural iteration operator induced by the splitting (2.1) is the case $\tau = 1$, which we denote by G. The nonstationary iterative methods that we consider use a sequence of iteration operators with parameters τ_l, $l = 0, 1, 2, \ldots$, where l denotes iteration number. We will abbreviate G_{τ_l} by G_l, where possible, to avoid subscripts on subscripts.

The iteration operator G_τ may also be thought of as being induced by the splitting

$$A = M_\tau - N_\tau, \tag{2.3}$$

where

$$M_\tau = \frac{1}{\tau}M \quad \text{and} \quad N_\tau = N + \left(\frac{1-\tau}{\tau}\right)M.$$

In the remainder of this section we list some lemmas about iteration operators that we will use. Throughout the rest of the paper, proofs to lemmas and some theorems have been deferred to the appendix.

LEMMA 2.1. *The iteration operators G_τ and G_κ commute, that is, $G_\tau G_\kappa = G_\kappa G_\tau$ for all τ, κ.*

LEMMA 2.2. *For a symmetric splitting, $G_\tau A^{-1}$ is symmetric.*

The following lemma determines the variance of noise terms in sampling algorithms.

LEMMA 2.3. $A^{-1} - G_\tau A^{-1} G_\tau^T = M_\tau^{-1}\left(M_\tau^T + N_\tau\right) M_\tau^{-T}$.

3. First-order iterative methods. We first consider iterative solvers of the equation

$$Ax^* = b,$$

where A is a given SPD matrix, b is a given vector, and the solution we seek is denoted by x^*.

3.1. First-order stationary iterative solver. The first-order stationary iterative solver uses the iteration

$$x^{l+1} = x^l - \tau M^{-1} r^l = G_\tau x^l + g_\tau, \tag{3.1}$$

where $r^l = Ax^l - b$ for $l = 1, 2, \ldots$, the iteration operator G_τ is given by (2.2) and $g_\tau = \tau M^{-1} b$. In the remainder of this section, we derive the fixed point, error polynomial, and average reduction factor for this iteration.

LEMMA 3.1. *The iteration in* (3.1) *has x^* as its unique fixed point, i.e.,*

$$x^* = G_\tau x^* + g_\tau \quad \Leftrightarrow \quad Ax^* = b. \tag{3.2}$$

Define the error at the lth iteration by

$$e^l = x^l - x^*. \tag{3.3}$$

Subtract (3.2) from (3.1) to get the iteration for error

$$e^{l+1} = x^{l+1} - x^* = G_\tau x^l + g_\tau - G_\tau x^* - g_\tau = G_\tau\left(x^l - x^*\right) = G_\tau e^l.$$

By recursion we prove the following theorem.

THEOREM 3.2.

$$e^m = G_\tau^m e^0 = \left(I - \tau M^{-1} A\right)^m e^0 = P_m\left(M^{-1} A\right) e^0,$$

where P_m is the (simple) mth-order polynomial $P_m(\lambda) = (1 - \tau\lambda)^m$.

Note that $P_m(0) = 1$ and $P_m(1/\tau) = 0$. The convergence and convergence rate of the stationary iterative solver follow from Theorem 3.2.

Axelsson [1, p. 176] gives the optimal relaxation parameter

$$\tau_{\text{opt}} = \frac{2}{\lambda_1 + \lambda_n},$$

where $\lambda_1 < \lambda_n$ are the extreme (positive) eigenvalues of $M^{-1}A$, giving the average reduction factor

$$\rho_0 = \frac{1 - \lambda_1/\lambda_n}{1 + \lambda_1/\lambda_n}. \tag{3.4}$$

Note that this implies that the iterative solver (3.1) converges for some value of τ. To be more precise, as long as $M^{-1}A$ has all positive eigenvalues, then $\lambda_1/\lambda_n \in (0, 1)$, which means that $\rho_0 \in (0, 1)$ and the iteration is guaranteed to converge.

3.2. First-order stationary iterative sampler. We will follow the same route to derive a first-order stationary iterative sampler that converges in distribution to a distribution with zero mean and (finite) covariance matrix A^{-1}. Consider the iteration

$$y^{l+1} = G_\tau y^l + g_l \tag{3.5}$$

for $l = 1, 2, \ldots$, where G_τ is the iteration operator defined in (2.2) and now g_l is an independent sample drawn from some density with zero mean, and covariance matrix $\mathrm{Cov}(g_l)$ is chosen so that A^{-1} is the unique invariant covariance of the iteration (3.5). That is, we construct the covariance matrix of g_l, $\mathrm{Cov}\,(g_l)$, so that $\mathrm{Cov}\left(y^l\right) = A^{-1}$ implies that $\mathrm{Cov}\left(y^{l+1}\right) = A^{-1}$. This invariance property is analogous to the fixed point we found for the first-order linear solver in Lemma 3.1.

LEMMA 3.3.

$$\mathrm{Cov}\left(y^{l+1}\right) = G_\tau \,\mathrm{Cov}\left(y^l\right) G_\tau^T + \mathrm{Cov}\,(g_l)\,. \tag{3.6}$$

We require that A^{-1} be the fixed point variance, i.e.,

$$A^{-1} = G_\tau A^{-1} G_\tau^{\mathrm{T}} + \mathrm{Cov}\,(g_l)\,. \tag{3.7}$$

COROLLARY 3.4. $\mathrm{Cov}\,(g_l) = M_\tau^{-1}\left(M_\tau^T + N_\tau\right) M_\tau^{-T}$.

Remark 1.

1. $M_\tau^{-1}\left(M_\tau^{\mathrm{T}} + N_\tau\right) M_\tau^{-\mathrm{T}}$ is always symmetric since A is symmetric.
2. Lemma 2.3 gives an alternative representation for $\mathrm{Cov}(g_l)$.
3. In [5] we use $g_l = M_\tau^{-1} b_l$, where b_l is a random vector with $\mathrm{Cov}\,(b_l) = M_\tau^{\mathrm{T}} + N_\tau$.

Now subtract (3.7) from (3.6) to get the iteration for variance error

$$\mathrm{Cov}\left(y^{l+1}\right) - A^{-1} = G_\tau\left(\mathrm{Cov}\left(y^l\right) - A^{-1}\right) G_\tau^{\mathrm{T}}$$

or

$$\mathcal{E}^{l+1} = G_\tau \mathcal{E}^l G_\tau^{\mathrm{T}},$$

where we have defined the error in variance as $\mathcal{E}^l = \mathrm{Cov}\left(y^l\right) - A^{-1}$ for $l = 0, 1, 2, \ldots$ (cf. (3.3)). By recursion we prove the following theorem (cf. Theorem 3.2).

THEOREM 3.5.

$$\mathcal{E}^m = G_\tau^m \mathcal{E}^0 \left(G_\tau^m\right)^T = P_m\left(M_\tau^{-1} A\right) \mathcal{E}^0 \left(P_m\left(M_\tau^{-1} A\right)\right)^T,$$

where P_m is the (simple) mth-order polynomial $P_m(\lambda) = (1 - \tau\lambda)^m$.

Note that $P_m(0) = 1$ and $P_m(1/\tau) = 0$. The convergence and convergence rate for the variance of the stationary iterative sampler follow from Theorem 3.5. The optimal relaxation parameter and the average reduction factor are the same as for the stationary iterative solver in (3.4).

3.3. First-order nonstationary Chebyshev iterative solver. Equation (3.1) gives a family of iterative methods, parameterized by the relaxation parameter τ, that all have a unique fixed point x^* given by (3.2). A natural idea is to not use a single iteration operator as in the stationary method but to run through a sequence of iteration operators. Perhaps this could give faster convergence. But how does one pick the sequence of iteration operators?

In this section we develop Chebyshev acceleration that makes an optimal choice of iteration operators. The resulting first-order algorithm is impractical due to numerical instability, though it does allow us to establish theoretical convergence results that hold for the second-order iteration developed in following sections.

The first-order nonstationary iterative solver uses the iteration

$$x^{l+1} = x^l - \tau_l M^{-1} r^l = G_l x^l + g_l, \tag{3.8}$$

where $l = 1, 2, \ldots, r^l = Ax^l - b$, $g_l = \tau_l M^{-1} b$, and

$$G_l = \left(I - \tau_l M^{-1} A\right) = M_l^{-1} N_l.$$

The fixed point for this iteration is essentially given by Lemma 3.1.

LEMMA 3.6. *The iteration in* (3.8) *has* x^* *as its unique fixed point, i.e.,*

$$x^* = G_l x^* + g_l \quad \Leftrightarrow \quad Ax^* = b. \tag{3.9}$$

Subtract (3.9) from (3.8) to get the iteration for error

$$e^{l+1} = x^{l+1} - x^* = G_l x^l + g_l - G_l x^* - g_l = G_l \left(x^l - x^*\right) = G_l e^l.$$

By recursion we prove the following theorem.

THEOREM 3.7.

$$e^p = \left(\prod_{l=0}^{p-1} G_l\right) e^0 = \left(\prod_{l=0}^{p-1} \left(I - \tau_l M^{-1} A\right)\right) e^0 = Q_p\left(M_l^{-1} A\right) e^0,$$

where Q_p *is the pth-order polynomial* $Q_p(\lambda) = \left(\prod_{l=0}^{p-1} (1 - \tau_l \lambda)\right)$ *(cf.* [1, *eq.* 5.26]*).*

Note that $Q_p(0) = 1$ and $Q_p(1/\tau_l) = 0$. That is, the relaxation parameters determine the zeros of the polynomial Q_p and hence the τ_l can be chosen to give any desired error polynomial. We may think of Q_p in Theorem 3.7 as representing a general pth-order polynomial, and we now consider how to "best" select the polynomial.

The term $\left\|Q_p\left(M_l^{-1} A\right)\right\|$ may be chosen to have minimum maximum value over the interval $[\lambda_1, \lambda_n]$ (where λ_1 and λ_n are the extreme eigenvalues of $M^{-1}A$) by choosing the specific polynomial

$$Q_p(\lambda) = \frac{T_p\left(\left(\lambda_1 + \lambda_n - 2\lambda\right) / \left(\lambda_1 - \lambda_n\right)\right)}{T_p\left(\left(\lambda_1 + \lambda_n\right) / \left(\lambda_1 - \lambda_n\right)\right)}, \tag{3.10}$$

where T_p is the Chebyshev polynomial of order p. The denominator ensures that $Q_p(0) = 1$. To make this choice we need to know the zeros of Q_p, which are just the zeros of $T_p\left(\left(\lambda_1 + \lambda_n - 2\lambda\right) / \left(\lambda_1 - \lambda_n\right)\right)$. Hence the relaxation parameters are given by (cf. [1, eq. 5.29], [14, Fig. 28.1.1])

$$\frac{1}{\tau_l} = \frac{\lambda_n + \lambda_1}{2} + \frac{\lambda_n - \lambda_1}{2} \cos\left(\pi \frac{2l+1}{2p}\right) \tag{3.11}$$

for $l = 0, 1, 2, \ldots, p-1$. Lemma 2.1 established commutativity of the operators G_l, and hence one can run through the sequence of relaxation parameters in any order.

Does this iteration converge faster than the stationary case? The answer is "yes," in the sense that it will do no worse. An indicative result is given by evaluating the norm of the error at step p in Theorem 3.7 to give

$$\left\|e^p\right\| \leq \max_{\lambda \in [\lambda_1, \lambda_n]} \left|Q_p(\lambda)\right| \left\|e^0\right\|,$$

and since the choice of the scaled Chebyshev minimizes $\max_{\lambda\in[\lambda_1,\lambda_n]}|Q_p(\lambda)|$ over all pth-order polynomials, it seems it will do better than the pth-order polynomial $P_p(\lambda)=(1-\tau\lambda)^p$. This is only an indicative result, as it does not guarantee that $\max_i|Q_p(\lambda_i)|$ is smaller than $\max_i|P_p(\lambda_i)|$. An unequivocal result is given by the explicit calculation [1, eq. 5.30] that for the scaled Chebyshev polynomial (3.10)

$$\max_{\lambda\in[\lambda_1,\lambda_n]}|Q_p(\lambda)|=\frac{1}{T_p\left(\frac{\lambda_1+\lambda_n}{\lambda_1-\lambda_n}\right)}=2\frac{\sigma^p}{1+\sigma^{2p}},$$

where

$$\sigma=\frac{1-\sqrt{\lambda_1/\lambda_n}}{1+\sqrt{\lambda_1/\lambda_n}} \tag{3.12}$$

and hence the asymptotic average reduction factor is σ. In contrast, the *smallest possible* average reduction factor for the unaccelerated stationary linear solver iteration (3.1), or the stochastic iteration (3.5) (over all possible τ), is given by (3.4). Figure 3.1 compares this reduction factor for the stationary case to the the Chebyshev average reduction factor of $(2\frac{\sigma^p}{1+\sigma^{2p}})^{1/p}$ after p iterations as a function of the ratio of eigenvalues $\lambda_1/\lambda_n\in[0,1]$. The two convergence factors are identical for $p=1$, and the figure shows that for $p>1$ and $0<\lambda_1<\lambda_n$ the average reduction factor is always smaller for the Chebyshev nonstationary iteration. Hence we have established the following theorem.

THEOREM 3.8. *Let $A=M_\tau-N_\tau$ be a symmetric splitting with the extreme eigenvalues of $M^{-1}A$ satisfying $0<\lambda_1<\lambda_n$, and let $p>1$ be a fixed integer. If the first-order stationary iterative method converges for some relaxation parameter τ, then the first-order nonstationary Chebyshev iterative method also converges and has a smaller average reduction factor at iteration p.*

3.4. First-order nonstationary Chebyshev iterative sampler. The convergence of the first-order Chebyshev sampler can be established in a straightforward manner, for a symmetric splitting, just as we did for the solver. First we find a fixed point for the covariance matrix, define the error, and then give the asymptotic reduction factor. Just as for the solver, the first-order sampler suffers from numerical instability, but this section lays the theoretical groundwork for convergence results of the second-order sampler introduced later.

The first-order nonstationary iterative sampler uses the iteration

$$y^{l+1}=G_l y^l+g_l$$

for $l=1,2,\ldots$, where $G_l=\left(I-\tau_l M^{-1}A\right)=M_l^{-1}N_l$ and g_l is an independent sample drawn from some density with zero mean, and covariance matrix $\mathrm{Cov}(g_l)=M_l^{-1}\left(M_l^{\mathrm{T}}+N_l\right)M_l^{-\mathrm{T}}$. As for the stationary sampler, the following two lemmas hold (with proofs given by the stationary case that hold for any τ).

LEMMA 3.9.

$$\mathrm{Cov}\left(y^{l+1}\right)=G_l\,\mathrm{Cov}\left(y^l\right)G_l^T+\mathrm{Cov}(g_l). \tag{3.13}$$

LEMMA 3.10. *The unique fixed point variance is A^{-1}, i.e.,*

$$A^{-1}=G_lA^{-1}G_l^T+\mathrm{Cov}(g_l). \tag{3.14}$$

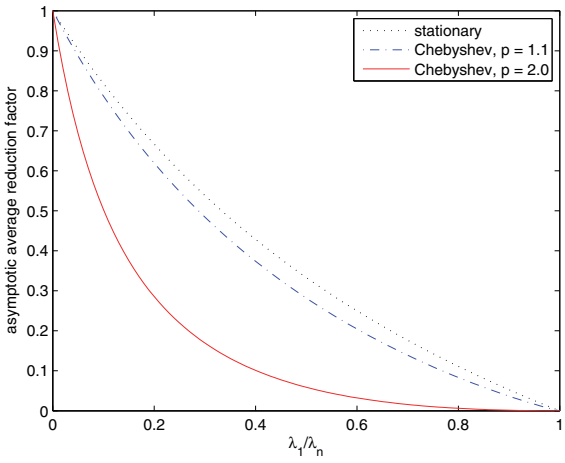

FIG. 3.1. *Comparison of the average reduction factor* $\rho_0 = \frac{1-\lambda_1/\lambda_n}{1+\lambda_1/\lambda_n}$ *for first-order stationary iterations to the the average reduction factor for the Chebyshev accelerated iteration,* $(2\frac{\sigma^p}{1+\sigma^{2p}})^{1/p}$*, with* σ *defined in* (3.12)*. Along the horizontal axis are values of the ratio of the extreme eigenvalues of* $M^{-1}A$*,* $\lambda_1/\lambda_n \in [0,1]$*.*

As before, subtract (3.14) from (3.13) to get the iteration for error in the covariance matrix

$$\mathrm{Cov}\left(y^{l+1}\right) - A^{-1} = G_l\left(\mathrm{Cov}\left(y^l\right) - A^{-1}\right)G_l^{\mathrm{T}}$$

or

$$\mathcal{E}^{l+1} = G_l \mathcal{E}^l G_l^{\mathrm{T}},$$

where $\mathcal{E}^l = \mathrm{Cov}\left(y^l\right) - A^{-1}$ denotes the error in variance for $l = 0, 1, 2, \ldots$. By recursion we prove the following statement.

THEOREM 3.11.

$$\mathcal{E}^m = \left(\prod_{l=0}^{m-1} G_l\right)\mathcal{E}^0\left(\prod_{l=0}^{m-1} G_l\right) = Q_m\left(M_\tau^{-1}A\right)\mathcal{E}^0\left(Q_m\left(M_\tau^{-1}A\right)\right)^T,$$

where Q_m *is the* m*th-order polynomial* $Q_m(\lambda) = \left(\prod_{l=0}^{m-1}(1-\tau_l\lambda)\right)$ *with properties as established for the first-order stationary iterative solver.*

As in the first-order iterative solver, the average reduction factor can be optimized for a given iteration number p by choosing the sequence of relaxation parameters in (3.11). The explicit calculation of the average reduction factor for the first-order iterative solver establishes the following theorem.

THEOREM 3.12. *Let* $A = M_\tau - N_\tau$ *be a symmetric splitting with the extreme eigenvalues of* $M^{-1}A$ *satisfying* $0 < \lambda_1 < \lambda_n$*, and let* $p > 1$ *be a fixed integer. If the splitting converges (as an iterative solver) for some relaxation parameter* τ*, then*

1. *the first-order stationary iterative sampler converges,*
2. *and the first-order nonstationary Chebyshev iterative sampler converges with a smaller average reduction factor than the stationary sampler at iteration* p*.*

4. Second-order methods. Axelsson points out [1, Rem. 5.11] two deficiencies of the first-order Chebyshev iterative method as a solver: First, the number of steps p needs to be selected in advance, with the method not being optimal for any other number of steps. Second, the first-order iteration is numerically unstable, so computer implementation probably will not show the nice theoretical behavior that we have established above. Both these deficiencies also hold for the first-order Chebyshev iterative sampler. The solution for iterative solvers, and hence for iterative samplers, is to develop the second-order methods, which have neither of these deficiencies.

First we establish a few theorems that can be stated with the definitions of first-order iterative operators as previously defined.

4.1. Second-order Chebyshev iterative solver. This section expands (a little) on [1, sect. 5.3.1] to put in place some tools needed for the sampler. We follow Axelsson by considering the splitting with $M = I$, from which the general case follows by considering the (preconditioned) equations with $M^{-1}A$ in place of A.

The second-order iteration is ($M = I$)

$$x^1 = x^0 - \frac{1}{2}\beta_0 r^0 \quad \text{and} \quad x^{l+1} = \alpha_l x^l + (1-\alpha_l)\, x^{l-1} - \beta_l r^l \tag{4.1}$$

for $l = 1, 2, \ldots$ and $r^l = Ax^l - b$.

LEMMA 4.1. *x^* is invariant under the iteration* (4.1)*; i.e., $x^0 = x^*$ implies that $x^l = x^*$ for $l = 1, 2, \ldots$.*

Now subtract the (invariance) expression $x^* = \alpha_l x^* + (1-\alpha_l)\, x^*$ from (4.1) and use $Ae^l = r^l$ to give

$$x^{l+1} - x^* = \alpha_l \left(x^l - x^*\right) + (1-\alpha_l)\left(x^{l-1} - x^*\right) - \beta_l A e^l,$$

which is the second-order iteration for error

$$e^{l+1} = \alpha_l e^l - \beta_l A e^l + (1-\alpha_l)\, e^{l-1}.$$

Assume the inductive hypothesis that

$$e^l = Q_l(A)\, e^0, \tag{4.2}$$

which is true for $l = 0$, and is true for $l = 1$ since $e^1 = e^0 - \frac{1}{2}\beta_0 r^0 = \left(I - \frac{1}{2}\beta_0 A\right) e^0$ (using $Ae^l = r^l$), giving the result with $Q_1(\lambda) = \left(1 - \frac{1}{2}\beta_0\lambda\right)$. The error recursion then gives

$$Q_{l+1}(A)\, e^0 = (\alpha_l I - \beta_l A)\, Q_l(A)\, e^0 + (1-\alpha_l)\, Q_{l-1}(A)\, e^0,$$

and since this is true for any e^0, we find the recursion

$$Q_{l+1}(A) = (\alpha_l I - \beta_l A)\, Q_l(A) + (1-\alpha_l)\, Q_{l-1}(A) \tag{4.3}$$

[1, p. 182]. By choosing the coefficients so that this recursion is the recursion formula for the scaled Chebyshev polynomials, we can ensure that Q_l equals the scaled Chebyshev polynomial in (3.10) which gives optimal error reduction at every step. Axelsson gives this result [1, p. 183], and it is interesting to note that a little good fortune happens; the three equations can be satisfied with just two coefficients because the recursion for the Chebyshev polynomials turns one equation into a second. That is to say, the second-order iteration can be made to fit the Chebyshev polynomials but not necessarily any other set of orthogonal polynomials!

The convergence and reduction factor for the second-order Chebyshev solver are given by the expression (4.2), with the analysis of the first-order Chebyshev solver in section 3.3 giving the result that the second-order Chebyshev method is faster than the stationary method. Unlike the first-order method for which acceleration is guaranteed only at a fixed iteration p, (4.2) also shows that the second-order implementation accelerates for any iteration p. These results are summarized in the following theorem.

THEOREM 4.2. *Let $A = M_\tau - N_\tau$ be a symmetric splitting with the extreme eigenvalues of $M^{-1}A$ satisfying $0 < \lambda_1 < \lambda_n$. If the stationary iterative method converges for some relaxation parameter τ, then the second-order nonstationary Chebyshev iterative method also converges and has a smaller average reduction factor (given by* (3.12)*) for all $p > 1$.*

Furthermore, for any $0 < \varepsilon < 1$, (4.2) shows that to ensure a decrease in error

$$||e^l||_{A^\nu}/||e^0||_{A^\nu} \leq \varepsilon$$

for some real number ν, it suffices to perform

$$p^* = \left\lceil \frac{\ln(\varepsilon/2)}{\ln \sigma} \right\rceil \tag{4.4}$$

iterations of the nonstationary second-order Chebyshev solver [1, eq. 5.32].

4.2. Second-order Chebyshev iterative sampler. Analysis of the second-order nonstationary sampler follows the same route as the first-order nonstationary sampler, with extensions as required in the analysis of the second-order iterative solver. That is, we work out the sequence of variances of iterates, with the noise term chosen so that A^{-1} is the invariant variance. We then subtract the iteration that states the invariance of the variance A^{-1} to get an iteration in the variance error and determine the polynomial in $M^{-1}A$ that acts on errors. An extension is required because the iterates y^l and y^{l+1} are correlated, and so the covariance term needs to be included in the iteration. We do this by writing the second-order iteration as a (block matrix) first-order iteration as Axelsson does [1, sect. 5.2.3] when analyzing the second-order stationary iterative method. There follows a bit of algebra to give the recursion in error polynomial that we got for the second-order iterative solver.

The second-order iterative solver can be written as

$$\begin{aligned} x^{l+1} &= \left(\alpha_l I - \beta_l M^{-1}A\right) x^l + (1-\alpha_l)\, x^{l-1} + \beta_l M^{-1} b \\ &= \alpha_l \left(G_l x^l + g_l\right) + (1-\alpha_l)\, x^{l-1}, \end{aligned} \tag{4.5}$$

where $l = 1, 2, \ldots,\ \beta_l = \alpha_l \tau_l$, and the iterative operator defined by G_l and g_l is the same as the first-order definition in (3.8) with relaxation parameter τ_l. Accordingly, we write the second-order nonstationary iterative sampler as

$$\begin{aligned} y^{l+1} &= \left(\alpha_l I - \beta_l M_l^{-1}A\right) y^l + (1-\alpha_l)\, y^{l-1} + \beta_l g_l \\ &= \alpha_l \left(G_l y^l + g_l\right) + (1-\alpha_l)\, y^{l-1} \end{aligned} \tag{4.6}$$

for $l = 1, 2, \ldots$ with the first step using $y^1 = G_0 y^0 + g_0$ and $\alpha_0 = 1$, and now $\{g_l\}$ are independent samples with $\mathrm{Cov}(g_l)$ chosen so that $\mathrm{Cov}\left(y^0\right) = A^{-1}$ ensures that $\mathrm{Cov}\left(y^l\right) = A^{-1}$ for $l \geq 1$. For the moment we will assume that is done and work out $\mathrm{Cov}\left(g_l\right)$ in section 4.3.

The iteration (4.6) can be written as a first-order iteration in the variables

$$Y^0 = \begin{pmatrix} y^0 \\ 0 \end{pmatrix}, \quad Y^1 = \begin{pmatrix} y^1 \\ y^0 \end{pmatrix}, \quad Y^2 = \begin{pmatrix} y^2 \\ y^1 \end{pmatrix}, \dots$$

with the iteration being

$$Y^{l+1} = \begin{pmatrix} \alpha_l G_l & (1-\alpha_l) I \\ I & 0 \end{pmatrix} Y^l + \alpha_l \begin{pmatrix} g_l \\ 0 \end{pmatrix}. \tag{4.7}$$

Denote

$$\mathcal{G}_l = \begin{pmatrix} \alpha_l G_l & (1-\alpha_l) I \\ I & 0 \end{pmatrix} \quad \text{and} \quad \gamma_l = \alpha_l \begin{pmatrix} g_l \\ 0 \end{pmatrix}.$$

LEMMA 4.3.

$$\begin{aligned} \operatorname{Cov}\left(Y^0\right) &= \begin{pmatrix} \operatorname{Cov}\left(y^0\right) & 0 \\ 0 & 0 \end{pmatrix}, \\ \operatorname{Cov}\left(Y^{l+1}\right) &= \mathcal{G}_l \operatorname{Cov}\left(Y^l\right) \mathcal{G}_l^T + \operatorname{Cov}\left(\gamma_l\right) \\ &= \mathcal{G}_l \operatorname{Cov}\left(Y^l\right) \mathcal{G}_l^T + \left(\alpha_l\right)^2 \begin{pmatrix} \operatorname{Cov}\left(g_l\right) & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned} \tag{4.8}$$

Proof. The lemma follows from the iteration (4.7) and the independence of Y^l and γ_l. ∎

For the second-order iteration we do not have an exact invariance (of variance) as we did in the first-order case in (3.14). This is because the covariance between iterations changes, and hence the off-diagonal blocks K_l and K_l^T of $\operatorname{Cov}\left(Y^l\right)$, cannot be made constant. It turns out this does not matter, and all we need is the sequence of $\operatorname{Cov}\left(Y^l\right)$ that is given by starting the iteration with $\operatorname{Cov}\left(y^0\right) = A^{-1}$ and asserting the requirement that $\operatorname{Cov}\left(y^l\right) = A^{-1}$ for $l = 1, 2, \dots$ with the off-diagonal terms changing. This leads to the sequence of variances denoted by $\operatorname{Cov}(Y^0) = \left(\begin{smallmatrix} A^{-1} & 0 \\ 0 & 0 \end{smallmatrix}\right)$ (so, by definition, $K_0 = 0$) and then $\operatorname{Cov}(Y^l) = \left(\begin{smallmatrix} A^{-1} & K_l \\ K_l^T & A^{-1} \end{smallmatrix}\right)$, where the (1,1) block equals A^{-1} because $\operatorname{Cov}(g_l)$ is chosen to make this true. The sequence of variances then satisfies the recursion given in the following lemma.

LEMMA 4.4. *The requirement that* $\operatorname{Cov}\left(y^0\right) = A^{-1}$ *implies that* $\operatorname{Cov}\left(y^l\right) = A^{-1}$ *for* $l = 1, 2, \dots,$ *and so*

$$\begin{pmatrix} A^{-1} & K_{l+1} \\ K_{l+1}^T & A^{-1} \end{pmatrix} = \mathcal{G}_l \begin{pmatrix} A^{-1} & K_l \\ K_l^T & A^{-1} \end{pmatrix} \mathcal{G}_l^T + \begin{pmatrix} \left(\alpha_l\right)^2 \operatorname{Cov}\left(g_l\right) & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.9}$$

where the (covariance) matrix K_l *satisfies*

$$K_{l+1} = \alpha_l G_l A^{-1} + (1-\alpha_l) K_l^T$$

for $l = 0, 1, 2, \dots,$ *and* $K_0 = 0$.

The recursion for error on variance is then given by subtracting (4.9) from (4.8) to give

$$\operatorname{Cov}\left(Y^{l+1}\right) - \begin{pmatrix} A^{-1} & K_{l+1} \\ K_{l+1}^{\mathrm{T}} & A^{-1} \end{pmatrix} = \mathcal{G}_l \left(\operatorname{Cov}\left(Y^l\right) - \begin{pmatrix} A^{-1} & K_l \\ K_l^{\mathrm{T}} & A^{-1} \end{pmatrix} \right) \mathcal{G}_l^{\mathrm{T}} \tag{4.10}$$

or

$$\mathcal{E}^{l+1} = \mathcal{G}_l \mathcal{E}^l \mathcal{G}_l^{\mathrm{T}}, \quad l = 0, 1, 2, \ldots,$$

where we have defined the error in variance

$$\mathcal{E}^0 = \mathrm{Cov}\left(Y^0\right) - \begin{pmatrix} A^{-1} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \mathrm{Cov}\left(y^0\right) - A^{-1} & 0 \\ 0 & 0 \end{pmatrix},$$
$$\mathcal{E}^l = \mathrm{Cov}\left(Y^l\right) - \begin{pmatrix} A^{-1} & K_l \\ K_l^{\mathrm{T}} & A^{-1} \end{pmatrix}, \quad l = 1, 2, \ldots.$$

Hence, by recurrence, we prove the following theorem.

THEOREM 4.5.

$$\mathcal{E}^m = \left(\prod_{l=0}^{m-1} \mathcal{G}_l\right) \mathcal{E}^0 \left(\prod_{l=0}^{m-1} \mathcal{G}_l\right)^T.$$

Denote the polynomial of the block matrix $\mathcal{H}^{m+1} = \prod_{l=0}^{m} \mathcal{G}_l$ that satisfies

$$\mathcal{H}^1 = \mathcal{G}_0 = \begin{pmatrix} G_0 & 0 \\ I & 0 \end{pmatrix},$$
$$\mathcal{H}^{l+1} = \mathcal{G}_l \mathcal{H}^l = \begin{pmatrix} \alpha_l G_l \mathcal{H}_{11}^l + (1-\alpha_l)\,\mathcal{H}_{21}^l & \alpha_l G_l \mathcal{H}_{12}^l + (1-\alpha_l)\,\mathcal{H}_{22}^l \\ \mathcal{H}_{11}^l & \mathcal{H}_{12}^l \end{pmatrix}.$$

Hence, $\mathcal{H}_{11}^l$ and $\mathcal{H}_{21}^l$ satisfy the recursion

$$\mathcal{H}_{11}^{l+1} = \alpha_l G_l \mathcal{H}_{11}^l + \mathcal{H}_{12}^l \quad \text{and} \quad \mathcal{H}_{21}^{l+1} = \mathcal{H}_{11}^l.$$

Eliminating $\mathcal{H}_{21}^l$ from the first equation, we establish the following statement.

THEOREM 4.6. *The* $(1,1)$ *block of the error polynomial* $\mathcal{H}^l$ *satisfies the recursion*

$$\mathcal{H}_{11}^{l+1} = \alpha_l G_l \mathcal{H}_{11}^l + (1-\alpha_l)\,\mathcal{H}_{11}^{l-1} = \alpha_l \left(I - \tau_l M^{-1} A\right) \mathcal{H}_{11}^l + (1-\alpha_l)\,\mathcal{H}_{11}^{l-1}$$

with $\mathcal{H}_{11}^1 = G_0$.

By setting $\beta_l = \alpha_l \tau_l$ we see that this is the same as the recursion relation (4.3) satisfied by the Q_l that gave the error polynomial for the second-order nonstationary iterative solver. Hence, by matching the coefficients to the terms in the recursion for the Chebyshev polynomials (as for second-order iterative solver), we can ensure that

$$\mathcal{H}_{11}^m(\lambda) = \frac{T_m\left(\left(\lambda_1 + \lambda_n - 2\lambda\right)/\left(\lambda_1 - \lambda_n\right)\right)}{T_m\left(\left(\lambda_1 + \lambda_n\right)/\left(\lambda_1 - \lambda_n\right)\right)} \tag{4.11}$$

as for the other Chebyshev iterative methods. The final step is to show that this is the polynomial that acts on the error in variance of the mth iterate y^m, which is the following theorem.

THEOREM 4.7. *The error in variance at the* m*th iteration is*

$$\mathrm{Cov}\left(y^m\right) - A^{-1} = \mathcal{H}_{11}^m \left(\mathrm{Cov}\left(y^0\right) - A^{-1}\right) \left(\mathcal{H}_{11}^m\right)^T. \tag{4.12}$$

Proof. Read off the $(1,1)$ block in the expansion

$$\mathcal{E}^m = \begin{pmatrix} \mathcal{H}_{11}^m & \mathcal{H}_{12}^{lm} \\ \mathcal{H}_{21}^m & \mathcal{H}_{22}^{lm} \end{pmatrix} \begin{pmatrix} \mathrm{Cov}\left(y^0\right) - A^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathcal{H}_{11}^m & \mathcal{H}_{12}^m \\ \mathcal{H}_{21}^m & \mathcal{H}_{22}^m \end{pmatrix}^{\mathrm{T}}. \qquad \square$$

So now we have all the pieces that establish the following result.

THEOREM 4.8. *Let $A = M - N$ be a symmetric splitting with M invertible. Then the second-order Chebyshev iterative sampler converges, and the variance converges with asymptotic average reduction factor*

$$\sigma^2 = \left(\frac{1 - \sqrt{\lambda_1 / \lambda_n}}{1 + \sqrt{\lambda_1 / \lambda_n}} \right)^2,$$

where $0 < \lambda_1 < \lambda_n$ are the extreme eigenvalues of $M^{-1}A$.

Proof. Each y^l has a zero mean and a variance that differs from A^{-1} according to (4.12). We have constructed the iteration parameters so that $\mathcal{H}_{11}^m$ is the scaled Chebyshev polynomial (4.11) that is bounded on the interval $[\lambda_1, \lambda_n]$ by

$$2\frac{\sigma^p}{1 + \sigma^{2p}},$$

where σ is the average asymptotic reduction factor in (3.12) [1, p. 181]. ∎

Axelsson's result in (4.4) that specifies the required number of iterations to achieve a desired error reduction in the solver suggests that, for any $\varepsilon > 0$, after

$$p^* = \left\lceil \frac{\ln(\varepsilon/2)}{\ln(\sigma^2)} \right\rceil \tag{4.13}$$

iterations, the variance error reduction is smaller than ε.

4.3. Noise variance in the second-order Chebyshev sampler. When establishing Theorem 4.8, we assumed that we knew how to set the variance of the noise term to ensure that A^{-1} was the invariant variance of the stochastic iteration (4.6). To determine $\mathrm{Cov}(g_l)$ it is necessary to explicitly determine the blocks K_l in (4.10). We need the following results.

LEMMA 4.9. *For a symmetric splitting, $K_l = K_l^T$.*

THEOREM 4.10. *For a symmetric splitting,*

$$K_l = G_{\kappa_l} A^{-1} \quad \textit{for} \quad l = 1, 2, \ldots,$$

where the parameter κ_l satisfies $\kappa_1 = \tau_0$ and the recursion

$$\kappa_{l+1} = \alpha_l \tau_l + (1 - \alpha_l)\, \kappa_l \quad \textit{for} \quad l = 1, 2, \ldots.$$

We are now able to derive the noise variance for a symmetric splitting. For $l = 0$, we have (cf. (3.7)) $A^{-1} = G_0 A^{-1} G_0^{\mathrm{T}} + \mathrm{Cov}\,(g_0)$, so

$$\mathrm{Cov}\,(g_0) = \tau_0^2 M^{-1} \left(\left(\frac{2 - \tau_0}{\tau_0} \right) M + N \right) M^{-1}$$

by Lemma 2.3. For $l \geq 1$, we use the expression for the $(1,1)$ block in (4.9) to get

$$\begin{aligned} A^{-1} = {} & \alpha_l^2 G_l A^{-1} G_l^{\mathrm{T}} + \alpha_l\,(1 - \alpha_l)\, G_l K_l + \alpha_l\,(1 - \alpha_l)\, K_l^{\mathrm{T}} G_l^{\mathrm{T}} \\ & + (1 - \alpha_l)^2\, A^{-1} + (\alpha_l)^2\, \mathrm{Cov}\,(g_l)\,, \end{aligned}$$

so $(\alpha_l)^2\, \mathrm{Cov}(g_l) = (1 - (1 - \alpha_l)^2) A^{-1} - \alpha_l^2 G_l A^{-1} G_l^{\mathrm{T}} - \alpha_l (1 - \alpha_l)(G_l K_l + K_l^{\mathrm{T}} G_l^{\mathrm{T}})$. Now using Lemma 4.9 and Theorem 4.10, $\mathrm{Cov}\,(g_l)$ can be rewritten as

$$M^{-1} \left[\tau_l^2 \left(\left(\frac{2 - \tau_l}{\tau_l} \right) M + N \right) + 2\,(1/\alpha_l - 1)\, \tau_l \kappa_l \left(\left(\frac{1}{\tau_l} + \frac{1}{\kappa_l} - 1 \right) M + N \right) \right] M^{-1}. \tag{4.14}$$

Thus, the variance of the noise term is $\mathrm{Cov}(g_l) = M^{-1}(c_l M + d_l N)M^{-1}$ for some real numbers c_l and d_l. We have the following theorem.

THEOREM 4.11. *Let A be SPD and $A = M - N$ be a symmetric splitting. Set the parameters $\{\alpha_l, \tau_l\}$ in the second-order Chebyshev sampler* (4.6) *by*

$$\tau_l = \frac{2}{\lambda_1 + \lambda_n}, \qquad \alpha_l = \beta_l/\tau_l, \qquad \beta_l = \left(\tau_l - \beta_{l-1}\left(\frac{\lambda_n - \lambda_1}{4}\right)^2\right)^{-1},$$

where $\alpha_0 = 1$ and $\beta_0 = \frac{4}{\lambda_n + \lambda_1}$. Let the noise vectors $\{g_l\}$ have

$$E(g_l) = 0 \quad \textit{and} \quad \mathrm{Cov}(g_l) = M^{-1}\left(c_l M + d_l N\right) M^{-1}$$

such that $c_l := \frac{2-\tau_l}{\tau_l} + (d_l - 1)(\frac{1}{\tau_l} + \frac{1}{\kappa_l} - 1)$, $d_l := \frac{2(1-\alpha_l)}{\alpha_l}(\frac{\kappa_l}{\tau_l}) + 1$, $\kappa_{l+1} := \alpha_l\tau_l + (1-\alpha_l)\kappa_l$, and $\kappa_1 = \tau_0$. Then for the Chebyshev samples $\{y^l\}$,

$$E(y^l) \to 0 \textit{ and } \mathrm{Cov}(y^l) \to A^{-1}$$

with asymptotic average reduction factors given by σ (defined in (3.12)*) and σ^2, respectively.*

Proof. The specifications for $\{\alpha_l, \tau_l\}$ are the same as for the second-order Chebyshev accelerated linear solver (4.5) [1, Theorem 5.12], (4.14) confirms the choices for c_l and d_l, and Theorem 4.10 provides κ_l. The reduction factors follow from Theorem 4.8. ∎

4.4. Second-order Chebyshev SSOR sampler. We now have all of the pieces necessary to present a second-order Chebyshev accelerated sampler algorithm. Since a symmetric splitting is required, Chebyshev acceleration in a linear solver is commonly implemented with a symmetric successive overrelaxation (SSOR) splitting $A = M_{\mathrm{SSOR}} - N_{\mathrm{SSOR}}$, with algorithms to be found, for example, in [10, 23]. This splitting depends on the SSOR parameter ω, $0 < \omega < 2$. The choice $\omega = 1$ corresponds to forward and backward sweeps of the Gauss–Seidel stationary solver. Implementations of SSOR for other choices of ω correspond to forward and backward sweeps of successive overrelaxation (SOR); that is, the SSOR splitting is never explicitly calculated. Starting with a Chebyshev SSOR solver, Theorem 4.11 shows how to construct a Chebyshev accelerated SSOR sampler that generates random vectors from any distribution with first moments that converge to zero and second moments that converge to A^{-1}. The simplest such sampler is from a multivariate Gaussian since a Gaussian is specified only by its mean and covariance matrix [5]. We present a Chebyshev accelerated Gibbs sampler from a Gaussian as Algorithm 3.

For arbitrary non-Gaussian distributions, Algorithm 3 still generates samples with the correct first and second moments, but the higher moments will be incorrect. One could conceivably apply Chebyshev acceleration to the higher moments as well, but we do not pursue that here.

The estimates of the extreme eigenvalues λ_1 and λ_n of $M_{\mathrm{SSOR}}^{-1}A$ required by Algorithm 3 can be found inexpensively using a conjugate gradient (CG) algorithm [16]. In addition to generating the eigenvalue estimates $\hat{\lambda_1}$ and $\hat{\lambda_n}$, one can also use CG to generate an approximate sample to $N(0, A^{-1})$ [18]. We investigate the effect of seeding Algorithm 3 with a CG sample elsewhere. In practice, the convergence of Chebyshev solvers and samplers is maintained, with modified reduction factors, as long as $\lambda_1 < \hat{\lambda}_1 < \hat{\lambda}_n < \lambda_n$ [23, p. 383]. The CG-Lanczos estimates $\hat{\lambda_1}$ and $\hat{\lambda_n}$ satisfy this requirement [2, p. 61], [16, p. 18].

Algorithm 3: Chebyshev accelerated SSOR sampler from $N(0, A^{-1})$

input : SSOR parameter $\omega : 0 < \omega < 2$; SOR splitting $A = M_\omega - N_\omega$; extreme eigenvalues $0 < \lambda_1 < \lambda_n$ of $M_{\text{SSOR}}^{-1}A$; initial state y^0; maximum iteration $l_{\max}$

output: $y^{l_{\max}+1}$ approximately distributed as $N(0, A^{-1})$

Set $D_\omega^{1/2} = \left(\left(\frac{2}{\omega} - 1\right)\text{diag}(A)\right)^{1/2}$, $\delta = \left(\frac{\lambda_n - \lambda_1}{4}\right)^2$, $\tau = \frac{2}{\lambda_n + \lambda_1}$;

$\beta = 2\tau$;
$\alpha = 1$;
$d = \frac{2}{\alpha} - 1$;
$c = \left(\frac{2}{\tau} - 1\right) d$;
$\kappa = \tau$;
for $l = 0, \ldots, l_{\max}$ **do**
  sample $z \sim N(0, I)$;
  $b = d^{1/2} D_\omega^{1/2} z$;
  $x = y^l + M_\omega^{-1}(b - Ay^l)$;
  sample $z \sim N(0, I)$;
  $b = c^{1/2} D_\omega^{1/2} z$;
  $w = x - y^l + M_\omega^{-T}(b - Ax)$;
  if $l = 0$ **then**
    $y^{l+1} = \alpha(y^l + \tau w)$;
  else
    $y^{l+1} = \alpha(y^l - y^{l-1} + \tau w) + y^{l-1}$;
  end
  $\beta = (1/\tau - \beta\delta)^{-1}$;
  $\alpha = \beta/\tau$;
  $d = 2\kappa(1 - \alpha)/\beta + 1$;
  $c = \left(\frac{2}{\tau} - 1\right) + (d - 1)\left(1/\tau + 1/\kappa - 1\right)$;
  $\kappa = \beta + (1 - \alpha)\kappa$;
end

Analogous to the SSOR solver algorithms in [10, 23], the Chebyshev sampler implements sequential forward and backward sweeps of an SOR sampler [7, 20] (i.e., the SSOR splitting is never calculated, and the SOR splitting is explicitly used). The feasibility of drawing a noise vector g_l with the correct variance for other splittings $A = M - N$ depends on how easy it is to solve the system $Mu = r$ for some vector u given a residual vector r (solvers must deal with this same issue), but it also depends on how easy it is to factor $c_l M + d_l N$ [5]. A simplifying aspect of using SSOR is that this factorization need never be explicitly computed.

By Theorem 4.11, the Chebyshev SSOR samples generated by Algorithm 3 have a mean which converges to zero as fast as the Chebyshev linear solver converges to $A^{-1}b$ (i.e., with the asymptotic average reduction factor σ); and the covariance matrix of the samples converges to A^{-1} with asymptotic average reduction factor σ^2.

5. Numerical examples sampling from Gaussians at different resolutions. The development we have given of polynomial accelerated samplers requires that the mean and inverse covariance of the target distribution be known, or at least

that operations required within the splitting of the precision matrix may be performed. The simplest such case is where the target distribution is a multivariate Gaussian distribution with a specified mean vector and precision matrix. We now give an example of accelerated sampling from a GMRF in three dimensions that has a sparse precision matrix defined via a partial differential equation (PDE) and boundary conditions.

Our example uses the relationship between stationary GMRFs and stochastic PDEs that was noted by Whittle [25] for the Matérn (or Whittle–Matérn; see [12]) class of covariance functions and that was also exploited in [3, 15]. Rather than stating the PDE, we find it more convenient to work with the equivalent variational form, in this case

$$\mathcal{Q}(\phi) = \int_\Omega \left(\frac{R}{4} |\nabla \phi|^2 + \frac{1}{4R} \phi^2 \right) dx + \int_{\partial\Omega} \frac{\phi^2}{2} ds,$$

which has Euler–Lagrange equations being the Helmholtz operator with (local) Robin boundary conditions $\phi + R\frac{\partial \phi}{\partial n} = 0$ on $\partial\Omega$. In our example we apply this operator twice, which can be thought of as squaring the Helmholtz operator. We compute with a finite-dimensional FEM (finite element method) discretization. When the resulting discrete quadratic form is $\bar{\mathcal{Q}}(\bar{\phi}) = \bar{\phi}^T H \bar{\phi}$, where $\bar{\phi}$ is a vector of nodal values and H is the Hessian, the resulting GMRF has density

$$\pi(\bar{\phi}) \propto \exp\left\{ -\bar{\phi}^T H^2 \bar{\phi} \right\}.$$

We chose this operator because the precision matrix is sparse, while the covariance function (after suitable scaling) is close to the Matérn-class covariance $\exp\{-r/R\}$ with length-scale R. Note that it is not quite the case that we have available a square root of the precision matrix H^2, as the notation suggests, since we have omitted the linear function-to-element operator for brevity of exposition.

The following examples both use a cubic-element discretization of the cubic domain $[0,1]^3$, with trilinear interpolation from nodal values within each element. The examples also both use $R = 1/4$, though they differ in the number of nodes (or elements) in each coordinate direction.

5.1. A $5 \times 5 \times 5$ example ($n = 125$). We first present a small example for which we were able to monitor convergence of the iterates generated by the second-order Chebyshev accelerated SSOR sampler (Algorithm 3 with $\omega = 1$). Convergence was assessed by the relative error $||A^{-1} - S^l||_2 / ||A^{-1}||_2$, where the precision matrix A is the square of the Helmholtz operator described above, and $S^l \approx \mathrm{Cov}(y^l)$ is the empirical covariance matrix calculated over 10^3 sampler runs. Figure 5.1 illustrates the convergence of the Chebyshev sampler iterates and compares the results to a Gibbs SSOR sampler and to sampling via Cholesky factorization. The initial state imputed into both the Chebyshev and SSOR Gibbs samplers for all runs was $y^0 = 0$. In addition, the Chebyshev sampler requires estimates of the extreme eigenvalues of $M_{\mathrm{SSOR}}^{-1} A$ for the SSOR splitting $A = M_{\mathrm{SSOR}} - N_{\mathrm{SSOR}}$. A preconditioned CG algorithm with preconditioner M_{SSOR} provided $\hat{\lambda}_1 = 1.268 \times 10^{-3}$ and $\hat{\lambda}_n = 0.9999$. Each of the 10^3 chains, for both the Chebyshev and Gibbs samplers, ran to iteration $l = 500$.

The benchmarks for the iterative samplers in finite precision are the results generated by sampling via the Cholesky factorization of A. The relative error in covariance estimation using 10^3 Cholesky samples was $4.99/||A^{-1}||_2 = 0.0525$ and is depicted by the green horizontal line in Figure 5.1. Using this criterion, the second-order Chebyshev accelerated sampler converged in finite precision after 60 iterations.

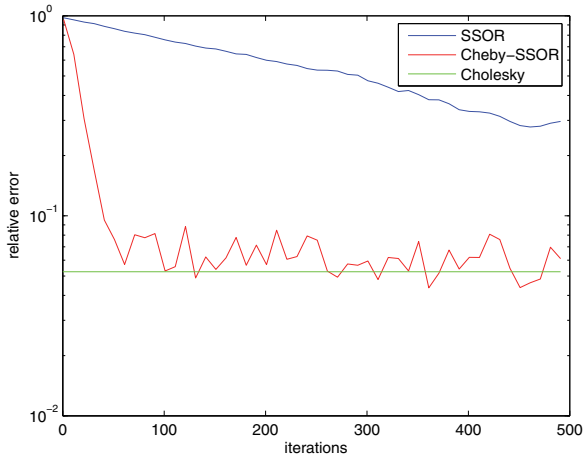

FIG. 5.1. *Iterative SSOR sampler convergence to* $N(0, A^{-1})$ *for a* 125×125 *precision matrix* A. *The vertical axis is the relative error* $||A^{-1} - S^l||_2/||A^{-1}||_2$. *On the horizontal axis is the number of iterations. Both SSOR and Chebyshev accelerated SSOR were implemented with relaxation parameter* $\omega = 1$.

The average asymptotic reduction factor was $\sigma = 0.9312$ for the Chebyshev linear solver and also for the first moment of a Chebyshev sampler (calculated via (3.12)). This suggests that $p^* = 269$ iterations are required to reduce the linear solver error to $\varepsilon = 10^{-8}$ (see (4.4)). In fact, when solving $Ax = b$ for a randomly generated b, the Chebyshev SSOR solver reduced the 2-norm of the residual to 10^{-8} after 296 iterations. The average reduction factor was $\sigma^2 = 0.8671$ for the error in covariance (Theorems 4.8 and 4.11), and so the number of iterations required by the linear solver is an upper bound of the number of iterations required for the mean and variance of the sampler to converge. Furthermore, (4.13) suggests that the error in the Chebyshev covariance should be reduced to a fraction of about $\varepsilon = 10^{-4}$ of the original error after about 70 iterations. For the unaccelerated SSOR sampler, the asymptotic reduction factors for the mean and covariance are $\rho(G) \approx 1 - \hat{\lambda}_1 = 0.9987$ and $\rho(G)^2 = 0.9974$, respectively. This suggests that a Gibbs SSOR sampler must perform about 3500 iterations to attain the same error reduction in covariance (since $(1 - \hat{\lambda}_1)^{2 \cdot 3500} \approx 10^{-4}$). Since $y^0 = 0$, convergence in mean is not shown.

5.2. A $30 \times 30 \times 30$ example ($n = 27{,}000$). This example illustrates the feasibility of Chebyshev accelerated sampling for large problems for which sampling by a Cholesky factorization of the precision matrix is computationally and memory intensive and hence not possible on a standard laptop or desktop computer. A problem like this on a three-dimensional domain is not amenable to bandwidth reducing permutations which sometimes can reduce the computational and memory requirements of the Cholesky factorization. Figure 5.2 shows a Chebyshev sample after $l = 5n$ iterations with a precision matrix A that is the square of the Helmholtz operator described above. The initial state for this run was $y^0 = 0$, and the extreme eigenvalues of $M_{\text{SSOR}}^{-1}A$ were estimated to be $\hat{\lambda}_1 = 1.366 \times 10^{-6}$ and $\hat{\lambda}_n = 1 - 1.56 \times 10^{-8}$ using a preconditioned CG algorithm. Thus, the average asymptotic reduction factor was $\sigma = 0.9977$ for the error in the first moment (calculated via (3.12)) and was $\sigma^2 = 0.9953$ for the error in covariance. After $5n = 1.35 \times 10^5$ iterations, (4.13) suggests

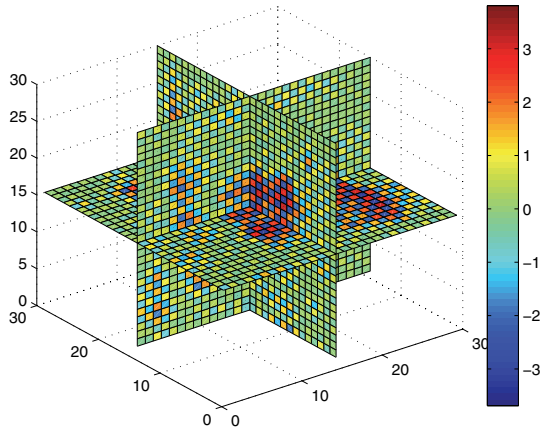

FIG. 5.2. *A draw from an* $n = (2.7 \times 10^4)$*-dimensional Gaussian using a Chebyshev accelerated SSOR sampler after* $5n$ *iterations.*

that, in infinite precision, the error in the Chebyshev covariance should be reduced to a fraction of about $\varepsilon = 10^{-50}$ of the original error. Contrast this with the reduction factors $\rho(G) \approx 1 - \hat{\lambda}_1 = 1 - 1.366 \times 10^{-6}$ and $\rho(G)^2 = 1 - 2.73 \times 10^{-6}$ for the Gibbs SSOR sampler. These average reduction factors suggest that after running the Gibbs sampler $5n$ iterations, the covariance error will be reduced to $\rho(G)^{2 \cdot 5n} \approx 0.69$ of the original error, with $250n = 6.75 \times 10^6$ iterations required for a 10^{-8} reduction.

6. Discussion. We have shown how matrix splitting (of the precision matrix), which is the standard route to understanding linear iterative solvers, is also useful for constructing stochastic iterations that converge to a distribution having a desired covariance matrix. Equivalence of convergence properties then allowed us to develop polynomial acceleration of the sampling algorithm to accelerate convergence in mean and covariance. Accordingly, we see that the value of this work lies in accelerating distributional convergence in those settings where the mean and covariance are statistics of primary interest.

We established the connection between stationary linear iterative solvers (1.1) and samplers (1.2) in section 1.1 by considering componentwise sweep Gibbs sampling from a Gaussian distribution with known mean μ and covariance A^{-1} and observed that the sampler derives from the same splitting that gives the Gauss–Seidel solver. When the noise term in (1.2) is Gaussian and when the initial state y^0 is also Gaussian, each of the l-step distributions (over y^l) is Gaussian, and so the limiting (target) distribution is necessarily Gaussian. However, when the noise terms are not Gaussian the limit is not Gaussian, and so the equivalence holds more generally than just for Gaussian distributions.

We are only able to offer some intuition on what defines the broader class of distributions that are potentially targeted by iterations of form (1.2). As we mentioned, convergence in distribution of iterates in (1.2) occurs iff $\rho(G) < 1$, i.e., G is a contraction. Since the addition of the independent random variable g_l has the effect of convolving the distribution over Gy^l by the noise distribution, the distributional effect of each iteration is to contract and then smear out through convolution. This

procedure seems well suited to convergence to (a subset of) unimodal distributions, though it seems unlikely to us that strongly multimodal distributions can be targeted under this procedure. However, strongly multimodal distributions are not usefully summarized by a mean and covariance and therefore are not the target of this paper.

Equivalence of convergence properties in the stationary case means that polynomial acceleration of linear solvers may then be applied to accelerate convergence in the mean and covariance of the stochastic iteration. Since the mean term in the stochastic iteration is exactly the deterministic iteration used as a linear solver, our contribution is to show how polynomial acceleration may be applied to the covariance matrix in the stochastic iteration. In principle, the analysis we have given can be extended to also design the noise distribution to correctly accelerate convergence to the third, and higher, moments of the target distribution. However, we have not pursued that analysis, as it is more difficult and of unclear worth.

The analysis we have given requires that the (global) mean and precision matrix of the target distribution be known in advance, or at least that the matrix vector operations required in the iteration may be performed. That is most commonly the case when the target distribution is Gaussian, as in the numerical examples in section 5. The recent advent of adaptive Monte Carlo methods [13, 20] does offer the possibility of adapting to the mean and covariance within the iteration, as in the adaptive Metropolis (AM) algorithm. We have implemented such an algorithm and found positive results in cases we have tried, but we have no convergence theory for the resulting algorithm.

One of the motivations for undertaking this work was to understand the relationship between *stochastic relaxation*, as Geman and Geman labeled Gibbs sampling [8], and (classical) *relaxation*, which is the term Southwell used for early stationary iterative solvers [24]. In particular, we were curious whether these two relaxations were related in a formal mathematical sense or just a colloquial sense. As we have shown, the two are mathematically equivalent in the setting of sampling from Gaussian distributions in that the iteration operator, error polynomial, and convergence rates are identical. This provides a formal basis for adapting more efficient solving algorithms to produce more efficient sampling algorithms. This correspondence has been noted before, e.g., by Goodman and Sokal [11], who applied the (classical) multigrid algorithm to Gibbs sampling.

We have a wider intention in writing this paper, which is to attract the numerical analysis community into developing sampling algorithms. By presenting the sampling algorithms in the language that is familiar to numerical analysts, we hope that we have shown how natural, even obvious, the application of polynomial acceleration to Gibbs sampling of normal distributions is. It may therefore come as a surprise to some that this is a very recent result (due to the authors) and that this paper presents the first ever analysis of convergence for first- and second-order Chebyshev accelerated sampling. By finding an equivalence between sampling algorithms and computational linear algebra we have revealed something about the current state of technology used in sampling, which indicates that the state of sophistication of sampling algorithms is presently akin to the state of linear solvers in the 1960's and that potentially great advances can be made in sampling by applying well-developed ideas from computational linear algebra and optimization.

It would be remiss, however, to leave the impression that computational methods for sampling are in need of advances because those developing them are less capable than those developing computational linear algebra. In general, establishing the convergence of sampling algorithms is a more delicate issue than establishing

the convergence of an optimization algorithm since the entire path taken must be considered if a sampler is to have the desired ergodic properties. Furthermore, convergence occurs in the space of distributions, not the space of the state vector, meaning that even calculating residuals is not directly feasible. Typically one must resort to sample-based estimates which are computationally expensive and subject to errors. Nevertheless, sophisticated ideas that have been hard-earned by the computational science community can constructively be applied to sampling, as we hope this paper demonstrates. For example, Chebyshev polynomial accelerated samplers are guaranteed to have a smaller average reduction factor than their unaccelerated stationary counterparts. Furthermore, equivalence of convergence factors means that the convergence rate of the accelerated sampler may be estimated by numerically estimating the convergence rate of the accelerated linear solver, rather than resorting to time consuming sample-based estimates using many runs of the sampler.

While performing this research we have recognized the debt we owe to (the late) Gene Golub, who pioneered first- and second-order Chebyshev acceleration for linear solvers [9], which we have built upon. We are pleased to demonstrate the connection between Gene's work and the sampling algorithms from statistics by publishing in this journal that Gene had wanted to remain titled the Journal on Scientific *and Statistical* Computing [22].

7. Appendix. This appendix contains lemmas that are not directly used in the main body of the paper and also proofs to lemmas and theorems in the paper.

The following lemma writes the iteration operator in (2.2) directly in terms of the splitting in (2.3).

LEMMA 7.1. $G_\tau = M_\tau^{-1} N_\tau = \tau G + (1-\tau)\, I$.

Proof. Substitute the splitting (2.3) into the definition for G_τ. ∎

LEMMA 2.1. *The iteration operators G_τ and G_κ commute, that is, $G_\tau G_\kappa = G_\kappa G_\tau$ for all τ, κ.*

Proof. Expand using (2.2) to give $G_\tau G_\kappa = I - (\tau+\kappa)\, M^{-1}A + \tau\kappa M^{-1}AM^{-1}A = G_\kappa G_\tau$. ∎

LEMMA 2.2. *For a symmetric splitting, $G_\tau A^{-1}$ is symmetric.*

Proof. $G_\tau A^{-1} = \left(I - \tau M^{-1}A\right) A^{-1} = A^{-1} - \tau M^{-1}$ is symmetric. ∎

The following lemmas are needed when we come to calculate the variance of noise terms used in sampling algorithms.

LEMMA 7.2. $A^{-1} - G_\tau A^{-1} G_\kappa^T = \tau\kappa M^{-1}\left(\frac{1}{\tau}M + \frac{1}{\kappa}M^T - A\right) M^{-T}$.

Proof. $A^{-1} - G_\tau A^{-1} G_\kappa^{\mathrm{T}} = M_\tau^{-1}\left(M_\tau A^{-1} M_\kappa^{\mathrm{T}}\right) M_\kappa^{-\mathrm{T}} - M_\tau^{-1} N_\tau A^{-1}\left(M_\kappa^{-1}N_\kappa\right)^{\mathrm{T}} = M_\tau^{-1}\left(M_\tau + M_\kappa^{\mathrm{T}} - A\right) M_\kappa^{-\mathrm{T}}$. Then use $M_\tau = M/\tau$, etc. ∎

LEMMA 2.3. $A^{-1} - G_\tau A^{-1} G_\tau^T = M_\tau^{-1}\left(M_\tau^T + N_\tau\right) M_\tau^{-T}$.

Proof. Set $N_\tau = M_\tau - A$ in the proof to Lemma 7.2. ∎

LEMMA 7.3. *For a symmetric splitting,*

$$A^{-1} - G_\tau A^{-1} G_\kappa^T = \tau\kappa M^{-1}\left(\left(\frac{1}{\tau} + \frac{1}{\kappa} - 1\right) M + N\right) M^{-1}.$$

Proof. Substitute $M^{\mathrm{T}} = M$ and $A = M - N$ into Lemma 7.2. ∎

LEMMA 7.4. *For a symmetric splitting,*

$$A^{-1} - G_\tau A^{-1} G_\tau^T = M_\tau^{-1}\left(M_\tau + N_\tau\right) M_\tau^{-1}.$$

Proof. Put $\kappa = \tau$ in Lemma 7.3, and use $M_\tau^{\mathrm{T}} = M_\tau$ in Lemma 2.3. ∎

LEMMA 3.1. *The iteration in* (3.1) *has* x^* *as its unique fixed point, i.e.,*

$$x^* = G_\tau x^* + g_\tau \quad \Leftrightarrow \quad Ax^* = b.$$

Proof. If $Ax^* = b$, then $G_\tau x^* + g_\tau = \left(I - \tau M^{-1}A\right) x^* + \tau M^{-1}b = x^* - \tau M^{-1}Ax^* + \tau M^{-1}b = x^*$, so x^* is a fixed point of the iteration. Conversely, any fixed point x^* satisfies $Ax^* = b$ and is unique by invertibility of A. ∎

LEMMA 3.3. $\mathrm{Cov}\left(y^{l+1}\right) = G_\tau \,\mathrm{Cov}\left(y^l\right) G_\tau^T + \mathrm{Cov}\left(g_l\right)$.

Proof. It follows since $y^l$and g_l are independent. ∎

COROLLARY 3.4. $\mathrm{Cov}\left(g_l\right) = M_\tau^{-1}\left(M_\tau^T + N_\tau\right) M_\tau^{-T}$.

Proof. The expression for the fixed point variance is $A^{-1} = G_\tau A^{-1} G_\tau^{\mathrm{T}} + \mathrm{Cov}\left(g_l\right)$, and hence $\mathrm{Cov}\left(g_l\right) = A^{-1} - G_\tau A^{-1} G_\tau^{\mathrm{T}}$, with the result following from Lemma 2.3. ∎

LEMMA 4.1. x^* *is invariant under the iteration* (4.1)*; i.e.,* $x^0 = x^*$ *implies that* $x^l = x^*$ *for* $l = 1, 2, \ldots$.

Proof. If $x^0 = x^*$, then $r^0 = 0$, in which case $x^1 = x^*$. If $x^l = x^{l-1} = x^*$, then $r^l = 0$, so $x^{l+1} = \alpha_l x^* + (1 - \alpha_l) x^* = x^*$. ∎

LEMMA 4.4. *The requirement that* $\mathrm{Cov}\left(y^0\right) = A^{-1}$ *implies that* $\mathrm{Cov}\left(y^l\right) = A^{-1}$ *for* $l = 1, 2, \ldots,$ *and so*

$$\begin{pmatrix} A^{-1} & K_{l+1} \\ K_{l+1}^T & A^{-1} \end{pmatrix} = \mathcal{G}_l \begin{pmatrix} A^{-1} & K_l \\ K_l^T & A^{-1} \end{pmatrix} \mathcal{G}_l^T + \begin{pmatrix} (\alpha_l)^2 \,\mathrm{Cov}\left(g_l\right) & 0 \\ 0 & 0 \end{pmatrix},$$

where the (covariance) matrix K_l *satisfies* $K_{l+1} = \alpha_l G_l A^{-1} + (1 - \alpha_l) K_l^T$ *for* $l = 0, 1, 2, \ldots,$ *and* $K_0 = 0$.

Proof. By definition, $K_0 = 0$. For $l = 0$, we have $\alpha_0 = 1$, and the result may be checked directly by setting $\mathrm{Cov}\left(y^0\right) = A^{-1}$, which gives $\mathrm{Cov}(Y^0) = \left(\begin{smallmatrix} A^{-1} & 0 \\ 0 & 0 \end{smallmatrix}\right)$, and hence Lemma 4.3 gives

$$\mathrm{Cov}\left(Y^1\right) = \mathcal{G}_0 \begin{pmatrix} A^{-1} & 0 \\ 0 & 0 \end{pmatrix} \mathcal{G}_0^{\mathrm{T}} + \begin{pmatrix} \mathrm{Cov}\left(g_0\right) & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} A^{-1} & G_0 A^{-1} \\ A^{-1} G_0^{\mathrm{T}} & A^{-1} \end{pmatrix},$$

which shows that $K_1 = G_0 A^{-1}$. To complete the proof for $l = 1, 2, \ldots,$ the recursion relation follows by expansion of the iteration (4.9) and writing the recurrence for the off-diagonal blocks

$$\begin{aligned} \begin{pmatrix} A^{-1} & K_{l+1} \\ K_{l+1}^{\mathrm{T}} & A^{-1} \end{pmatrix} &= \begin{pmatrix} \alpha_l G_l & (1 - \alpha_l) I \\ I & 0 \end{pmatrix} \begin{pmatrix} A^{-1} & K_l \\ K_l^{\mathrm{T}} & A^{-1} \end{pmatrix} \begin{pmatrix} \alpha_l G_l^{\mathrm{T}} & I \\ (1 - \alpha_l) I & 0 \end{pmatrix} \\ &\quad + \begin{pmatrix} (\alpha_l)^2 \,\mathrm{Cov}\left(g_l\right) & 0 \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} A^{-1} & \alpha_l G_l A^{-1} + (1 - \alpha_l) K_l^{\mathrm{T}} \\ \alpha_l A^{-1} G_l^{\mathrm{T}} + (1 - \alpha_l) K_l & A^{-1} \end{pmatrix} \\ &\quad + \begin{pmatrix} (\alpha_l)^2 \,\mathrm{Cov}\left(g_l\right) & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$ ∎

LEMMA 4.9. *For a symmetric splitting,* $K_l = K_l^T$.

Proof. The result is true for $l = 0$, trivially. The result follows by induction since the expression $K_{l+1} = \alpha_l G_l A^{-1} + (1 - \alpha_l) K_l^{\mathrm{T}}$ for $l = 0, 1, 2, \ldots$ shows that K_{l+1} is symmetric when K_l is symmetric because $G_l A^{-1}$ is symmetric by Lemma 2.2. ∎

THEOREM 4.10. *For a symmetric splitting,* $K_l = G_{\kappa_l} A^{-1}$ *for* $l = 1, 2, \ldots,$ *where the parameter* κ_l *satisfies* $\kappa_1 = \tau_0$ *and* $\kappa_{l+1} = \alpha_l \tau_l + (1 - \alpha_l)\, \kappa_l$ *for* $l = 1, 2, \ldots$.

Proof. Since $K_1 = G_0 A^{-1}$, the expansion holds for $l = 1$ with $\kappa_1 = \tau_0$. Assuming the result holds for l, then by Lemmas 4.4 and 4.9, the recursion in K_l gives

$$\begin{aligned} K_{l+1} &= \alpha_l G_l A^{-1} + (1 - \alpha_l)\, K_l \\ &= \alpha_l \left(I - \tau_l M^{-1} A\right) A^{-1} + (1 - \alpha_l) \left(I - \kappa_l M^{-1} A\right) A^{-1} \\ &= \left(I - \left[\alpha_l \tau_l + (1 - \alpha_l)\, \kappa_l\right] M^{-1} A\right) A^{-1}, \end{aligned}$$

so the expansion and recursion hold for $l + 1$, and hence the result follows by induction. ∎

REFERENCES

[1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1996.

[2] J. BRANDTS AND H. VAN DER VORST, *The convergence of Krylov methods and Ritz values*, in Conjugate Gradient Algorithms and Finite Element Methods, M. Krizek, P. Neittaanmaki, R. Glowinski, and S. Korotov, eds., Springer, New York, 2004, pp. 47–68.

[3] T. CUI, C. FOX, AND M. J. O'SULLIVAN, *Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm*, Water Resources Research, 47 (2011).

[4] M. DUFLO, *Random Iterative Models*, Springer-Verlag, New York, 1997.

[5] C. FOX AND A. PARKER, *Gibbs Sampling of Normal Distributions Using Matrix Splittings and Polynomial Acceleration*, manuscript.

[6] L. FOX AND I. PARKER, *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, Oxford, UK, 1968.

[7] A. GALLI AND H. GAO, *Rate of convergence of the Gibbs sampler in the Gaussian case*, Math. Geol., 33 (2001), pp. 653–677.

[8] S. GEMAN AND D. GEMAN, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Mach. Intell., 6 (1984), pp. 721–741.

[9] G. GOLUB AND R. VARGA, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods*, Numer. Math., 3 (1961), pp. 147–156.

[10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[11] J. GOODMAN AND A. D. SOKAL, *Multigrid Monte Carlo method: Conceptual foundations*, Phys. Rev. D (3), 40 (1989), pp. 2035–2071.

[12] P. GUTTORP AND T. GNEITING, *On the Whittle-Matérn Correlation Family*, NRCSE Technical Report Series 80, University of Washington, Seattle, WA, 2005.

[13] H. HAARIO, E. SAKSMAN, AND J. TAMMINEN, *An adaptive Metropolis algorithm*, Bernoulli, 7 (2001), pp. 223–242.

[14] R. W. HAMMING, *Numerical Methods for Scientists and Engineers*, McGraw–Hill, New York, 1962.

[15] F. LINDGREN, H. RUE, AND J. LINDSTRÖM, *An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach*, J. R. Stat. Soc. Ser. B Stat. Methodol., 73 (2011), pp. 423–498.

[16] G. MEURANT, *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*, SIAM, Philadelphia, 2006.

[17] O. NEVANLINNA, *Convergence of Iterations for Linear Equations*, Birkhäuser, Basel, 1993.

[18] A. PARKER AND C. FOX, *Sampling Gaussian distributions in Krylov spaces with conjugate gradients*, SIAM J. Sci. Comput., 34 (2012), pp. B312–B334.

[19] C. ROBERT AND G. CASELLA, *Monte Carlo Statistical Methods*, 2nd ed., Springer, New York, 2004.

[20] G. O. ROBERTS AND J. ROSENTHAL, *Optimal scaling for various Metropolis-Hastings algorithms*, Statist. Sci., 16 (2001), pp. 351–367.

[21] G. O. ROBERTS AND S. SAHU, *Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler*, J. Roy. Statist. Soc. Ser. B, 59 (1997), pp. 291–317.

[22] R. D. Russell, *private communication*, 2011.
[23] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[24] R. Southwell, *Relaxation Methods in Theoretical Physics*, Clarendon Press, Oxford, UK, 1946.
[25] P. Whittle, *On stationary processes in the plane*, Biometrika, 41 (1954), pp. 434–449.
[26] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.