CONVERGENCE IN VARIANCE OF FIRST-ORDER AND SECOND-ORDER 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 $\mu$ 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 Chebyshev-accelerated SSOR (symmetric successive over-relaxation) 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, geometric convergence, sampler, linear solver, stochastic iteration, SSOR

1. Introduction. Iterations of the form

$$x^{l+1} = Gx^l + g$$ \hspace{1cm} l = 1, 2, \ldots, \hspace{1cm} (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, 9, 16, 22]. We often refer to the associated algorithm as a solver. We consider these iterations, and also the related stochastic iteration

$$y^{l+1} = Gy^l + g_l$$ \hspace{1cm} l = 1, 2, \ldots, \hspace{1cm} (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 from a desired target distribution, we often refer to the associated algorithm as a sampler. An example is the conventional Gibbs sampling algorithm [20] 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 [4, 6].

Novel Gibbs samplers may be designed by considering matrix splittings other than the Gauss-Seidel splitting [4]. Matrix splittings are considered further in section 2. Interestingly, the deterministic and stochastic iterations converge under exactly

*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).
the same conditions, with a necessary and sufficient condition being that the spectral radius of $G$ is strictly less than 1, that is $\rho(G) < 1$ [3, 25]. 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 [18]).

A standard method of reducing the asymptotic average reduction factor is by polynomial acceleration, particularly using Chebyshev polynomials [1, 5, 9, 22]. The original formulation used a modified first-order iteration, as above, though the resulting algorithm has numerical difficulties [1]. Typical modern implementations use a non-stationary 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 non-stationary 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 non-zero mean is simply achieved by adding the deterministic iteration, or, equivalently, adding a fixed vector to the noise term. The covariance of the target distribution is denoted by $A^{-1}$ which is a symmetric positive definite matrix. One can think of the target distribution being $N(0, A^{-1})$. In spatial models with a Markov structure, the precision matrix $A$ is often sparse, so efficient algorithms are usually written in terms of operation by $A$ that has reduced numerical cost.

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 non-stationary (Chebyshev) first-order and second-order methods, set out in §5.2 (Stationary Iterative Methods) and §5.3 (The Chebyshev Iterative Method). We could equally have followed the excellent presentations of the same methods in Golub and van Loan [9], or Saad [22]. Our own work, and computational implementation, actually takes 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.

We have mentioned the application of these methods to sampling from high-dimensional Gaussian distributions, which are defined by the mean vector $\mu$ and covariance matrix $A^{-1}$. The polynomial acceleration we present correctly samples from that Gaussian distribution. However, 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. We present an example which shows the convergence of the Chebyshev sampler in finite precision applied to a Gaussian Markov random field with a known sparse precision matrix corresponding to a Matérn-class covariance function [11, 14].

2. Matrix splitting and iteration operators. Consider the splitting

$$A = M - N$$

(2.1)

where \(A\) is a symmetric positive definite (SPD) matrix and \(M\) is invertible. 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

$$G_\tau = (I - \tau M^{-1} A)$$

(2.2)

parameterized by the relaxation parameter \(\tau \neq 0\). The natural iteration operator induced by the splitting (2.1) is the case \(\tau = 1\), that we denote by \(G_1\). Non-stationary iterative methods use a sequence of iteration operators with parameters \(\tau_l\), \(l = 0, 1, 2, \ldots\), where \(l\) denotes iteration number. We will abbreviate \(G_\tau\) by \(G_l\), where possible, to avoid subscripts on subscripts.

The iteration operator \(G_\tau\) may also be thought of as being induced by the splitting

$$A = M_\tau - N_\tau$$

(2.3)

where

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

The following lemma writes the iteration operator directly in terms of the splitting.

**Lemma 2.1.** \(G_\tau = M_\tau^{-1} N_\tau = \tau G - (1 - \tau) I\).

**Proof.** Substitute the splitting (2.3) into the definition for \(G_\tau\).

In the remainder of this section we list some simple Lemmas about iteration operators that we will need later when we develop second-order non-stationary samplers. Throughout the rest of the manuscript, longer proofs to lemmas and theorems have been deferred to the Appendix.

**Lemma 2.2.** The iteration operators \(G_\tau\) and \(G_\kappa\) commute, that is \(G_\tau G_\kappa = G_\kappa G_\tau\), for all \(\tau, \kappa\).

**Proof.** Substitute the splitting (2.3) into the definition for \(G_\tau\).

**Lemma 2.3.** For a symmetric splitting \(G_\tau A^{-1}\) is symmetric.

**Proof.** Substitute in \(M_T = M\) and \(A = M - N\) into Lemma 2.4.

**Lemma 2.4.** \(A^{-1} - G_\tau A^{-1} G_\kappa^T = \tau \kappa M^{-1} \left( \frac{1}{\tau} M + \frac{1}{\kappa} M^T - A \right) M^{-T}\).

**Lemma 2.5.** \(A^{-1} - G_\tau A^{-1} G_\tau^T = M_\tau^{-1} (M_\tau^T + N_\tau) M_\tau^{-T}\).

**Lemma 2.6.** For a symmetric splitting

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

**Proof.** Substitute in \(M_T = M\) and \(A = M - N\) into Lemma 2.4.
Lemma 2.7. For a symmetric splitting
\[ A^{-1} - G_\tau A^{-1} G_\tau^T = M_\tau^{-1} (M_\tau + N_\tau) M_\tau^{-1}. \]

Proof. Put \( \kappa = \tau \) in Lemma 2.6 and use \( M_\tau^T = M_\tau \) in Lemma 2.5. \( \square \)

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 \( x^* \).

3.1. First-order stationary iterative solver. The first-order stationary iterative solver uses the iteration
\[ x^{l+1} = x^l - \tau M^{-1} r^l = G_\tau x^l + g_\tau \]  
where \( r^l = Ax^l - b \) for \( l = 1, 2, \ldots \), the iteration operator \( G_\tau \) 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.
\[ x^* = G_\tau x^* + g_\tau \iff Ax^* = b. \]  

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

Define the error at the \( l \)th iteration by
\[ e^l = x^l - x^*. \]  

Lemma 3.2. \( A e^l = r^l \).
Proof. \( A (x^l - x^*) = Ax^l - b. \) \( \square \)

Subtract (3.2) from (3.1) to get the iteration for error
\[ e^{l+1} = x^{l+1} - x^* = G_\tau x^l + g_\tau - G_\tau x^* - g_\tau = G_\tau (x^l - x^*) = G_\tau e^l. \]

By recursion we prove the following theorem:

Theorem 3.3.
\[ e^m = G_\tau^m e^0 = (I - \tau M^{-1} A)^m e^0 = P_m (M^{-1} A) e^0 \]
where \( P_m \) is the (simple) \( m \)th-order polynomial
\[ P_m (\lambda) = (1 - \tau \lambda)^m. \]

Note that \( P_m (0) = 1 \), and also that \( P_m (1/\tau) = 0 \). Convergence and convergence rate of the stationary iterative solver follow from Theorem 3.3.

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
\[
\rho_0 = \frac{1 - \lambda_1 / \lambda_n}{1 + \lambda_1 / \lambda_n}.
\]
(3.4)

Note that this implies that the iterative solver converges for some value of \( \tau \). 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
\[
y^{l+1} = G_\tau y^l + g_l,
\]
(3.5)
for \( l = 1, 2, \ldots \), where \( G_\tau \) 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 \( \text{Cov}(g_l) \) chosen so that \( A^{-1} \) is the unique invariant covariance of the iteration (3.5).

That is, we construct the covariance matrix of \( g_l \). \( \text{Cov} (g_0) \), so that \( \text{Cov} (y^l) = A^{-1} \)
implies that \( \text{Cov} (y^{l+1}) = A^{-1} \). This invariance property is analogous to the fixed point we found for the first-order linear solver in Lemma 3.1.

**Lemma 3.4.**
\[
\text{Cov} (y^{l+1}) = G_\tau \text{Cov} (y^l) G_\tau^T + \text{Cov} (g_l).
\]
(3.6)

**Proof.** Follows since \( y^l \) and \( g_l \) are independent. \( \Box \)

**Corollary 3.5.** \( \text{Cov} (g_l) = M_\tau^{-1} (M_\tau^T + N_\tau) M_\tau^{-T} \).

**Proof.** The expression for the fixed point variance is
\[
A^{-1} = G_\tau A^{-1} G_\tau^T + \text{Cov} (g_l),
\]
(3.7)
hence \( \text{Cov} (g_l) = A^{-1} - G_\tau A^{-1} G_\tau^T \), with the result following from Lemma 2.5. \( \Box \)

**Remark 1.**
1. \( M_\tau^{-1} (M_\tau^T + N_\tau) M_\tau^{-T} \) is always symmetric since \( A \) is symmetric.
2. Lemma 2.7 gives an alternative representation for \( \text{Cov}(g_l) \).
3. In [4] we use \( g_l = M_\tau^{-1} b_l \) where \( b_l \) is a random vector with \( \text{Cov}(b_l) = M_\tau^T + N_\tau \).

Now subtract (3.7) from (3.6) to get the iteration for variance error
\[
\text{Cov} (y^{l+1}) - A^{-1} = G_\tau \left( \text{Cov} (y^l) - A^{-1} \right) G_\tau^T
\]
or
\[
\mathcal{E}^{l+1} = G_\tau \mathcal{E}^l G_\tau^T
\]
where we have defined the error in variance as \( \mathcal{E}^l = \text{Cov} (y^l) - A^{-1} \) for \( l = 0, 1, 2, \ldots \) (cf. (3.3)). By recursion we prove the following theorem (cf. Theorem 3.3):

**Theorem 3.6.**
\[
\mathcal{E}^m = G_\tau^m \mathcal{E}^0 (G_\tau^m)^T = P_m (M_\tau^{-1} A) \mathcal{E}^0 (P_m (M_\tau^{-1} A))^T
\]
where \( P_m \) is the (simple) \( m \)th-order polynomial \( P_m (\lambda) = (1 - \tau \lambda)^m \).

Note that \( P_m (0) = 1 \) and also that \( \frac{1}{\tau} P_m (1/\tau) = 0 \). Convergence and the convergence rate for the variance of the stationary iterative sampler follow from Theorem 3.6.

The optimal relaxation parameter and the average reduction factor are the same as for stationary iterative solver in (3.4).

5
3.3. First-order non-stationary Chebyshev iterative solver. Equation (3.1) gives a family of iterative methods, parameterized by the relaxation parameter $\tau$, that all have a unique fixed point $x^*$ given by (3.2). So the obvious 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?

The first-order non-stationary iterative solver uses the iteration
\[ x^{l+1} = x^l - \tau M^{-1} r^l = G_l x^l + g_l \] (3.8)
where $l = 1, 2, \ldots$, $r^l = Ax^l - b$, $g_l = \tau M^{-1} b$, and
\[ G_l = (I - \tau M^{-1} A) = M_l^{-1} N_l. \]

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

**Lemma 3.7.** The iteration in (3.8) has $x^*$ as it’s unique fixed point, i.e.
\[ x^* = G_l x^* + g_l \iff Ax^* = b. \] (3.9)

Subtract (3.9) from (3.8) to get the iteration for error
\[ e^{l+1} = x^{l+1} - x^* = G_l x^l + g_l - G_l x^* - g_l = G_l (x^l - x^*) = G_l e^l. \]

By recursion we prove the following theorem:

**Theorem 3.8.**
\[ e^p = \left( \prod_{l=0}^{p-1} G_l \right) e^0 = \left( \prod_{l=0}^{p-1} (I - \tau_l M^{-1} A) \right) e^0 = Q_p (M^{-1}_l A) e^0 \]
where $Q_p$ is the $p^{th}$-order polynomial $Q_p (\lambda) = \left( \prod_{l=0}^{p-1} (1 - \tau_l \lambda) \right)$ (cf. [1, eqn 5.26]).

Note that $Q_p (0) = 1$ and also that $Q_p (1/\tau_l) = 0$. That is, the relaxation parameters determine the zeros of the polynomial $Q_p$ and hence the $\tau_l$ can be chosen to give any desired error polynomial. We may think of $Q_p$ in Theorem 3.8 as representing a general $p^{th}$-order polynomial, and we now consider how to ‘best’ select the polynomial.

The term $\|Q_p (M^{-1}_l A)\|$ may be chosen to have minimum maximum value over the interval $[\lambda_1, \lambda_n]$ by choosing the specific polynomial
\[ Q_p (\lambda) = \frac{T_p ((\lambda_1 + \lambda_n - 2\lambda) / (\lambda_1 - \lambda_n))}{T_p ((\lambda_1 + \lambda_n) (\lambda_1 - \lambda_n))} \] (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 $T_p ((\lambda_1 + \lambda_n - 2\lambda) / (\lambda_1 - \lambda_n))$. Hence the relaxation parameters are given by (cf. [1, eqn 5.29], [13, Fig 28.1.1])
\[ \frac{1}{\tau_l} = \frac{\lambda_n + \lambda_1}{2} + \frac{\lambda_n - \lambda_1}{2} \cos \left( \frac{\pi (2l + 1)}{2p} \right) \] (3.11)
for $l = 0, 1, 2, \ldots, p - 1$. Lemma 2.2 established commutativity of the operators $G_l$, 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.’ An indicative result is given by evaluating the norm of the error at step \( p \) in Theorem 3.8 to give

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

and since the choice of the scaled Chebyshev minimizes \( \max_{\lambda \in [\lambda_1, \lambda_n]} |Q_p(\lambda)| \) over all \( p \)th-order polynomials, it seems it will do better than the \( p \)th-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 \([1, \text{eqn 5.30}]\) that for the scaled Chebyshev polynomial (3.10)

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

where

\[
\sigma = \frac{1 - \sqrt{\lambda_1 / \lambda_n}}{1 + \sqrt{\lambda_1 / \lambda_n}}
\]

and hence the asymptotic average reduction factor is \( \sigma \) (for \( p \) large). In contrast, the smallest possible average reduction factor for the un-accelerated stationary linear solver iteration (3.1), or the stochastic iteration (3.5), (over all possible \( \tau \)) is given by (3.4). Figure 3.1 compares this reduction factor for the stationary case to the Chebyshev average reduction factor of \( \left( \frac{2 \sigma^p}{1 + 3 \sigma^2} \right)^{1/p} \) 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 non-stationary iteration. Hence we have established the following theorem:

**THEOREM 3.9.** Let \( A = M - N \) 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 \( \tau \), then the non-stationary Chebyshev iterative method also converges, and has a smaller average reduction factor for all \( p > 1 \).

### 3.4. First-order non-stationary Chebyshev iterative sampler.

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, then give the asymptotic reduction factor.

The first-order non-stationary iterative sampler uses the iteration

\[
y^{l+1} = G_ly^l + g_l
\]

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

**LEMMA 3.10.**

\[
\text{Cov}(y^{l+1}) = G_l \text{Cov}(y^l) G_l^T + \text{Cov}(g_l).
\]
The unique fixed point variance is $A^{-1}$, i.e.,

$$A^{-1} = G_1 A^{-1} G_1^T + \text{Cov}(y_l).$$  \hspace{1cm}(3.14)$$

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

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

or

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

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

**Theorem 3.12.**

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

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 3.13.** Let $A = M_\tau - N_\tau$ be a symmetric splitting with the extreme eigenvalues of $M^{-1}_\tau A$ satisfying $0 < \lambda_1 < \lambda_n$. If the splitting converges (as an iterative solver), for some relaxation parameter $\tau$, then the stationary iterative sampling algorithm also converges, and the non-stationary Chebyshev iterative sampling algorithm also converges, and has a smaller average reduction factor for all $p > 1$. 

---

**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, $\left( \frac{2 - \sigma_p}{1 + \sigma_p} \right)^{1/p}$, with $\sigma$ defined in (3.12). Along the horizontal axis are values of the ratio of the extreme eigenvalues of $M^{-1} A$, $\lambda_1 / \lambda_n \in [0, 1]$. 

**Lemma 3.11.** The unique fixed point variance is $A^{-1}$, i.e.,

$$A^{-1} = G_1 A^{-1} G_1^T + \text{Cov}(y_l).$$  \hspace{1cm}(3.14)$$

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

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

or

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

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

**Theorem 3.12.**

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

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 3.13.** Let $A = M_\tau - N_\tau$ be a symmetric splitting with the extreme eigenvalues of $M^{-1}_\tau A$ satisfying $0 < \lambda_1 < \lambda_n$. If the splitting converges (as an iterative solver), for some relaxation parameter $\tau$, then the stationary iterative sampling algorithm also converges, and the non-stationary Chebyshev iterative sampling algorithm also converges, and has a smaller average reduction factor for all $p > 1$. 

---
4. Second-order methods. Axelsson points out [1, Remark 5.11] two deficiencies of the first-order Chebyshev iterative method as a solver: Firstly, the number of steps \( p \) needs to be selected in advance, with the method not being optimal for any other number of steps. Secondly, the first-order iteration is numerically unstable, so computer implementation probably won’t 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, that 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, §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 (\( M = I \))

\[
x^l = x^0 - \frac{1}{2} \beta_0 r^0 \quad \text{and} \quad x^{l+1} = \alpha_l x^l + (1 - \alpha_l) x^{l-1} - \beta_l r^l
\]

(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 \( l = 1, 2, \ldots \).

**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 \( x^{l+1} = \alpha_l x^* + (1 - \alpha_l) x^* = x^* \). \( \Box \)

Now subtract the (invariance) expression \( x^* = \alpha_l x^* + (1 - \alpha_l) x^* \) from (4.1) and use Lemma 3.2 to give

\[
x^{l+1} - x^* = \alpha_l (x^l - x^*) + (1 - \alpha_l) (x^{l-1} - x^*) - \beta_l A e^l
\]

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

\[
e^l = Q_l(A) e^0
\]

(4.2)

which is true for \( l = 0 \), and is true for \( l = 1 \) since \( e^1 = e^0 - \frac{1}{2} \beta_0 r^0 = (I - \frac{1}{2} \beta_0 A) e^0 \) (using Lemma 3.2) giving the result with \( Q_1(\lambda) = (1 - \frac{1}{2} \beta_0 \lambda) \). 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 for \( Q_{l+1}(\cdot) \) that

\[
Q_{l+1}(A) = (\alpha_l I - \beta_l A) Q_l(A) + (1 - \alpha_l) Q_{l-1}(A)
\]

(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 \( Q_l(\cdot) \) 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’s 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. Which is to say that the second-order iteration can be made to fit the Chebyshev polynomials, but not necessarily any other set of orthogonal polynomials!

Convergence and the reduction factor are given by the expression (4.2), with the analysis for the first-order Chebyshev iterative solver giving the result that the Chebyshev accelerated method is faster than the stationary method. In fact, for any $0 < \varepsilon < 1$, (4.2) shows that to assure a decrease in error

$$||e_l'||_A^{-1} = \frac{||e_0'||_A^v}{\epsilon} \leq \epsilon$$

for some real number $\nu$, it suffices to perform

$$p^* = \lceil \frac{\ln(\epsilon/2)}{\ln \sigma} \rceil$$

iterations of the non-stationary second-order Chebyshev solver [1, eqn 5.32].

4.2. Second-order Chebyshev iterative sampler. Analysis of the second-order non-stationary sampler follows the same route as the first-order non-stationary 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 $A^{-1} - 1$ is the invariant variance. We then subtract the iteration that states the invariance of the variance $A^{-1} - 1$ 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, §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

$$x_l+1 = (\alpha_l I - \beta_l M^{-1}A) x_l + (1 - \alpha_l) x_l^{-1} + \beta_l M^{-1}b$$

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 $\tau_l$. Accordingly, we write the second-order non-stationary iterative sampler as

$$y_l+1 = (\alpha_l I - \beta_l M_l^{-1}A) y_l + (1 - \alpha_l) y_l^{-1} + \beta_l g_l$$

for $l = 1, 2, \ldots$ with the first step using $y^1 = G_0 y^0 + g_0$ and $a_0 = 1$, and now $\{g_l\}$ are independent samples with Cov($g_l$) chosen so that Cov($y^l$) = $A^{-1} - 1$ ensures that Cov($y^l$) = $A^{-1}$ for $l \geq 1$. For the moment we will assume that is done, and work out Cov($g_l$) later in section 4.3.

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

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

with the iteration being

$$Y^{l+1} = \begin{pmatrix} \alpha_l G_l I \\ \alpha_l \tau_l I \end{pmatrix} Y^l + \begin{pmatrix} g_l \\ 0 \end{pmatrix}$$
(the first step has \( \alpha_0 = 1 \) as mentioned before). Denote

\[
G_l = \begin{pmatrix}
\alpha_l G_l & (1 - \alpha_l) I \\
I & 0
\end{pmatrix}
\quad \text{and} \quad
\gamma_l = \alpha_l \begin{pmatrix}
g_l \\
0
\end{pmatrix}.
\]

**Lemma 4.2.**

\[
\begin{align*}
\text{Cov} (Y^0) &= \begin{pmatrix}
\text{Cov} (y^0) & 0 \\
0 & 0
\end{pmatrix}, \\
\text{Cov} (Y^{l+1}) &= G_l \text{Cov} (Y^l) G_l^T + \text{Cov} (\gamma_l) \\
&= G_l \text{Cov} (Y^l) G_l^T + (\alpha_l)^2 \begin{pmatrix}
\text{Cov} (g_l) & 0 \\
0 & 0
\end{pmatrix}.
\end{align*}
\]  

(4.8)

**Proof.** Follows from the iteration (4.7) and the independence of \( Y^l \) and \( \gamma_l \). \( \square \)

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 \( \text{Cov} (Y^l) \) cannot be made constant. It turns out this does not matter, and all we need is the sequence of \( \text{Cov} (Y^l) \) that are given by starting the iteration with \( \text{Cov} (y^0) = A^{-1} \) and asserting the requirement that \( \text{Cov} (y^l) = A^{-1} \) for \( l = 1, 2, \ldots \), with the off-diagonal terms changing. This leads to the sequence of variances denoted \( \text{Cov} (Y^0) = \begin{pmatrix}
A^{-1} & 0 \\
0 & 0
\end{pmatrix} \) (so by definition \( K_0 = 0 \)) and then

\[
\text{Cov} (Y^l) = \begin{pmatrix}
A^{-1} & K_l \\
K_l^T & A^{-1}
\end{pmatrix}
\]

where the (1,1) block equals \( A^{-1} \) because \( \text{Cov} (g_l) \) is chosen to make this true. The sequence of variances then satisfy the recursion given in the following Lemma.

**Lemma 4.3.** The requirement that \( \text{Cov} (y^0) = A^{-1} \) implies that \( \text{Cov} (y^l) = A^{-1} \) for \( l = 1, 2, \ldots \) and so

\[
\begin{pmatrix}
A^{-1} \\
K_l^T
\end{pmatrix} = G_l \begin{pmatrix}
A^{-1} & K_l \\
K_l^T & A^{-1}
\end{pmatrix} G_l^T + \begin{pmatrix}
(\alpha_l)^2 \text{Cov} (g_l) & 0 \\
0 & 0
\end{pmatrix}
\]  

(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 \( l = 0, 1, 2, \ldots \), and \( K_0 = 0 \). The recursion for error on variance is then given by subtracting (4.9) from (4.8) to give

\[
\text{Cov} (Y^{l+1}) - \begin{pmatrix}
A^{-1} \\
K_l^T
\end{pmatrix} = G_l \left( \text{Cov} (Y^l) - \begin{pmatrix}
A^{-1} \\
K_l^T
\end{pmatrix} \right) G_l^T
\]

(4.10)

or

\[
\mathcal{E}^{l+1} = G_l \mathcal{E}^l G_l^T \quad l = 0, 1, 2, \ldots
\]

where we have defined the error in variance

\[
\mathcal{E}^0 = \text{Cov} (Y^0) - \begin{pmatrix}
A^{-1} & 0 \\
0 & 0
\end{pmatrix} = \text{Cov} (y^0) - A^{-1} \begin{pmatrix}
0 & 0 \\
0 & 0
\end{pmatrix}
\]

\[
\mathcal{E}^l = \text{Cov} (Y^l) - \begin{pmatrix}
A^{-1} & K_l \\
K_l^T & A^{-1}
\end{pmatrix} \quad l = 1, 2, \ldots.
\]
Hence by recurrence, we prove the following theorem:

**Theorem 4.4.**

\[ E_m = \left( \prod_{l=0}^{m-1} G_l \right) \left( \prod_{l=0}^{m-1} G_l \right)^T. \]

Denote the polynomial of the block matrix \( H_{m+1} = (\prod_{l=0}^{m} G_l) \) that satisfies

\[ H_1 = G_0 = \begin{pmatrix} G_0 & 0 \\ I & 0 \end{pmatrix} \]

\[ H_{l+1} = G_l H_l = \begin{pmatrix} \alpha_l G_l H_{l11} + (1 - \alpha_l) H_{l21} \\ \alpha_l G_l H_{l12} + (1 - \alpha_l) H_{l22} \end{pmatrix}. \]

Hence, \( H_{l11} \) and \( H_{l21} \) satisfy the recursion

\[ H_{l11}^{l+1} = \alpha_l G_l H_{l11}^{l} + H_{l21}^{l} \quad \text{and} \quad H_{l21}^{l+1} = H_{l11}^{l}. \]

Eliminating \( H_{l21}^{l} \) from the first equation, we establish the following statement:

**Theorem 4.5.** The 1, 1 block of the error polynomial \( H^l \) satisfies the recursion

\[ H_{l11}^{l+1} = \alpha_l G_l H_{l11}^{l} + (1 - \alpha_l) H_{l11}^{l-1} = \alpha_l (I - \tau_l M^{-1} A) H_{l11}^{l} + (1 - \alpha_l) H_{l11}^{l-1} \]

with \( H_{l11}^{0} = G_0 \).

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 non-stationary 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

\[ H_{11}^{m} (\lambda) = \frac{T_m((\lambda_1 + \lambda_n - 2\lambda) / (\lambda_1 - \lambda_n))}{T_m((\lambda_1 + \lambda_n) / (\lambda_1 - \lambda_n))} \]

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 \( m \)th iterate \( y_m \), which is the following theorem.

**Theorem 4.6.** The error in variance at the \( m \)th iteration is

\[ \text{Cov} (y_m) - A^{-1} = H_{11}^{m} (\text{Cov} (y_0) - A^{-1}) \left( H_{11}^{m} \right)^T. \]

**Proof.** Read off the 1, 1 block of the expansion

\[ E^{m} = \begin{pmatrix} H_{11}^{m} & H_{12}^{m} \\ H_{21}^{m} & H_{22}^{m} \end{pmatrix} \begin{pmatrix} \text{Cov} (y_0) - A^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} H_{11}^{m} & H_{12}^{m} \\ H_{21}^{m} & H_{22}^{m} \end{pmatrix}^T. \]

\[ \sigma^2 = \left( \frac{1 - \sqrt{\lambda_1 / \lambda_n}}{1 + \sqrt{\lambda_1 / \lambda_n}} \right)^2 \]
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 $H^m_l$ is the scaled Chebyshev polynomial (4.11), that is bounded on the interval $[\lambda_1, \lambda_n]$ by

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

where $\sigma$ is the average asymptotic reduction factor in (3.12) [1, p. 181]. $\square$

In practice, numerical estimates $\hat{\lambda}_1$ and $\hat{\lambda}_n$ of the extreme eigenvalues $\lambda_1$ and $\lambda_n$ of $M^{-1}A$ are used in the Chebyshev algorithms. Convergence is maintained, with modified reduction factors, as long as $\lambda_1 < \hat{\lambda}_1 < \hat{\lambda}_n < \lambda_n$ [22, p. 383].

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 $\epsilon > 0$, after

$$p^* = \lceil \frac{\ln(\epsilon/2)}{\ln(\sigma^p)} \rceil$$

iterations, the variance error reduction is smaller than $\epsilon$.

### 4.3. Noise variance in the second-order Chebyshev sampler.

When establishing Theorem 4.7, 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 $\text{Cov}(g_l)$ it is necessary to explicitly determine the blocks $K_l$ in (4.10).

We need the following results.

**Lemma 4.8.** For a symmetric splitting, $K_l = K_l^T$.

**Theorem 4.9.** For a symmetric splitting,

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

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

$$\kappa_{l+1} = \alpha\kappa_l + (1 - \alpha_l) \kappa_l, \text{ for } l = 1, 2, \ldots$$

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

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

by Lemma 2.7. For $l \geq 1$ we use the expression for the $1,1$ block in (4.9) to get

$$A^{-1} = \alpha_l^2 G_l A^{-1} G_l^T + \alpha_l (1 - \alpha_l) G_l K_l + \alpha_l (1 - \alpha_l) K_l^T G_l^T + (1 - \alpha_l)^2 A^{-1} + (\alpha_l)^2 \text{Cov}(g_l)$$

so $(\alpha_l)^2 \text{Cov}(g_l) = \left( 1 - (1 - \alpha_l)^2 \right) A^{-1} - \alpha_l^2 G_l A^{-1} G_l^T - \alpha_l (1 - \alpha_l) (G_l K_l + K_l^T G_l^T)$.

Now using Lemma 4.8 and Theorem 4.9, $\text{Cov}(g_l)$ can be rewritten as

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

(4.14)

Thus, the variance of the noise term is $\text{Cov}(g_l) = M^{-1}(c_l M + d_l N)M^{-1}$ for some real numbers $c_l$ and $d_l$. We have the following theorem:
Theorem 4.10. 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

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

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

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

such that $c_l := \frac{2-\tau_l}{\lambda_1} + (d_l - 1) \left(\frac{1}{\lambda_1} + \frac{1}{\tau_l} - 1\right)$, $d_l := \frac{2(1-\alpha_l)}{\alpha_l} \left(\frac{\alpha_l}{\tau_l}\right) + 1$, $\kappa_{l+1} := \alpha_l \tau_l + (1 - \alpha_l) \kappa_l$, and $\kappa_1 = \tau_0$. Then for the Chebyshev samples $\{y_l\}$,

$$E(y_l) \to 0 \quad \text{and} \quad \text{Cov}(y_l) \to A^{-1}$$

with asymptotic average reduction factors given by $\sigma$ and $\sigma^2$ respectively (cf. (3.12)).

Proof. The specification 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 4.9 provides $\kappa_l$. The reduction factors follow from (4.2) and Theorem 4.7. □

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 over-relaxation (SSOR) splitting $A = M_{SSOR} - N_{SSOR}$, with algorithms to be found, for example, in [9, 22]. This splitting depends on the SSOR parameter $\omega$, $0 < \omega < 2$. The choice $\omega = 1$ corresponds to forward and backward sweeps of the Gauss-Seidel stationary solver. Implementations of SSOR for other choices of $\omega$ correspond to forward and backward sweeps of successive over-relaxation (SOR); that is, the SSOR splitting is never explicitly calculated. Starting with a Chebyshev SSOR solver, Theorem 4.10 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 [4]. Thus, we present a Chebyshev accelerated Gibbs sampler from a Gaussian as Algorithm 1.

For arbitrary non-Gaussian distributions, Algorithm 1 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 $\lambda_1$ and $\lambda_n$ of $M_{SSOR}^{-1} A$ required by Algorithm 1 can be found inexpensively using a conjugate gradient algorithm [15, 17]. Analogous to the SSOR solver algorithms in [9, 22], the Chebyshev sampler implements sequential forward and backward sweeps of an SOR sampler [6, 19] (i.e., the SSOR splitting is never calculated, 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 multiply by $M^{-1}$ (solvers must deal with this same issue), but it also depends on how easy it is to factor $c_l M + d_l N$ [4]. A simplifying aspect of using SSOR is that this factorization need never be explicitly computed.
**Algorithm 1:** 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 $M_{SSOR}^{-1}A$; initial state $y^0$; maximum iteration $l_{\text{max}}$

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

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

for $l = 1, \ldots, l_{\text{max}}$ do
  if $l = 0$ then
    $d = \frac{2}{\alpha} - 1$;
    $c = (\frac{2}{\tau} - 1)d$;
    $\kappa = \tau$;
  else
    $d = 2(1 - \alpha)/\beta + 1$;
    $c = (\frac{2}{\tau} - 1) + (d - 1)(1/\tau + 1/\kappa - 1)$;
    $\kappa = \beta + (1 - \alpha)\kappa$;
  end

sample $z \sim N(0, I)$;

$b = d^{1/2}D^{1/2}_\omega z$;

$x = y^l + M_\omega^{-1}(b - Ay^l)$;

sample $z \sim N(0, I)$;

$b = c^{1/2}D^{1/2}_\omega z$;

$w = x - y^k + M_\omega^{-T}(b - Ax)$;

$y^{l+1} = \alpha(y^l - y^{l-1} + \tau w) + y^{l-1}$;

$\beta = (1/\tau - \beta\delta)^{-1}$;

$\alpha = \beta/\tau$;

end

By Theorem 4.10, the Chebyshev SSOR samples generated by Algorithm 1 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 $\sigma$); and the covariance matrix of the samples converges to $A^{-1}$ with asymptotic average reduction factor $\sigma^2$.

5. Numerical examples sampling from Gaussians at different resolutions. The development we have given of polynomial acceleration of sampling requires that the mean and inverse covariance of the target distribution are 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 specified mean vector and precision matrix. We now give an example of accelerating sampling from a Gaussian Markov random field (GMRF) in 3-dimensions that has a sparse precision matrix defined via a PDE (partial differential equation) and boundary conditions.

Our example uses the relationship between stationary Gaussian random fields and stochastic PDEs that was noted by Whittle [24] for the Matérn (or Whittle-Matérn [11]) class of covariance functions, that was also exploited in [2, 14]. Rather than stating the PDE, we find it more convenient to work with the equivalent variational

15
form, in this case

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

that 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 \(\mathcal{Q}(\bar{\phi}) = \bar{\phi}^T H \bar{\phi}\), where \(\bar{\phi}\) is a vector of nodal values and \(H\) is the Hessian, the resulting Gaussian Markov random field 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 \(\exp\left\{ -r/R \right\}\) 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 tri-linear interpolation from nodal values within each element. The examples also both use \(R = 1/4\), though differ in the number of nodes (or elements) in each coordinate direction.

### 5.1. A 5 × 5 × 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 1 with \(\omega = 1\)). Convergence was assessed by the relative error

\[
||A^{-1} - S||_2 / ||A^{-1}||_2,
\]

where the precision matrix \(A\) is the square of the Helmholtz operator described above, and \(S \approx \text{Cov}(y)\) is the empirical covariance matrix calculated over \(10^3\) sampler runs. Figure 5.1 illustrates convergence of the Chebyshev sampler iterates and compares the results to a Gibbs SSOR sampler in addition 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 \(M^{-1}_{SSOR} A\) for the SSOR splitting \(A = M_{SSOR} - N_{SSOR}\). A preconditioned conjugate gradient algorithm with preconditioner \(M_{SSOR}\) provided \(\lambda_1 = 1.268 \times 10^{-3}\) and \(\lambda_n = 0.9999\). Each of the \(10^3\) chains, for both the Chebyshev and Gibbs samplers, ran to iteration \(l = 500\).

The benchmark 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 Chebyshev accelerated sampler converged in finite precision after 60 iterations.

The average asymptotic reduction factor was \(\sigma = 0.9312\) for the Chebyshev error in the linear solver and for the error in the first moment of a Chebyshev sample (calculated via (3.12)). This suggests that \(p^* = 269\) iterations are required to reduce the linear solver error to \(\varepsilon = 10^{-8}\) (cf. (4.4)), and in fact a Chebyshev SSOR solver of \(Ax = b\) for a randomly generated \(b\) 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 (Theorem 4.7), 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
Fig. 5.1. Iterative SSOR sampler convergence to $N(0, A^{-1})$ for a $125 \times 125$ precision matrix $A$. The vertical axis is the relative error $\|A^{-1} - S^t\|_2/\|A^{-1}\|_2$. On the horizontal axis is the number of iterations. Both SSOR and Chebyshev accelerated SSOR were implemented with relaxation parameter $\omega = 1$.

Fig. 5.2. A draw from a $n = 2.7 \times 10^4$ dimensional Gaussian using a Chebyshev accelerated SSOR sampler after $5n$ iterations.

converge. In fact, (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 un-accelerated 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 imputed into the Chebyshev sampler for this run was \( y^0 = 0 \), and the extreme eigenvalues of \( M^{-1}SSOR \) 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 conjugate gradient 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 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 \( \rho(G)^2 5n \approx 0.69 \) 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) that is the standard route to understanding linear iterative solvers is also useful for constructing stationary stochastic iterations that converge to a distribution having a desired covariance matrix. The simplest such case is Gibbs sampling of a Gaussian distribution, when the 'noise' vectors are iid from a multivariate Gaussian. However, the stochastic iteration, and matrix splitting analysis, applies more generally.

The Chebyshev polynomial acceleration of linear solvers may then be applied to accelerate convergence in mean and covariance of the stochastic iteration, used to sample from a desired target distribution. 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 is 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 [12, 19] does offer the possibility of adapting to the mean and covariance within the iteration, as in the AM (adaptive Metropolis) algorithm. We have 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 [7], and (classical) relaxation which is the term Southwell used for early stationary iterative solvers [23]. In particular, we were curious if 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 forms the basis for applying polynomial acceleration to the stochastic relax-
The correspondence has been noted before, e.g. by Goodman and Sokal [10] 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, is the application of polynomial acceleration to Gibbs sampling of normal distributions. 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 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 sampling algorithms is presently in a state of sophistication 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 convergence of sampling algorithms is a more delicate issue than establishing 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 un-accelerated 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.

This paper has followed the development given by Axelsson [1] in specifying polynomial acceleration and determining convergence properties. 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 [8], that 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 [21].

7. Appendix.

Lemma 2.4 \( A^{-1} - G \tau A^{-1} G^T \tau = \tau M^{-1} \left( \frac{1}{\tau} M + \frac{1}{\kappa} M^T - A \right) M^{-T}. \)

Proof. \( A^{-1} - G \tau A^{-1} G^T \tau = M^{-1} (M \tau A^{-1} M^T) M^{-T} - M^{-1} N_\tau A^{-1} (M \tau^{-1} N_\tau)^T = M^{-1} (M \tau + M^T - A) M^{-T}. \)

Lemma 2.5 \( A^{-1} - G \tau A^{-1} G^T \tau = M^{-1} (M \tau^T + N_\tau) M^{-T}. \)

Proof. Set \( N_\tau = M \tau - A \) in the proof to Lemma 2.4. □
LEMMA 4.3 The requirement that Cov \( (y^0) = A^{-1} \) implies that Cov \( (y^l) = A^{-1} \) for \( l = 1, 2, \ldots \) and so

\[
\begin{pmatrix}
A^{-1} & K_{l+1} \\
K_{l+1}^T & A^{-1}
\end{pmatrix} = G_l \begin{pmatrix}
A^{-1} & K_l \\
K_l^T & A^{-1}
\end{pmatrix} G_l^T + \begin{pmatrix}
(\alpha_l)^2 \text{Cov}(g_l) & 0 \\
0 & 0
\end{pmatrix}
\]

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 \( \text{Cov}(y^0) = A^{-1} \) which gives \( \text{Cov}(y^0) = \begin{pmatrix} A^{-1} & 0 \\ 0 & 0 \end{pmatrix} \)

and hence Lemma 4.2 gives

\[
\text{Cov}(y^1) = G_0 \begin{pmatrix}
A^{-1} & 0 \\
0 & 0
\end{pmatrix} G_0^T + \begin{pmatrix}
\text{Cov}(g_0) & 0 \\
0 & 0
\end{pmatrix} = \begin{pmatrix}
A^{-1} & G_0 A^{-1} \\
A^{-1} G_0^T & A^{-1}
\end{pmatrix}
\]

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

\[
\begin{pmatrix}
A^{-1} & K_{l+1} \\
K_{l+1}^T & A^{-1}
\end{pmatrix} = \begin{pmatrix}
\alpha_l G_l & (1 - \alpha_l) I \\
I & 0
\end{pmatrix} \begin{pmatrix}
A^{-1} & K_l \\
K_l^T & A^{-1}
\end{pmatrix} \begin{pmatrix}
\alpha_l G_l^T & (1 - \alpha_l) I \\
(1 - \alpha_l) I & 0
\end{pmatrix} + \begin{pmatrix}
(\alpha_l)^2 \text{Cov}(g_l) & 0 \\
0 & 0
\end{pmatrix}
\]

\[
= \begin{pmatrix}
\alpha_l A^{-1} G_l^T + (1 - \alpha_l) K_l & \alpha_l G_l A^{-1} + (1 - \alpha_l) K_l^T \\
(1 - \alpha_l) K_l & (1 - \alpha_l) K_l^T
\end{pmatrix} + \begin{pmatrix}
(\alpha_l)^2 \text{Cov}(g_l) & 0 \\
0 & 0
\end{pmatrix}.
\]

\[\square\]

LEMMA 4.8 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 \( K_{l+1} = \alpha_l G_l A^{-1} + (1 - \alpha_l) K_l^T \) 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 2.3. \[\square\]

THEOREM 4.9 For a symmetric splitting, \( K_l = G_k, A_{-1} \), for \( l = 1, 2, \ldots \), where the parameter \( \kappa_l \) 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 4.3 and 4.8, the recursion in \( K_l \) gives

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

\[
= \alpha_l (I - \tau l M^{-1} A) A^{-1} + (1 - \alpha_l) (I - \kappa l M^{-1} A) A^{-1}
\]

\[
= (I - [\alpha_l \tau_l + (1 - \alpha_l) \kappa_l] M^{-1} A) A^{-1}
\]

so the expansion and recursion hold for \( l + 1 \), and the result follows by induction. \[\square\]

Acknowledgements. This work was supported by the New Zealand Institute for Mathematics and its Applications thematic programme on PDEs.

REFERENCES


