



Polynomial accelerated solutions to a LARGE Gaussian model for imaging biofilms: in theory and finite precision

Authors: Albert E. Parker, ,Betsey Pitts, Lindsey
Lorenz, and Philip S. Stewart

This is an Accepted Manuscript of an article published in [Journal of the American Statistical Association](#) on June 28, 2018 available online: <http://www.tandfonline.com/10.1080/01621459.2017.1409121>.

Parker, Albert E., Betsey Pitts, Lindsey Lorenz, and Philip S. Stewart. "Polynomial Accelerated Solutions to a Large Gaussian Model for Imaging Biofilms: In Theory and Finite Precision." *Journal of the American Statistical Association* (June 28, 2018): 1–12.
doi:10.1080/01621459.2017.1409121.

Made available through Montana State University's [ScholarWorks](#)
scholarworks.montana.edu

Polynomial Accelerated Solutions to a Large Gaussian Model for Imaging Biofilms: In Theory and Finite Precision

Albert E. Parker^a, Betsey Pitts^b, Lindsey Lorenz^b, and Philip S. Stewart^c

^aCenter for Biofilm Engineering, Department of Mathematical Sciences, Montana State University, Bozeman, MT; ^bCenter for Biofilm Engineering, Montana State University, Bozeman, MT; ^cCenter for Biofilm Engineering, Department of Chemical and Biological Engineering, Montana State University, Bozeman, MT

ABSTRACT

Three-dimensional confocal scanning laser microscope images offer dramatic visualizations of living biofilms before and after interventions. Here, we use confocal microscopy to study the effect of a treatment over time that causes a biofilm to swell and contract due to osmotic pressure changes. From these data (the video is provided in the supplementary materials), our goal is to reconstruct biofilm surfaces, to estimate the effect of the treatment on the biofilm's volume, and to quantify the related uncertainties. We formulate the associated massive linear Bayesian inverse problem and then solve it using iterative samplers from large multivariate Gaussians that exploit well-established polynomial acceleration techniques from numerical linear algebra. Because of a general equivalence with linear solvers, these polynomial accelerated iterative samplers have known convergence rates, stopping criteria, and perform well in finite precision. An explicit algorithm is provided, for the first time, for an iterative sampler that is accelerated by the synergistic implementation of preconditioned conjugate gradient and Chebyshev polynomials. Supplementary materials for this article are available online.

1. Introduction

Microbial biofilms are ubiquitous in nature. They form on our teeth, on rocks in creek bottoms, in pipes on oil drilling rigs, and inside intravenous catheters. They are everywhere there is water and a carbon source (Hall-Stoodley, Costerton, and Stoodley 2004). A bacterial biofilm is a community of bacteria aggregated together in a gel-like matrix of extracellular polymers. This appears to be the preferred growth mode for bacteria because it confers several advantages to the individual bacteria that compose the biofilm, including increased tolerance against antimicrobial treatments (Stewart 2015).

The venerable approach to quantifying bacterial abundances is to put a sample of bacteria onto agar in a petri dish and then count *colony forming units* that become visible to the naked eye as the bacteria grow exponentially on the agar. This approach is still used by researchers, government agencies, and standard setting organizations (e.g., ASTM International) to quantify bacterial populations found in many different environments. Incredible advances in technology now allow more in-depth analyses to be performed. Molecular techniques identify bacterial phylogenies, mass spectrometry reveals how bacteria communicate and conduct warfare, and microscopy allows fantastic visualizations of individual cells interacting with each other and their surroundings.

Confocal scanning laser microscopy allows three-dimensional (3D) images to be constructed of dynamic living

biofilms over time at resolutions smaller than 1 μm . Confocal microscopes (CM) capture a set of planar “slices” or images, parallel to, and at different distances from, the bottom of the biofilm where it is attached to a surface. The 3D image is generated by stacking the 2D slices. The laser illuminates bacteria that have either been stained, or genetically modified, to fluoresce when excited by the laser. In this article, we analyze a sequence of CM images over time (i.e., a video, see supplementary material) of a green fluorescing *Staphylococcus aureus* biofilm grown under controlled conditions in an engineered system. *S. aureus* is a common human pathogen that is notorious for its potential to evolve into an antibiotic-invulnerable methicillin resistant strain (MRSA).

At each spatial location corresponding to a pixel in the image, the CM records the intensity of the biofilm's fluorescence as an 8-bit integer (i.e., a value between 0 and 255). In our example, the horizontal (xy) field of view for each planar slice is $620 \mu\text{m} \times 620 \mu\text{m}$ with a vertical (z) range of $112 \mu\text{m}$. The 3D pixelation is $512 \times 512 \times 17$ pixels with a 512×512 pixel representation for each planar slice (i.e., the physical representation of an xy pixel is $1.2 \mu\text{m}$); and there are 17 planar slices stacked together with $7 \mu\text{m}$ between each pair of z -slices. Each z -slice is identified with an integer value between $z = 1$ (where the biofilm is attached to a surface) and $z = 17$ (the z -slice at the very top of the image). In the video that we analyze, approximately four $512 \times 512 \times 17$ images are captured each

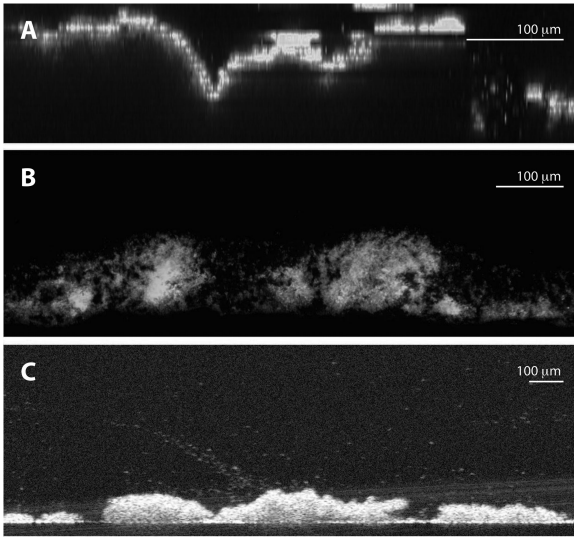


Figure 1. Cross-sectional views (i.e., pixels in the xz dimensions) of fluorescent *S. aureus* biofilms using three different imaging techniques. The attachment surface is along the bottom ($z = 0$) in each panel and the bulk fluid interface is at the top. Illumination in all three panels is from above (the bulk fluid side). A: A CM image of a biofilm that shows the typical attenuation of fluorescence intensity with increasing depth into the biofilm. B: A cryosection of a biofilm grown in an independent experiment under the same conditions as in (A) showing that these biofilms are typically solid, with microcolonies that are at least $100 \mu\text{m}$ thick. Imaging of this cryosection was done on an upright epifluorescent microscope. C: An image of another biofilm from the same experiment as (B), collected using Optical Coherence Tomography (OCT). OCT uses interferometry rather than fluorescence to form an image. The biofilm is shown to be solid with this method.

minute over 45 min. During the course of the video, one can see the effect of a salt water treatment on the biofilm. The biofilm goes through a series of contraction and swelling events due to osmotic pressure changes after multiple applications and removals of the treatment. Here, we present a Bayesian analysis of 10 min of the video (40 frames) that captures the response of the biofilm as the salt water is removed and then applied again.

CM images are commonly analyzed using tools available in the software packages Imaris or COMSTAT (Heydorn et al. 2000). To quantify biofilm characteristics (without uncertainty quantification), these packages typically perform calculations on bright pixels (Lewandowski and Beyenal 2014). For example, to estimate a biofilm volume as we do here, Imaris simply counts bright pixels. Such an approach is reasonable if the biofilm being imaged is thin, that is, less than $100 \mu\text{m}$. We have found that when imaging thicker biofilms with a CM, the light attenuates markedly as it passes through the top layers of the biofilm so that the bacteria in the interior of the biofilm do not fluoresce at all (Figure 1 A). Independent analyses suggest that the biofilm contains viable bacteria all the way through (Figure 1 B and C). For the thick biofilm featured in Figure 1 A, simply counting bright pixels would clearly produce a biased volume estimate. We apply a polynomial accelerated iterative sampler to the problem of constructing biofilm surfaces and quantifying biofilm volumes and the associated uncertainties from a video of 3D CM images. This inverse problem is cast within a linear Bayesian framework with a Gaussian likelihood. After sampling variance parameters, we apply the methods of preconditioned conjugate gradients (PCG) and Chebyshev polynomials to iteratively sample from a multivariate Gaussian in order to garner information about the posterior density of the biofilm surface from which characteristics such as volume can be inferred. For this biofilm

imaging problem, it is too computationally and memory intensive to sample by either the conventional Cholesky factorization or conventional componentwise-wise iterative Gibbs sampling.

The rest of this article is structured as follows. In Section 2, we present the linear Bayesian inverse problem that we solve. In Section 3, we review recent polynomial methods for iteratively sampling from Gaussians that have been derived from numerical linear algebra. This review includes: conditions for convergence; convergence rates; and the performance of these samplers in finite precision. These previous results are built upon in Section 4 to present an explicit algorithm, for the first time, for the synergistic implementation of a PCG sampler and a Chebyshev sampler that capitalizes on the strengths and overcomes some of the weaknesses when PCG and Chebyshev samplers are used alone. In Section 5, image analysis results are presented. We conclude in Section 6 with a discussion and future directions.

2. Inverse Problem for Quantifying 3D Images of Biofilms

We consider a linear model of the biofilm’s surface given CM images of thick biofilms such as presented in Figure 1 A (as suggested, e.g., in Sheppard and Shotton 1997). Each image represents the biofilm in a $620 \mu\text{m} \times 620 \mu\text{m}$ field of view on a 512×512 pixelated lattice. To represent the biofilm’s surface from CM data, first the data is thresholded so that pixels with an intensity value less than 50 are set to 0, and other pixels are set to 1. The biofilm’s surface (or thickness) at the i th pixel location in the 512×512 lattice is set by first identifying the set of z values for which there are adjacent planar z -slices with nonzero pixels in the thresholded data. The thickness is then set to the largest value of z in this identified set. Surface representations of two images from the video are shown in Figure 2. The results of this edge detection scheme agree well with a Sobel edge detector implementation but is seven times faster (in Matlab). Given a surface representation, y , of a biofilm over the 512×512 lattice, we estimate the volume of the biofilm by summing the components of y .

The linear statistical model that we apply to the surface profile is

$$y = F\theta + \varepsilon.$$

The random vector y is a representation of the biofilm surface calculated from the CM image; θ is the true biofilm surface that we want to estimate; the matrix F implements possible blurring of the surface due to the point spread function of the CM; and $\varepsilon \sim N(0, \Sigma_y)$. Based on the above specification, the likelihood is $\pi(y|\theta, \Sigma_y) = N(F\theta, \Sigma_y)$. We introduce our prior assumption that the surface changes smoothly by assuming $\pi(\theta) = N(0, \frac{1}{\lambda}W^{-1})$, where λ is an unknown precision parameter (i.e., a regularizer (Bardsley 2012)) that controls the level of smoothing of the surface. The $512^2 \times 512^2$ unscaled prior precision matrix W is the Laplacian considered by Higdon (2006) and Rue and Held (2005)

$$[W]_{ij} = 10^{-4}\delta_{ij} + \begin{cases} n_i & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } \|s_i - s_j\|_2 \leq 1. \\ 0 & \text{otherwise} \end{cases}$$

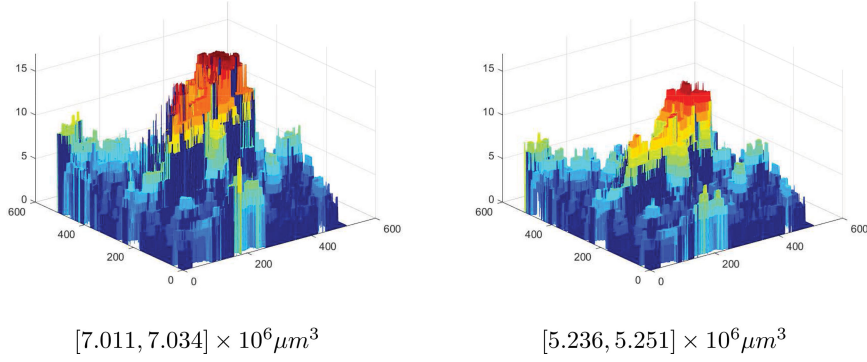


Figure 2. Biofilm surfaces for two images (frames 31 and 35) captured just over a minute apart by the CM. The coloring scheme provides better visualization of the surface, with red indicating higher features and blue indicating lower features on the surface of the biofilm. Axes are in pixels, where each xy pixel corresponds to $1.2 \mu\text{m}$, and each vertical pixel corresponds to $7 \mu\text{m}$. Under each image is a 99% credible interval for the corresponding biofilm’s volume. The drop in the biofilm’s thickness, and the corresponding drop in volume, at frame 35 is due to application of the salt water treatment after frame 31.

The locations $\{s_i\}$ are on the 512×512 lattice over the $620 \mu\text{m} \times 620 \mu\text{m}$ 2D domain. The scalar n_i is the number of points in the lattice that neighbor s_i (four in the interior), that is, that have distance $1.2 \mu\text{m}$ (1 pixel) from s_i . This specification presumes that the biofilm surface at each location, conditioned on the surface at the closest locations in the lattice, is independent of the rest of the surface (Geman and Geman 1984; Higdon 2006; Rue and Held 2005). We have investigated neighborhood sizes as large as five for a small subset of the imaging data, with no discernable effect on the volumes that we report here. Increasing the neighborhood size makes the computations even more expensive due to the decrease in the sparsity of the precision matrix of the posterior (see Section 4.3).

The distribution of θ given everything else is the multivariate Gaussian

$$\pi(\theta|y, \Sigma_y, \lambda, W) = N\left(A^{-1}F^T \Sigma_y^{-1}y, A^{-1}\right), \quad (1)$$

with precision matrix $A = F^T \Sigma_y^{-1}F + \lambda W$ (Calvetti and Somersalo 2007; Higdon 2006). This shows how to apply the sampler presented here to any linear model with arbitrary and fixed F , Σ_y , λ , and W . Our goal, is to find estimates of θ , Σ_y , and λ given data y , the process F , and the unscaled precision W . We make some simplifying assumptions that are appropriate for these data: $\Sigma_y = \sigma^2 I$, the errors when measuring the surface at each location in the lattice are iid with unknown variance σ^2 ; and $F = I$, there is no blurring of the surface due to the point spread function because adjacent pixel locations in space are far from each other ($1.2 \mu\text{m}$) compared to the 200 nm xy resolution of the CM (Sheppard and Shotton 1997). Nontrivial F might be required, for example, when using the higher pixel resolution capabilities of the CM that we use when imaging biofilms that are not dynamically changing over time, or when interpolating the biofilm surface at spatial resolutions finer than the pixel resolution of collected CM data. Diffuse Gamma hyperpriors for each of $1/\sigma^2$ and λ (using the parameterization $\pi(\lambda) \propto \lambda^{\alpha-1} e^{-\beta\lambda}$ with $\alpha = 1$ and $\beta = 10^{-4}$), that we assume are independent, complete the Bayesian specification.

Our goal is to estimate the posterior

$$\pi(\theta, \sigma^2, \lambda|y) \propto \pi(y|\theta, \sigma^2, \lambda)\pi(\theta)\pi(\sigma^2, \lambda). \quad (2)$$

We use the mean of the posterior as the Bayesian estimate $\hat{\theta}$ of the biofilm surface θ . Uncertainty of this estimate is quantified

by constructing a Markov chain of samples from the posterior (2). We use the conditional (Gibbs) sampling approach used by Higdon (2006) and Bardsley (2012) to draw samples $(\theta, \sigma^2, \lambda)$ from the posterior. Given the specification above, the distribution of $1/\sigma^2$ and λ conditioned on everything else is a product of Gamma distributions (Higdon 2006). With the assumptions made earlier regarding F and Σ_y , the distribution of θ conditioned on everything else (1) simplifies to a 512^2 dimensional Gaussian

$$\pi(\theta|y, \sigma^2, \lambda) = N\left(\frac{1}{\sigma^2}A^{-1}y, A^{-1}\right) \quad (3)$$

with precision matrix $A = \frac{1}{\sigma^2}I + \lambda W$.

To draw samples from (2), we first sample from the conditional Gammas for $1/\sigma^2$ and λ . Given these precision parameter values, the second step is to sample from (3). The Gaussian in (3) is massive. Therefore, conventional sampling techniques that utilize the Cholesky factorization are too expensive to apply. Instead, we apply an iterative PCG-Chebyshev sampler, derived from the PCG and Chebyshev iterative optimizers, to generate Gaussian samples.

3. Iteratively Sampling the Massive Multivariate Gaussian

3.1. Iterative Sampling and Linear Solving

The Cholesky factorization is the conventional way to produce samples from a multivariate Gaussian and is also the preferred method for solving moderately sized linear systems (Rue and Held 2005). For *large* linear systems, iterative solvers are the methods of choice due to their inexpensive cost per iteration and small computer memory requirements. For very large dimensional multivariate Gaussians of the form $N(A^{-1}b, A^{-1})$, given an $n \times n$ SPD precision matrix A and fixed vector b , the well-known component-wise Gibbs sampler (Gelman et al. 1995; Gilks, Richardson, and Spiegelhalter 1996) is one of the few general iterative samplers available that samples each component of a random vector conditioned on the current state of the other components. At the k th iteration, one sweep of this Gibbs sampler may be written in matrix form as

$$\theta^{k+1} = M^{-1}N\theta^k + M^{-1}c^k, \quad (4)$$

where $c^k \stackrel{\text{iid}}{\sim} N(b, M^T + N)$, $M = L + D$, $N = -L^T$, L is the strictly lower triangular part of A , and D is the diagonal of A . Note that $M^T + N = D$. Repeating this sweep indefinitely produces iterates $\{\theta^k\}$ that converge in distribution to $N(A^{-1}b, A^{-1})$ as long as A is SPD (Adler 1981; Amit and Grenander 1991).

Perhaps, it is not so well known that the forward component sweep Gibbs sampler is essentially identical to the Gauss–Seidel iterative method that solves $Ax = b$ for x given an $n \times n$ matrix A and fixed vector b (Adler 1981; Amit and Grenander 1991). At the k th iteration, one sweep of the Gauss–Seidel linear solver may be written in matrix form as

$$x^{k+1} = M^{-1}Nx^k + M^{-1}b. \quad (5)$$

Repeating this sweep indefinitely produces iterates $\{x^k\}$ that converge to $A^{-1}b$ as long as A is SPD.

Remarkably, the only difference between the sampler iteration (4) and the solver iteration (5) is the introduction of a random vector c^k instead of a fixed right-hand side b ! This equivalency in form shows that both the forward component sweep Gibbs sampler from a multivariate Gaussian and the Gauss–Seidel linear solver are equivalent in the sense that both utilize the same iteration operator $M^{-1}N$ and also converge under the same conditions (A is SPD) with the same convergence rate (Roberts and Sahu 1997; Young 1971). Extensions of this Gibbs sampler (Barone and Frigessi 1990; Roberts and Sahu 1997), equivalent to the successive-over-relaxation (SOR) linear solver and the symmetric-SOR (SSOR) linear solver (Axelsson 1996; Golub and Van Loan 1989; Saad 2003) were the state-of-the-art for iterative samplers until only recently. SOR and SSOR were used as linear solvers in the 1950’s and are now considered rather slow (Saad and van der Vorst 2000). These solvers and samplers are referred to as *stationary* methods by numerical analysts because the same operator is applied to the current state at each iteration to generate the next state. Today, stationary iterative solvers are used as preconditioners at best, while CG polynomial methods (Hestenes and Stiefel 1952) are the current state-of-the-art because they can solve a linear system in a finite number of steps (Saad and van der Vorst 2000). Iterative samplers, on the other hand, have lagged behind. There has been a recent push to adapt more sophisticated iterative linear solvers to the job of sampling.

The prescription for the sampler in (4) and for the linear solver in (5) emphasizes that there is a general equivalence between sampling and solving. The first step is to identify matrices M and N such that $A = M - N$ is a *matrix splitting* of the precision matrix A . Fox and Parker (2017) applied this matrix splitting formalism from numerical analysis to show how to convert *any* solver of $Ax = b$ of the form

$$x^{k+1} = (1 - \alpha_k)x^k + \alpha_k c^k + \alpha_k \tau_k M^{-1}(b - Ax^k), \quad (6)$$

(as described, e.g., in Golub and Van Loan (1989) and Axelsson (1996)) into an iterative sampler of a multivariate Gaussian $N(A^{-1}b, A^{-1})$

$$\theta^{k+1} = (1 - \alpha_k)\theta^k + \alpha_k \theta^k + \alpha_k \tau_k M^{-1}(c^k - A\theta^k). \quad (7)$$

The parameters α_k and τ_k in both (6) and (7) are updated according to the specific linear solver method. For example, the assignment $\alpha_k = \tau_k = 1$ corresponds to the stationary solver (5) and sampler (4). As was the case when comparing (5) and (4), the solver (6) utilizes a fixed vector b , while the sampler (7) uses a random vector c^k ; in this case, $c^k \sim N(b, a_k M^T + b_k N)$, where a_k and b_k are functions of α_k and τ_k (Fox and Parker 2017). The key when implementing either the solver or the sampler is to pick a splitting for which it is inexpensive to perform the operations by M^{-1} (e.g., when M is triangular); for the sampler, it is also crucial to be able to inexpensively sample from $c^k \sim N(b, a_k M^T + b_k N)$.

This similarity in form assures that these solvers and samplers have the same conditions for convergence.

Lemma 3.1 (Fox and Parker 2017, Theorem 5). Let A be SPD and $A = M - N$ be any matrix splitting. The linear solver (6) with a set of parameters $\{\alpha_k\}$, $\{\tau_k\}$ that are independent of $\{x^k\}$ converges to $A^{-1}b$ (i.e., $x^k \rightarrow A^{-1}b$) if and only if the sampler (7) converges, $\theta^k \xrightarrow{D} N(\mu = A^{-1}b, A^{-1})$.

After k iterations of the solver (6), a k th-order polynomial p_k is generated that reduces the solver’s error, $\|x^{k+1} - A^{-1}b\|$, according to

$$(x^{k+1} - A^{-1}b) = p_k[I - M^{-1}N](x^0 - A^{-1}b),$$

(Axelsson 1996). The notation $p[\cdot]$ indicates the (scalar or matrix) argument to the polynomial, and the notation $p_k[\cdot](v - w)$ (for vectors or matrixes v and w) indicates (matrix) multiplication of $p_k[\cdot]$ and $(v - w)$. To ease notation, we set $P_k := p_k[I - M^{-1}N]$. For example, the assignment $\alpha_k = \tau_k = 1$ in (6) and (7), that correspond to the stationary solver and sampler in (5) and (4), respectively, yields the polynomial $p_k[\lambda] = (1 - \lambda)^k$ so that $p_k[I - M^{-1}N] = P_k = (M^{-1}N)^k$. The assignment of $\{\alpha_k\}$ and $\{\tau_k\}$ in (6) to other values corresponds to a *polynomial accelerated solver* when $P_k \neq (M^{-1}N)^k$ and convergence is faster than the stationary solver (5). The following Theorem shows that this same polynomial reduces the sampler error in the first and second moments, $\|E(\theta^k) - A^{-1}b\|$ and $\|\text{var}(\theta^k) - A^{-1}\|$. In other words, applying the prescription (7) based on a polynomial accelerated solver (6) always results in a polynomial accelerated sampler.

Theorem 3.1 (Fox and Parker 2017, Corollaries 6 and 7). Suppose that the polynomial accelerated linear solver (6) converges. Then, it converges with geometric convergence rate $\rho = (\lim_{k \rightarrow \infty} \max_{\lambda} |p_k[\lambda]|)^{1/k}$, where p_k is the k th-order polynomial recursively generated by iterating (6). Under the conditions of Lemma 3.1, the polynomial accelerated sampler (7) also converges with

$$E(\theta^{k+1}) = \mu + P_k(E(\theta^0) - \mu) \rightarrow \mu = A^{-1}b$$

with geometric convergence rate ρ where $P_k := p_k[I - M^{-1}N]$; and

$$\text{var}(\theta^{k+1}) = A^{-1} + P_k(\text{var}(\theta^0) - A^{-1})P_k^T \rightarrow A^{-1}$$

with geometric convergence rate ρ^2 .

Theorem 3.1 shows that solvers and samplers have the same convergence rate. Hence, the geometric rate of convergence of these iterative samplers can be found by looking up the corresponding solver in a numerical linear algebra textbook (e.g., Axelsson 1996; Golub and Van Loan 1989; Saad 2003; Young 1971). In fact, the theorem shows that samplers from distributions that have zero mean converge *faster* than the corresponding solver because the covariance matrix of the sampler converges with convergence rate $\rho^2 < \rho < 1$. For a solver (5) or sampler (4), because the linear operator is the same at each iteration then $P_k = (M^{-1}N)^k$, which shows that the convergence rate of these iterations is the spectral radius $\rho = \varrho(M^{-1}N)$ (Axelsson 1996; Golub and Van Loan 1989; Saad 2003; Young 1971). Hence, convergence of the sampler and solver can be assessed by simply checking whether $\varrho(M^{-1}N) < 1$ (Young 1971). This inequality is always satisfied for a component sweep Gibbs sampler of a Gaussian and also for a Gauss–Seidel linear solver given an SPD A . The solver in (5) and the sampler in (4) are actually accelerated by the polynomial iterations (6) and (7) when the polynomial convergence rate ρ is less than the convergence rate $\varrho(M^{-1}N)$.

3.2. Optimal Iterative Samplers

The previous section gave a general method, that is, the correspondence between equations (6) and (7), for deriving a polynomial accelerated sampler from a polynomial accelerated solver. The goal is to find a sampler by tweaking α_k and τ_k and implicitly generating a different operator ($\neq M^{-1}N$) at each iteration so that the resulting polynomial sampler (7) converges faster than the stationary sampler (4). Fox and Parker (2014) applied this approach to attain an iterative sampler with an optimal geometric convergence rate using Chebyshev polynomials. We mean optimal with respect to all iterations (7) that have coefficients $\{\alpha_k, \tau_k\}$ independent of the states $\{\theta^k\}$. Parker and Fox (2012) accelerate sampler convergence even more to only a finite number of iterations using CG polynomials (Algorithm 1 with $M = I$); in this case, the coefficients are not independent of the states.

In the rest of this section, we present the strengths and limitations of these and other available iterative Gaussian samplers. The goal is to derive a sampler (in Section 4), that is, provably convergent in exact arithmetic, has an optimal geometric convergence rate, and performs well in finite precision.

3.2.1. Chebyshev Accelerated Sampling

Chebyshev polynomial acceleration can be applied via equations (6) and (7) for any *symmetric matrix splitting* (i.e., M and N are symmetric) (Golub and Van Loan 1989; Fox and Parker 2014). The coefficients $\{\tau_k, \alpha_k\}$ in a Chebyshev implementation are functions of the extreme real eigenvalues λ_{\min} and λ_{\max} of $M^{-1}A$ (Axelsson 1996). **Theorem 3.1** shows that Chebyshev samplers are guaranteed to be accelerated compared to the stationary sampler (4) because the geometric convergence rate, ρ_{Cheby} , for the Chebyshev polynomial accelerated sampler satisfies

$$\rho_{\text{Cheby}} := \frac{1 - \sqrt{\text{cond}(A^{-1})}}{1 + \sqrt{\text{cond}(A^{-1})}} < \varrho(M^{-1}N), \quad (8)$$

where $\text{cond}(\cdot)$ is the condition number of a matrix (Axelsson 1996; Fox and Parker 2017). In fact, ρ_{Cheby} is the smallest geometric convergence rate among all polynomials generated by either (6) or (7) when $\{\alpha_k\}$ and $\{\tau_k\}$ are independent of the iterates x^k and θ^k (Axelsson 1996).

Theorem 3.1 shows that the errors in the mean and covariance of the samplers (7) decrease according to a specific polynomial. This allows, a priori to running a solver or sampler, for one to determine the number of iterations required for convergence to the target normal distribution. For example, after $k_{\text{stat}}^* = \lceil \frac{\log(\varepsilon)}{\log \varrho(M^{-1}N)} \rceil$ iterations, the stationary sampler (4) with mean $\mu = A^{-1}b \neq 0$ attains an error reduction in the mean $\|e^{k_{\text{stat}}^*}\|/\|e^0\| \leq \varepsilon$ for any $\varepsilon > 0$, where $e^k = E(\theta^k) - \mu$. The mean of the Chebyshev sampler converges even faster so that after

$$k_{\text{Cheby}}^* = \left\lceil \frac{\log(\varepsilon/2)}{\log(\rho_{\text{Cheby}})} \right\rceil, \quad (9)$$

iterations the error reduction in the mean is $\|e^{k_{\text{Cheby}}^*}\|_{A^\nu}/\|e^0\|_{A^\nu} \leq \varepsilon$ for some real number ν , where ρ_{Cheby} is specified in (8) (Fox and Parker 2014). Convergence in the variance is even faster after only $k_{\text{stat}}^{**} = k_{\text{stat}}^*/2$ iterations for stationary samplers, or after

$$k_{\text{Cheby}}^{**} = k_{\text{Cheby}}^*/2, \quad (10)$$

iterations for Chebyshev accelerated samplers (Fox and Parker 2014).

3.2.2. CG Accelerated Sampling

A CG solver also takes the form of (6) by setting $M = I$ and setting α_k and τ_k to functions of the residuals (Golub and Van Loan 1989, sec. 10.3.6). Using CG with other symmetric matrix splittings is called *preconditioned* CG (PCG). In this case, M is referred to as a preconditioner because M^{-1} is viewed as an approximation to A^{-1} (Saad 1992). The corresponding CG sampler was investigated in Parker and Fox (2012). We provide an explicit algorithm for a PCG sampler in Section 4.

Neither **Lemma 3.1** nor **Theorem 3.1** applies to CG polynomials because the CG coefficients ($\{\alpha_k, \tau_k\}$ in (6) and (7)) are functions of the residuals and hence not independent of either the solutions $\{x^k\}$ or of the samples $\{\theta^k\}$. The theory guaranteeing convergence of the CG sampler relies on the fact that a CG solver and CG sampler are equivalent to a Lanczos eigensolver, which implies that if the n eigenvalues of A are distinct, then the CG sample $\theta^n \sim N(A^{-1}b, A^{-1})$. The following theorem describes the results of the CG sampler in exact arithmetic when it terminates at iteration $k < n$.

Theorem 3.2 (Corollary 3.2 of Parker and Fox (2012)). If the CG sampler terminates at iteration k with $\|b - Ax^k\|_2 = 0$, then the CG sampler has successfully sampled from the k eigenspaces of A corresponding to the well separated eigenvalues $\{\lambda_1, \dots, \lambda_k\}$ of A . More specifically, if $\{w^i\}_{i=1}^k$ are the corresponding eigenvectors of A , then $(\text{var}(\theta^k|\theta^0, b) - A^{-1})v = 0$ for any $v \in \text{span}(w^1, \dots, w^k)$ and $\|\text{var}(\theta^k|\theta^0, b) - A^{-1}\|_2 = 1/\lambda^*$, where λ^* is the smallest eigenvalue of A such that $\lambda^* \notin \{\lambda_1, \dots, \lambda_k\}$.

Theorem 3.2 shows that the error in the variance of a CG sample is as large as the largest eigenvalue of A^{-1} associated with the eigenspaces not sampled. This result is a consequence of the action of the CG polynomial that reduces the error of the solver and the sampler. Put another way, when setting α_k and τ_k in (7) to the same values used by the CG solver, the resulting CG sampler is accelerated by the same CG polynomial (Parker and Fox 2012).

Like Chebyshev, CG polynomial solvers and samplers are guaranteed to accelerate stationary methods. The acceleration is even faster than Chebyshev because CG converges in a finite number of steps (Nocedal and Wright 2000; Parker and Fox 2012).

3.3. Sampling in Finite Precision

Numerical analysts have invested decades to develop a Chebyshev accelerated linear solver that provably converges geometrically in finite precision (Axelsson 1996). The Chebyshev accelerated sampler implementation in Fox and Parker (2014) is such an implementation. In all of the examples that Fox and Parker (2014, 2017) have studied using computationally expensive diagnostics, the Chebyshev accelerated samplers behave like the corresponding solvers, and converge with the predicted convergence rates in finite precision.

CG is a member of a class of Krylov methods that at the k th iteration, after initialization with a starting state x^0 , have traditionally been used to find a linear solution of $Ax = b$ in a *Krylov space* with basis $\{x^0, Ax^0, A^2x^0, \dots, A^{k-1}x^0\}$ (Meurant 2006). Lanczos methods are Krylov eigensolvers that find eigensolutions of A in the same Krylov space (Lanczos 1950). Lanczos's methods were adapted to sample from Gaussians by Schneider and Willsky (2003); Simpson, Turner, and Pettitt (2008); Aune, Eidsvik, and Pokern (2013); and Chow and Saad (2014). Like the CG sampler, these Lanczos samplers converge in a finite number of steps in exact arithmetic. Unfortunately, Lanczos methods may be challenging to implement for massive Gaussians because the states from all iterations must be either saved or recalculated in order to generate a sample. This is the same memory demanding and computationally intensive calculation that a Lanczos eigensolver must perform when determining eigenvectors of the matrix A (Meurant 2006; Saad 1992).

Relying on existing results from numerical linear algebra, all of the samplers described above are provably convergent in exact arithmetic (Sections 3.1 and 3.2). Unfortunately, provably convergent methods (whether linear solvers, eigensolvers, or samplers) in exact arithmetic do not always lead to convergent algorithms when implemented in finite precision (i.e., when implemented on a computer). All algorithms are affected by finite precision, some worse than others. There are many well-known examples of this phenomenon in numerical linear algebra. Notably, the Lanczos eigensolver is only able to estimate the eigenpairs of a matrix associated with well-separated eigenvalues before numerical instability makes further progress impossible without corrective measures (Meurant 2006).

Not surprisingly, in finite precision, the Krylov samplers (Aune, Eidsvik, and Pokern 2013; Chow and Saad 2014; Parker and Fox 2012; Simpson, Turner, and Pettitt 2008; Schneider

and Willsky 2003) appear to perform like a Lanczos eigensolver without correction. That is, while provably convergent in exact arithmetic, in finite precision, they effectively sample only from k of the eigenspaces of A after k iterations (Theorem 3.2) before numerical instability thwarts further progress. Among the eigenspaces not sampled, if the smallest eigenvalue of A is equal to $\lambda_{(\text{not sampled})}$, then when the Krylov sampler terminates at iteration k

$$\|A^{-1} - \text{var}(\theta^k | b, \theta^0)\|_2 \approx 1/\lambda_{(\text{not sampled})},$$

(Theorem 3.2). Schneider and Willsky (2003) implemented a potentially expensive corrective measure (i.e., reorthogonalization of the sampling directions) that allows a Lanczos algorithm to run longer in finite precision in order to converge to more of the eigenpairs of A , and also allows Krylov sampling from more of the eigenspaces of A . The preconditioning techniques applied by Chow and Saad (2014) actually seek to decrease the number of Lanczos sampler iterations by generating an approximation to $A^{1/2}z$ for $z \sim N(0, I)$ and using a residual stopping criterion. It is not clear whether their Gaussian sampler generates samples with the correct moments in finite precision.

Without corrective measures (e.g., reorthogonalization) or without a favorable spectrum (the small eigenvalues of A are well separated), Krylov samplers such as CG and Lanczos fail to produce either exact samplers or exact eigenproblem solutions due to finite precision.

4. A Fast Iterative Polynomial Accelerated Sampler

In this section, we present our new methodological contribution. First, we present a PCG sampler that is constructed by adding a single line of code to a PCG solver. Given the strengths and limitations of the stand-alone applications of the CG and Chebyshev samplers described in Section 3, our contribution is a synergistic implementation of the PCG and Chebyshev samplers.

4.1. PCG Accelerated Sampling

The following algorithm accelerates iterative sampling by the same PCG polynomial that a PCG solver uses. Although not immediately obvious, this algorithm can be written in the form (7) (Golub and Van Loan 1989, sec. 10.3.6). Removing the single line of code in Algorithm 1 that updates θ^k yields a PCG solver (see Saad 2003, Algorithm 9.2). Setting $M = C = I$ in Algorithm 1 yields the CG sampler presented in Parker and Fox (2012).

It is not necessary to factor $M = CC^T$ to implement a PCG sampler as might be suggested by Algorithm 1. Rather, one can implement a PCG sampler by starting with the PCG solver presented in one of Axelsson (1996), Golub and Van Loan (1989), and Nocedal and Wright (2000) that directly operate by M^{-1} instead of by C^{-1} and C^{-T} ; and add in the single line of code $\theta^k = \theta^{k-1} + \frac{z}{\sqrt{d_{k-1}}} p^{k-1}$. We focus on Algorithm 1 because the symmetric matrix splittings that we implement come naturally as $M = CC^T$. For example, the SSOR sampler of Roberts and Sahu (1997) can be implemented by the conventional forward sweep component-wise Gibbs sampler (4) with splitting $M_1 =$

Algorithm 1: Preconditioned Conjugate Gradient Accelerated Sampler of $N(A^{-1}b, A^{-1})$

input : SPD precision matrix A , $M = CC^T$ a symmetric splitting of A , maximum number of iterations k_{\max} , initial state θ^0 , b , and residual stopping criterion ϵ

output: $x^{k+1} \approx \mu = A^{-1}b$ and θ^{k+1} approximately distributed as $N(0, A^{-1})$

$x^0 = \theta^0$, $r^0 = C^{-1}(Ax^0 - b)$, $p^0 = -C^{-T}r^0$;

for $k = 1, \dots, k_{\max}$ **do**

$d_{k-1} = p^{(k-1)T}Ap^{k-1}$;

$\gamma_{k-1} = \frac{r^{(k-1)T}r^{k-1}}{d_{k-1}}$;

$x^k = x^{k-1} + \gamma_{k-1}p^{k-1}$;

$\theta^k = \theta^{k-1} + \frac{z}{\sqrt{d_{k-1}}}p^{k-1}$ for $z \sim N(0, 1)$;

$r^k = r^{k-1} + \gamma_{k-1}C^{-1}Ap^{k-1}$;

$\beta_k = \frac{r^{kT}r^k}{r^{(k-1)T}r^{k-1}}$;

$p^k = -C^{-T}r^k + \beta_k p^{k-1}$;

Check for convergence: quit if $\|r^k\| < \epsilon$;

end

$L + D$ (defined after (4)) and then a backward sweep with splitting $M_2 = M_1^T$. The resulting symmetric matrix splitting for SSOR sampling (and solving) is $M_{\text{SSOR}} = M_1M_1^T$. Any sampling scheme (4) (or solver (5)) for a matrix splitting M_1 can be implemented by forward and backward sweeps to generate a symmetric matrix splitting $M = M_1M_1^T$ to be used in Algorithm 1 with $C = M_1$. The key is to pick a splitting for which it is inexpensive to perform the operations by C^{-1} and C^{-T} in Algorithm 1.

In general, it is challenging to check whether an iterative sampler has converged in distribution. The PCG sampler, on the other hand, monitors the residual $\|Ax^k - b\|$ as a stopping criteria, just as does a linear solver. Chow and Saad (2014) and Simpson, Turner, and Pettitt (2008) used an approximate residual as a stopping criterion that monitors the distance of a current sample from $A^{-1/2}z$ where $z \sim N(0, I)$. As for CG and Lanczos solvers (Meurant 2006), a small residual at iteration k before numerical instability indicates that a CG or Lanczos sampler has effectively sampled from k of the eigenspaces of A^{-1} (Theorem 3.2).

Convergence of the PCG sampler is assured by viewing PCG as CG applied to the random vector $\tilde{\theta}^k = C^T\theta^k$. Theorem 3.2 shows that the PCG sampler successfully samples from k^* of the eigenspaces of $C^TA^{-1}C$ corresponding to the k^* well-separated eigenvalues of $C^{-1}AC^{-T}$. Hence, $\theta^k = C^{-T}\tilde{\theta}^k \sim N(0, A^{-1})$, the output of the PCG sampler, represents a sample from the corresponding k^* eigenspaces of A^{-1} . Preconditioners specific for CG and Lanczos sampling have been investigated by Schneider and Willsky (2003), Fox (2008), and Chow and Saad (2014).

4.2. PCG-Chebyshev Accelerated Sampling

We have seen that, in exact arithmetic, the PCG sampler is guaranteed to sample from $N(A^{-1}b, A^{-1})$ in a finite number of

steps (Theorem 3.2). But in finite precision, PCG fails to sample from the eigenspaces that do not correspond to the well-separated eigenvalues of A (Section 3.3). This is only a problem if the magnitude of the eigenvalues of A^{-1} associated with the excluded eigenspaces are large (Theorem 3.2). To capitalize on PCG's strengths (convergence in a finite number of steps to the eigenspaces corresponding to the well-separated eigenvalues), the sampler we propose first runs the PCG sampler. We “clean up” the resulting PCG sample by secondly running a Chebyshev sampler that does sample well in finite precision and has optimal geometric convergence rate. Interestingly, even for linear solvers, CG has been used to seed Chebyshev-accelerated deterministic iterations when there are multiple right-hand sides (Golub, Ruiz, and Touhami 2007). The resulting PCG-Chebyshev sampler is outlined in Algorithm 2.

Algorithm 2: PCG-Chebyshev Accelerated Sampler of $N(A^{-1}b, A^{-1})$

input : SPD precision matrix A , $M = M_1M_1^T$ where M_1 is a matrix splitting of A , initial state θ^0 , b , and initial estimate of x^0 of $A^{-1}b$, PCG residual stopping criterion ϵ_{PCG} , maximum number of PCG iterations k_{PCG} , number of Chebyshev iterations k_{Cheby}

output: $\theta \sim N(A^{-1}b, A^{-1})$ and $x \approx A^{-1}b$

PCG sampling

input : θ^0, x^0, A , split preconditioner $C = M_1$,

$\epsilon = \epsilon_{\text{PCG}}, k_{\max} = k_{\text{PCG}}$

output: $\theta_{\text{PCG}} \sim N(0, A^{-1})$, $x_{\text{PCG}} \approx A^{-1}b$ and $\{\gamma_k, \beta_k\}$
Implement Algorithm 1, get approximate solution x^{k+1} and approximate sample θ^{k+1} ;

end

Get the extreme eigenvalues of $M^{-1}A$ from $\{\gamma_k, \beta_k\}$ using the prescription in (Parker and Fox (2012), Lemma 2.1);

Chebyshev sampling

input : Number of sampler iterations k_{Cheby} ,

$\theta^0 = \theta_{\text{PCG}}, x^0 = x_{\text{PCG}}, b_{\text{Cheby}} = 0$, extreme eigenvalues of $M_{-1}A$

output: $\theta_{\text{Cheby}} \sim N(0, A^{-1})$

Run Algorithm 3 of Fox and Parker (2014) for k_{Cheby} iterations. At the k_{Cheby} th iteration, get approximate sample $\theta_{\text{Cheby}}^{k+1}$;

end

$\theta = \theta_{\text{Cheby}} + x_{\text{PCG}}$ and $x = x_{\text{PCG}}$;

In addition to sampling from the k eigenspaces of A^{-1} corresponding to the k well-separated eigenvalues of $M_{-1}^{-1}AM_{-1}^{-T}$ (by seeding θ_{PCG} into Chebyshev), Algorithm 2 makes clear that the PCG sampler also accomplishes two other crucial tasks

1. The PCG sampler, with preconditioner equal to the splitting matrix $M = M_1M_1^T$, provides an avenue to estimate the extreme eigenvalues of $M^{-1}A$ that are required by Chebyshev. Strictly speaking, a $k \times k$ tridiagonal matrix is built from the PCG parameters $\{\gamma_k, \beta_k\}$. The extreme eigenvalues of this tridiagonal, found at a negligible k^2 flops when $k \ll n$, are the required extreme eigenvalues of $M_{-1}^{-1}AM_{-1}^{-T}$ or equivalently, of $M^{-1}A$.

2. PCG provides an estimate of the mean $\mu = A^{-1}b \neq 0$. The PCG sampler is used to perform the mean calculation because PCG is a faster linear solver than Chebyshev and will find μ after a finite number of iterations. Put another way, the Chebyshev sampler can sample from $N(0, A^{-1})$ much faster (i.e., after only k_{Cheby}^{**} iterations with convergence rate ρ_{Cheby}^2) compared to sampling from $N(\mu \neq 0, A^{-1})$ that requires $k_{\text{Cheby}}^* > k_{\text{Cheby}}^{**}$ iterations with convergence rate $\rho > \rho^2$) see [Section 3.2.1](#).

Acceptance of the PCG sample $\theta_{\text{PCG}} \sim N(0, A^{-1})$ as an initialization into the Chebyshev sampler further reduces the geometric convergence rate by a constant factor, according to [Theorem 3.1](#): $\text{var}(\theta^k) = A^{-1} + P_k(\text{var}(\theta_{\text{PCG}}) - A^{-1})P_k^T$. That is, Chebyshev converges faster the better that $\text{var}(\theta_{\text{PCG}})$ approximates A^{-1} .

4.3. Implementation Details

For each frame in the video, we procure samples $(\theta, \sigma^2, \lambda)$ from (2) by performing 10^4 iterations of the following: sample $(1/\sigma^2, \lambda)|(y, \theta)$ using a product of Gammas; then sample $\theta|(y, \sigma^2, \lambda)$ from the Gaussian (3) using the PCG-Chebyshev sampler (Algorithm 2) implemented in Matlab. $M = M_1 M_1^T$ in the PCG-Chebyshev sampler is set so that M_1 is the lower triangular matrix splitting defined after (5) (i.e., M implements forward and backward sweeps of a component-wise Gibbs sampler). When analyzing the first image in the video, the initialization for the sampler was $\theta^0 = 1/2y + 1/2\bar{y}\mathbf{1}$, where \bar{y} is a scalar value equal to the mean surface thickness in y and $\mathbf{1}$ is a 512^2 vector of 1's. For each subsequent image in the video, initialization for the sampler was $\theta^0 = 1/2y + 1/2\hat{\theta}_{\text{pre}}$, where $\hat{\theta}_{\text{pre}}$ is the Bayesian estimate for the previous image in the video. Our experience confirms the theory ([Theorem 3.1](#)) that shows that while the convergence rate is the same for any initial condition, the number of iterations is adversely affected, for solvers and samplers, by a poor starting choice. For example, using white noise $\theta^0 \sim N(0, I)$ is a terrible initialization, resulting in a dismal reconstruction of the biofilm's surface even after substantial error reductions of 10^{-8} or more because of the large initial errors $\|E(\theta^0) - A^{-1}b\|$ and $\|\text{var}(\theta^0) - A^{-1}\|$. We considered other starting choices for a few frames, for example, $\theta^0 = 1/2y + 1/2\bar{y}\mathbf{1}$, with no discernable impact on the surface reconstructions.

Half of the 10^4 iterations were considered burn-in (Gelman et al. 1995) and hence for each image Markov chains of length $10^4/2$ were used to estimate the posterior in (2). For each image, in each run of the 10^4 "outer-iterations" that generated a state $(\theta, \sigma^2, \lambda)$ in the Markov chain, the Chebyshev component of the PCG-Chebyshev sampler ran for k_{Cheby} "inner iterations" to sample $\theta|(y, \lambda, \sigma^2)$ from (3). We set the PCG residual stopping criterion of $\epsilon = 10^{-4}$ and the maximum number of PCG iterations to $k_{\text{PCG}} = 10^3$ which we have found works well for procuring acceptable eigenvalue estimates. These were the same criteria used for the stand-alone CG sampler implementation. The number of Chebyshev iterations was set according to $k_{\text{Cheby}} = \min(\max(10, k_{\text{Cheby}}^{**}), 100)$, where k_{Cheby}^{**} is the number of iterations calculated by (10) in order to attain an error reduction in variance of $\epsilon = 10^{-8}$. For early outer-iterations

during burn-in, k_{Cheby}^{**} was sometimes larger than 100 because the convergence rate ρ_{Cheby} (see (8)) was close to 1; in these instances, k_{Cheby} was set to 100. But for later outer-iterations, especially after burn-in, k_{Cheby}^{**} was typically less than 10 corresponding to cases when $\rho_{\text{Cheby}} < 0.1$. Nonetheless, Chebyshev was always run a minimum number of $k_{\text{Cheby}} = 10$ iterations that assured a minimum reduction in the variance error of $\epsilon = 10^{-8}$.

The main computational costs of the iterative samplers are the matrix-vector multiplication by A and the forward solve to implement multiplication by M^{-1} when generating the next sample θ^{k+1} in (7). The cost of matrix multiplication is about $2n^2$ flops for a dense precision matrix and is reduced to about $(2n_A - 1)n$ flops for a sparse precision matrix A , such as we consider here, that has only $n_A = 5$ nonzero elements per row. The cost of a forward or backward solve using a triangular M_1 is n^2 flops for dense matrices and $(2n_M - 1)n$ flops for sparse M_1 with $n_M = 3$ nonzero elements per row (Watkins 2002). The stand-alone PCG and Chebyshev samplers each multiply by A and operate by M_1^{-1} and M_1^{-T} in each iteration. Hence, each iteration of the PCG-Chebyshev sampler costs at most $(19k_{\text{PCG}} + 19k_{\text{Cheby}})n$ flops. A Cholesky factorization, on the other hand, costs about b^2n flops, where $b = \sqrt{n} = 512$ is the bandwidth of the precision matrix A that we consider, regardless of the sparseness of the matrix (Watkins 2002). Hence, the PCG sampler on this 2D problem will be less expensive than Cholesky as long as the total number of sampler iterations is less than $b^2/19 = 1.4 \times 10^4$. We will see (in [Figure 3](#)) that the Cholesky factorization incurs 10 times more operations and more CPU time for the 2D biofilm surface problem that we present. Iterative samplers are expected to outperform Cholesky even more when solving inverse problems using 3D Gaussian fields for which the posterior precision matrix has bandwidths $b \approx n$ or more (see, e.g., Fox and Parker 2017).

5. Results

[Figure 3](#) compares results from the PCG-Chebyshev sampler (Algorithm 2) to a Cholesky sampler and a stand-alone CG sampler (Algorithm 1 with $M = I$). For a single image from the video, the posterior mean estimates of the biofilm surface for these three samplers are depicted in the first row of the figure. The estimated biofilm surface ($\hat{\theta}$), the associated uncertainty (assessed via the point-wise standard deviation over the 512×512 lattice across the samples), and variance parameter estimates ($\hat{\sigma}$ and $\hat{\lambda}$) for the PCG-Chebyshev sampler are similar to results for the Cholesky sampler as predicted by the theory ([Section 3](#)).

When CG is used by itself, [Figure 3](#) shows that the estimated surface is over-smoothed, and the uncertainty associated with the estimated surface is vastly underrepresented. This is due to the known finite precision issues with the CG sampler ([Section 3.3](#)). Over-smoothed samples have been noted previously when CG sampling with a Laplacian prior precision (Parker and Fox 2012) or when purposely terminating CG early (Feron, Orioux, and Giovannelli 2016; Wikle et al. 2001). The over-smoothing in the CG samples can, when compared to the actual samples, be quantified by the eigenspaces of the posterior precision matrix that the CG sampler successfully sampled from ([Theorem 3.2](#)).

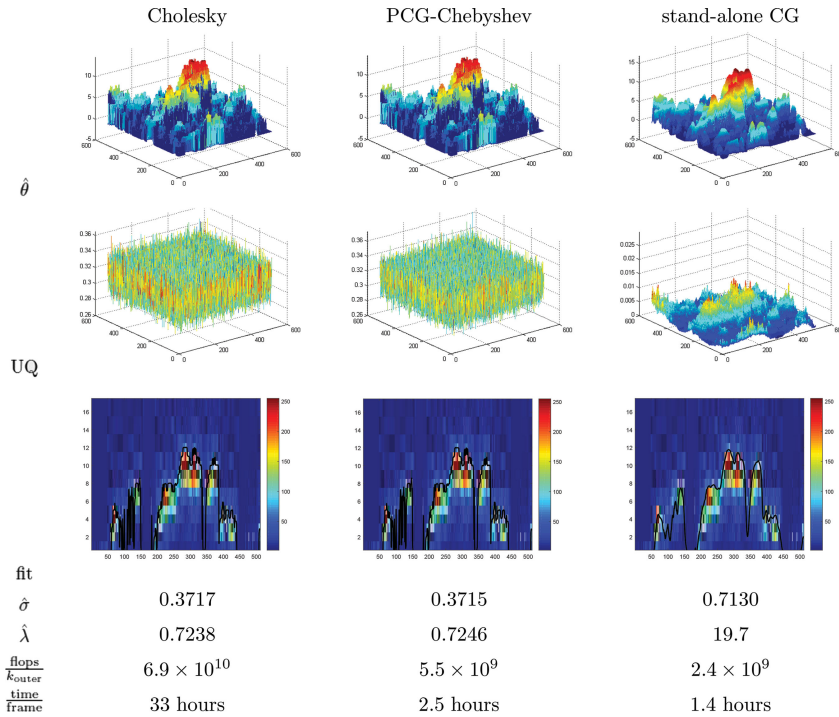


Figure 3. Comparison of three different samplers on the first frame of the video that we analyze. $\hat{\theta}$ row: Posterior mean estimates of the true biofilm surface. In each pane in the row, the x -axis is the left horizontal axis, the y -axis is the right horizontal axis. Axes are in pixels, where the distance between xy -pixels is $1.2 \mu\text{m}$, and the distance between vertical pixels is $7 \mu\text{m}$. UQ row: Uncertainty quantification of the estimate $\hat{\theta}$ with a standard deviation calculated at each xy location across the samples. fit row: An yz cross-sectional view of the intensities in the raw CM data at $x = 280$. The black curve shows the fit of the posterior mean estimate to the CM intensities. $\hat{\sigma}$ row: Posterior median estimate of the standard deviation of the biofilm surface measurement process. $\hat{\lambda}$ row: Posterior median estimate of the prior precision that controls smoothing. flops/ k_{outer} row: maximum number of floating point operations to generate a single Gaussian sample via (3) when processing the image. $\frac{\text{time}}{\text{frame}}$ row: actual time to process this image over 10^4 outer iterations using a Matlab implementation of the samplers on THE BEAST, a Xeon X5690 with a 3.47 GHz processor and 11GB of RAM.

These eigenspaces represent the low frequency components of the image. Figure 3 (in the “fit” row) shows an example of the graphical technique that we use to assess model fit at a single slice through the 3D data, although in practice the assessment includes similar plots over multiple xz and yz slices. In this case, acceptable fit of the reconstruction $\hat{\theta}$ to the imaging data is shown in the yz cross-sectional slice shown except perhaps between $90 \leq y\text{-pixels} \leq 130$; here, it appears that the Cholesky and PCG-Chebyshev samplers overfit to the data (i.e., the estimated surface exhibits high frequency). To impose more smoothing when using the PCG-Chebyshev sampler, we could use a more informative prior over larger values of the smoothing parameter λ .

One advantage of a sampling approach to statistical inference is that, once samples of the biofilm surface are procured from the posterior (2), we can calculate whatever function of the surface samples we like, whether linear or nonlinear, thereby constructing a representation of the posterior of the corresponding parameter. We calculated a volume for each sample given a biofilm surface y . This volume is a sample from the marginal posterior $\pi(\text{volume}|y)$. Using this approach, we estimated the biofilm volume with 99% credible intervals from 40 frames (i.e., about 10 min) of the video (see Figure 4). These 40 frames capture the response of the biofilm as the salt water is removed (before frame 5) and then applied again (after frame 31). Application of the salt water treatment is associated with a 26% reduction in volume. A 99% credible interval for this reduction is (25%, 27%). Such reduction calculations are the norm when assessing the efficacy of antimicrobial treatments. Figure 2

shows the biofilm surface before (frame 31) and after (frame 35) the application of the treatment. As a comparison, the biased volumes calculated by counting bright pixels (i.e., in this analysis, pixels with intensity values larger than 49) is also presented.

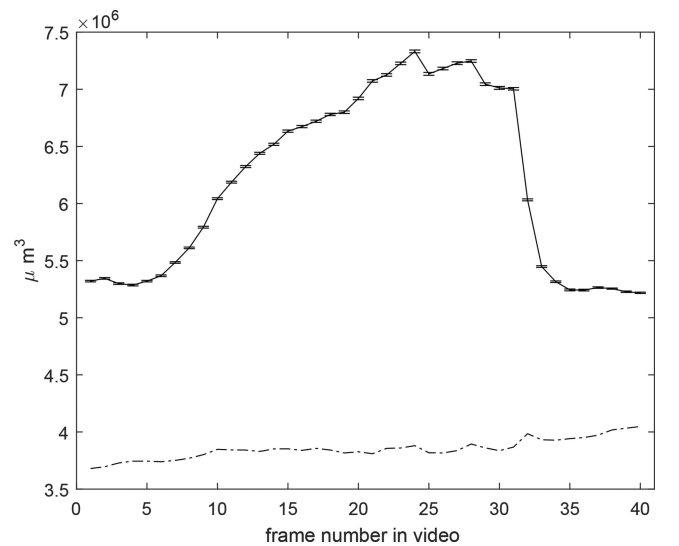


Figure 4. Posterior mean estimates of the volume for 40 frames (about 10 min) of the video are indicated by the solid line. These 40 frames capture the response of the biofilm as the salt water treatment is removed and then reapplied. Error bars indicate 99% credible intervals. Volume is underestimated by an estimator that only counts bright pixels as indicated by the dash-dotted line. The salt water treatment was applied sometime between frames 31 and 32 that is associated with a large drop in volume between frames 31 and 33.

6. Discussion and Conclusion

For the first time, we apply a PCG-Chebyshev accelerated iterative sampler (Algorithm 2) to efficiently solve the Gaussian step in a Bayesian linear inverse problem. There are more efficient ways to sample the variances (σ^2 and $1/\lambda$) than the conditional sampler that we show here (Agapiou et al. 2014; Feron, Orioux, and Giovannelli 2016; Fox and Norton 2016). Fixed on values of the variances, to our knowledge, the PCG-Chebyshev implementation is the fastest and most memory efficient sampler from a LARGE Gaussian with arbitrary variance structure.

This is also the first time that the drastic attenuation of the CM laser intensity into thick biofilms has been quantitatively addressed (Pitts and Stewart 2008). The artifact of attenuation or “shadowing” is fairly typical in our experience when imaging biofilms with a CM. The thickness of the biofilm that can be viewed satisfactorily from top to bottom in the z -dimension without this artifact appears to depend upon the density and composition of the particular sample. Given this artifact, we do not consider fluorescence microscopy (confocal or otherwise) to be the technique of choice for measuring biofilm thickness or examining stratification of activity with depth. Instead, cryoembedding and cryosectioning (Figure 1B), or optical coherence tomography (OCT) (Figure 1C) can be used. Cryoembedding and sectioning involves freezing the biofilm in standard tissue embedding medium, cutting 5 μm thick cross-sections through the sample on a cryostat, and placing those sections flat on a microscope slide. Sections can then be viewed using widefield fluorescence microscopy or CM, which eliminate any top-down viewing artifact. OCT is a relatively new addition to biofilm imaging techniques, but the method has been used widely in ophthalmology and in industry for at least 20 years. OCT is an interferometric technique, where an infrared laser is incident upon a sample, and the reflected light is compared to a reference beam to provide an image of a sample in a manner similar to an ultrasound. OCT does not use fluorescence and has a penetration depth on the order of 1 mm in biofilms. In general, all three methods (in Figure 1) are used in concert to provide a robust, fully dimensional picture of a biofilm that includes information regarding thickness, topography, stratification of activity, structure, and function. While confocal microscopy and OCT enable in situ, fully hydrated imaging, only cryosectioning provides fluorescence data that is free of the top-down imaging artifact that we illustrate here.

Previous Bayesian analyses of CM images of thin layers of human cells considered less severe attenuation effects (Al-Awadhi, Hurn, and Jennison 2011). Our approach was inspired by a desire to accurately calculate biofilm volumes from CM images with an associated measure of uncertainty. Fitting a surface the way we do is simple and subsequent samples from the posterior can be generated quickly using the PCG-Chebyshev accelerated sampler. This approach presumes that the precision of the CM’s identification of the top edge of the biofilm is small compared to the variability of the surface across the entire biofilm. The disadvantage of this surface model is that the data have been manipulated from pixels in 3D to a surface in 2D. Hence, this model cannot reconstruct holes or overhanging features in the biofilm (e.g., Figure 1B). We are developing a more computationally demanding nonlinear approach that

deals with these issues. Perhaps most importantly, the nonlinear approach does not require thresholding, a very common step of CM data preprocessing by today’s microscopists. Future work also includes developing a more computationally demanding framework that directly models the temporal relationship of the frames in the video (see, e.g., Higdon 2006).

These results demonstrate the dramatic osmotic response of a biofilm to a targeted treatment. This behavior is not widely appreciated among biofilm researchers and merits further exploration. Is the biofilm more vulnerable under osmotic stress? Could manipulation of the osmotic response be paired with an antimicrobial treatment to better kill or remove biofilms? Our analyses show that the salt water treatment is associated with a statistically significant 26% reduction in volume. For manufacturers of antimicrobials, quantifiable reductions of microbial abundances are crucial to bring products to market, convince consumers to buy them, and positively affect human health. Our future work will focus on determining how the surface representations of biofilms presented in this article, and the reductions of biofilm volumes in other scenarios, might be used to predict reductions of biofilm microbial abundances.

This work helps us to begin to answer the most frequently asked questions that we receive regarding CM experimental design. Because increased pixel resolution (set by the user) decreases temporal resolution (i.e., it takes the CM more time to capture more pixel data), microscopists want to know: How many xy pixels should be used in each planar z -slice? How many z -slices should be collected in the vertical dimension? How many different fields of view should be collected? These questions pertain to the precision of the volume estimate across the entire 1.27 cm^2 surface from which the 620 $\mu\text{m} \times 620 \mu\text{m}$ image was collected. This work can begin to answer the first two questions regarding pixelation: If one wishes to use the CM solely as a half-a-million dollar estimator of volumes, then our analyses suggest that the pixel resolution is much too fine because the error bars, that is, the 99% credible intervals in Figure 4 are extremely tight. The results presented here provide a first step toward the application of Bayesian experimental design techniques (e.g., see Solonen, Haario, and Laine 2012) that will quantify how much less spatial resolution in CM images is allowed before the uncertainty in biofilm volumes, or some other imaging outcome, becomes too large. Based on our experience with other techniques that provide quantitative assessments of biofilms, we expect that the most important level of replication is to collect CM data from multiple independent environmental sites or experiments. In the latter setting, biofilms are grown independently with different inocula on different days in each experiment.

CG and Chebyshev samplers have been applied as stand-alone samplers for Bayesian problems before. Fox and Parker (2017) applied a Chebyshev sampler to refine the pixelation of CM images by Bayesian interpolation. Gilavert and Moussaoui (2015) apply CG for linear Bayesian image reconstruction. They clean up CG’s possible poor performance in finite precision by instituting a Metropolis Hastings step. Bardsley et al. (2012) applied the CG sampler within the ensemble Kalman filter and showed improved performance compared to other ensemble filter implementations. Feron, Orioux, and Giovannelli (2016)

consider a linear Bayesian model with the same variance structure as we do, but, for every draw of the variance parameters, they implement a CG sampler with only a small number of jittered CG search directions to attain a Markov chain that is provably convergent in exact arithmetic. Unfortunately, that article contains no specification of a convergence rate and its performance in finite precision is unknown.

Two other promising methods directly adapt any solver to the task of iterative sampling. Conditioned on values for the variances (i.e., Σ_y and $\frac{1}{\lambda}W^{-1}$ in (1)), the method of randomized maximum likelihood (RML) solves a linear Bayesian inverse problem with a Gaussian likelihood (Chen and Oliver 2012). Randomize-then-optimize (RTO) is the extension of RML to nonlinear problems (Bardsley et al. 2014) with a Gaussian likelihood. At each iteration, these algorithms jitter the data using Gaussian noise with variance Σ_y then perform a nonlinear least squares optimization step that generates a sample from the posterior. The randomization step can easily be effected for the common case where the variance of the likelihood is $\Sigma_y = \sigma^2 I$ (as for the example in this article); but for general Σ_y in large problems, the randomization step would require that either Σ_y be factored or that a method such as introduced here be applied. Another potentially limiting issue for large problems is that RML and RTO require a factorization of the prior precision λW .

We suggest that PCG-Chebyshev is the current state-of-the-art iterative sampler from a LARGE Gaussian with an unstructured precision matrix that does not require any (precision or covariance) matrix factorization and has minimal memory requirements (only vectors from two previous iterations need to be saved). Our methodological contribution in this work is to present a two-phase PCG-Chebyshev iterative sampler that harnesses CG's ability to converge in a finite number of steps when the spectrum of A is favorable (i.e., the small eigenvalues are well separated). For covariance matrices with less favorable spectra, where CG may fail to converge satisfactorily, the Chebyshev sampler has an optimal geometric convergence rate and reliably samples in finite precision. Because Krylov methods like CG are the current state-of-the-art for linear solvers, we expect work to continue to obtain a truly iterative (i.e., only requires a few iteration's worth of information) Krylov sampler that converges to the full Gaussian target in a finite number of steps in theory (exact arithmetic) and in practice (finite precision).

Supplementary Materials

An avi file that contains a video of the image data that we analyze is provided.

Acknowledgments

The authors thank Professor Ross Carlson and Dr. Kris Hunt for access to THE BEAST for running simulations.

Funding

The authors gratefully acknowledge NIH award GM109452 and the industrial associates of the Center for Biofilm Engineering for support.

References

- Adler, S. L. (1981), "Over-Relaxation Method for the Monte Carlo Evaluation of the Partition Function for Multiquadratic Actions," *Physical Review D*, 23, 2901–2904. [4]
- Agapiou, S., Bardsley, J., Papaspiliopoulos, O., and Stuart, A. (2014), "Analysis of the Gibbs Sampler for Hierarchical Inverse Problems," *SIAM Journal on Uncertainty Quantification*, 2, 511–544. [10]
- Al-Awadhi, F., Hurn, M., and Jennison, C. (2011), "Three-Dimensional Bayesian Analysis and Confocal Microscopy," *Journal of Applied Statistics*, 38, 29–46. [10]
- Amit, Y., and Grenander, U. (1991), "Comparing Sweep Strategies for Stochastic Relaxation," *Journal of Multivariate Analysis*, 37, 197–222. [4]
- Aune, E., Eidsvik, J., and Pokern, Y. (2013), "Iterative Numerical Methods for Sampling from High Dimensional Gaussian Distributions," *Statistics and Computing*, 23, 501–521. [6]
- Axelsson, O. (1996), *Iterative Solution Methods*, Cambridge: Cambridge University Press. [4,5,6]
- Bardsley, J., Solonen, A., Haario, H., and Laine, M. (2014), "Randomize-then-Optimize: A Method for Sampling from Posterior Distributions in Nonlinear Inverse Problems," *SIAM Journal on Scientific Computing*, 36, A1359–C399. [11]
- Bardsley, J., A. Solonen, A. Parker, Haario, H., and Howard, M. (2013). "An Ensemble Kalman Filter Using the Conjugate Gradient Sampler," *International Journal of Uncertainty Quantification*, 3, 357–370. [10]
- Bardsley, J. M. (2012), "MCMC-Based Image Reconstruction with Uncertainty Quantification," *SIAM Journal on Scientific Computing*, 34, A1316–A1332. [2,3]
- Barone, P., and Frigessi, A. (1990), "Improving Stochastic Relaxation for Gaussian Random Fields," *Probability in the Engineering and Informational Sciences*, 23, 2901–2904. [4]
- Calvetti, D., and Somersalo, E. (2007), *Introduction to Bayesian Scientific Computing*, New York: Springer. [3]
- Chen, Y., and Oliver, D. (2012), "Ensemble Randomized Maximum Likelihood Method as an Iterative Ensemble Smoother," *Mathematical Geosciences*, 44, 1–26. [11]
- Chow, E., and Saad, Y. (2014), "Preconditioned Krylov Subspace Methods for Sampling Multivariate Gaussian Distributions," *SIAM Journal on Scientific Computing*, 36, A588–A608. [6,7]
- Feron, O., Orioux, F., and Giovannelli, J. (2016), "Gradient Scan Gibbs Sampler: An Efficient Algorithm for High Dimensional Gaussian distributions," *IEEE Journal Of Selected Topics In Signal Processing*, 10, 343–352. [8,10]
- Fox, C. (2008), "A Conjugate Direction Sampler for Normal Distributions, with a Few Computed Examples," Technical Report 2008-1, Electronics Group, University of Otago. [7]
- Fox, C., and Norton, R. (2016), "Fast Sampling in Linear-Gauss Inverse Problems," *SIAM/ASA Journal on Uncertainty Quantification*, 4, 1191–1218. [10]
- Fox, C., and Parker, A. (2014), "Convergence in Variance of Chebyshev Accelerated Gibbs Samplers," *SIAM Journal on Scientific Computing*, 36, A124–A147. [5,6]
- (2017), "Accelerated Gibbs Sampling of Normal Distributions Using Matrix Splittings and Polynomial Acceleration," *Bernoulli*, 23, 3711–3743. [4,5,6,8,10]
- Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995), *Bayesian Data Analysis*, New York: Chapman & Hall. [3,8]
- Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions Pattern Analysis and Machine Intelligence*, 6, 721–741. [3]
- Gilavert, C., and Moussaoui, S. (2015), "Efficient Gaussian Sampling for Solving Large-Scale Inverse Problems Using MCMC," *IEEE Transactions on Signal Processing*, 63, 70–80. [10]
- Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1996), "Introducing Markov Chain Monte Carlo," in *Markov Chain Monte Carlo in Practice*, eds. W. R. Gilks, S. Richardson, and D. Spiegelhalter, New York: Chapman & Hall, pp. 1–16. [3]
- Golub, G. H., Ruiz, D., and Touhami, A. (2007), "A Hybrid Approach Combining Chebyshev Filter and Conjugate Gradient for Solving Linear

- Systems with Multiple Right Hand Sides,” *SIAM Journal of Matrix Analysis and Applications*, 29, 774–795. [7]
- Golub, G. H., and Van Loan, C. F. (1989), *Matrix Computations* (2nd ed.), Baltimore, MD: The Johns Hopkins University Press. [4,5,6]
- Hall-Stoodley, L., Costerton, J., and Stoodley, P. (2004), “Bacterial Biofilms: From the Natural Environment to Infectious Diseases,” *Nature Reviews Microbiology*, 2, 95–108. [1]
- Hestenes, M. R., and Stiefel, E. (1952), “Methods of Conjugate Gradients for Solving Linear Systems,” *Journal of Research of the National Bureau of Standards*, 49, 409–436. [4]
- Heydorn, A., Nielsen, A. T., Hentzer, M., Sternberg, C., Givskov, M., Ersboll, B. K., and Molin, S. (2000), “Quantification of Biofilm Structures by the Novel Computer Program COMSTAT,” *Microbiology*, 146, 2395–2407. [2]
- Higdon, D. (2006), “A Primer on Space-Time Modelling from a Bayesian Perspective,” in *Statistics of Spatio-Temporal Systems*, eds. B. Finkenstadt, L. Held, and V. Isham, New York: Chapman & Hall/CRC, pp. 217–279. [2,3,10]
- Lanczos, C. (1950), “An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators,” *Journal of Research of the National Bureau of Standards*, 45, 255–282. [6]
- Lewandowski, Z., and Beyenal, H. (2014), *Fundamentals of Biofilm Research*, Boca Raton, FL: CRC Press. [2]
- Meurant, G. (2006), *The Lanczos and Conjugate Gradient Algorithms*, Philadelphia: SIAM. [6,7]
- Nocedal, J., and Wright, S. J. (2000), *Numerical Optimization*, New York: Springer. [6]
- Parker, A., and Fox, C. (2012), “Sampling Gaussian Distributions in Krylov Spaces with Conjugate Gradients,” *SIAM Journal on Scientific Computing*, 34, B312–B334. [5,6,8]
- Pitts, B., and Stewart, P. (2008), “Confocal Laser Microscopy on Biofilms: Successes and Limitations,” *Microscopy Today*, 16, 18–23. [10]
- Roberts, G. O., and Sahu, S. (1997), “Updating Schemes, Correlation Structure, Blocking and Parameterization for the Gibbs Sampler,” *Journal of the Royal Statistical Society, Series B*, 59, 291–317. [4]
- Rue, H., and Held, L. (2005), *Gaussian Markov Random Fields: Theory and Applications*, New York: Chapman Hall. [2,3]
- Saad, Y. (1992), *Numerical Methods for Large Eigenvalue Problems*, Manchester: Manchester University Press. [5,6]
- (2003), *Iterative Methods for Sparse Linear Systems* (2nd ed.), SIAM. [4,5,6]
- Saad, Y., and van der Vorst, H. A. (2000), “Iterative Solution of Linear Systems in the 20th Century,” *Journal of Computational and Applied Mathematics*, 123, 1–33. [4]
- Schneider, M. K., and Willsky, A. S. (2003), “Krylov Subspace Method for Covariance Approximation and Random Processes and Fields,” *Multidimensional Systems and Signal Processing*, 14, 295–318. [6,7]
- Sheppard, C. J. R., and Shotton, D. R. (1997), *Confocal Laser Scanning Microscopy*, New York: Garland Science. [2,3]
- Simpson, D. P., Turner, I. W., and Pettitt, A. N. (2008), “Fast Sampling from a Gaussian Markov Random Field Using Krylov Subspace Approaches,” Technical Report 14376, School of Mathematical Sciences, Queensland University of Technology, Brisbane, Australia. [6,7]
- Solonen, A., Haario, H., and Laine, M. (2012), “Simulation-Based Optimal Design Using a Response Variance Criterion,” *Journal of Computational and Graphical Statistics*, 21, 234–252. [10]
- Stewart, P. (2015), “Antimicrobial Tolerance in Biofilms,” *Microbiology Spectrum*, 3. [1]
- Watkins, D. (2002), *Fundamentals of Matrix Computations* (2nd ed.), New York: Wiley. [8]
- Wikle, C. K., Milliff, R. F., Nychka, D., and Berliner, M. (2001), “Spatiotemporal Hierarchical Bayesian Modeling: Tropical Ocean Surface Winds,” *Journal of the American Statistical Association*, 96, 382–397. [8]
- Young, D. M. (1971), *Iterative Solution of Large Linear Systems*, New York: Academic Press. [4,5]