A Scalable Method to Exploit Screening in Gaussian Process Models with Noise

Christopher J. Geoga and Michael L. Stein
Department of Statistics, Rutgers University, New Brunswick, NJ

ABSTRACT
A common approach to approximating Gaussian log-likelihoods at scale exploits the fact that precision matrices can be well-approximated by sparse matrices in some circumstances. This strategy is motivated by the screening effect, which refers to the phenomenon in which the linear prediction of a process $Z$ at a point $x_0$ depends primarily on measurements nearest to $x_0$. But simple perturbations, such as iid measurement noise, can significantly reduce the degree to which this exploitable phenomenon occurs. While strategies to cope with this issue already exist and are certainly improvements over ignoring the problem, in this work we present a new one based on the EM algorithm that offers several advantages. While in this work we focus on the application to Vecchia’s approximation (Vecchia, a particularly popular and powerful framework in which we can demonstrate true second-order optimization of M steps, the method can also be applied using entirely matrix-vector products, making it applicable to a very wide class of precision matrix-based approximation methods. Supplementary materials for this article are available online.

1. Introduction
Gaussian process models are increasingly important in a broad variety of fields. In spatial statistics, they often arise via process models like

$$Y(x) = Z(x) + \varepsilon(x),$$

where $Z$ is assumed to be a Gaussian process with nontrivial dependence structure and $\varepsilon$, typically independent of $Z$, is Gaussian white noise or something with a comparably simple covariance structure. Throughout this work, we will assume that $Z$ is mean-zero, although extensions to linear mean models would be straightforward. When measured at a finite number of locations $\{x_j\}_{j=1}^n$, say, this corresponds to the distributional model for $z = \{Z(z_j)\}_{j=1}^n$ and $\varepsilon$ defined similarly given by $y = z + \varepsilon \sim \mathcal{N}(0, \Sigma + R)$, where $\Sigma$ is the covariance matrix for $z$ and $R$ denotes the covariance matrix of $\varepsilon$, with a prototypical example of $R = \eta^2 I$, in which case $\varepsilon$ is often called a nugget in the geostatistical literature. A particularly common modeling strategy is to write a parametric model $K_{\theta} (\cdot, \cdot)$ for the covariance function of $Z$, so that $\Sigma_{j,k} (\theta) = \text{cov}(Z(x_j), Z(x_k)) = K_{\theta} (x_j, x_k)$, and then to numerically optimize the negative log-likelihood with respect to parameters $\theta$, where the parameter set $\theta$ includes any parameters used in $R$ as well.

When the data size $n$ is sufficiently large, operations with the typically dense covariance matrix $\Sigma$ can be prohibitively expensive. Many direct matrix approximations have been applied to this problem in the past in order to cope with this issue, such as low-rank approximations (Cressie and Johannesson 2008), matrix tapering (Kaufman, Schervish, and Nychka 2008) (one of the few approximation paradigms to have supporting theory), implicit methods that employ fast algorithms like the fast multipole method (FMM) and its descendants (Anitescu, Chen, and Wang 2012), hierarchical matrices (Ambikasaran et al. 2016; Litvinenko et al. 2019; Geoga, Anitescu, and Stein 2020; Chen and Stein 2021), and many others (to say nothing of dynamical approximations that are less directly matrix-oriented (Wikle and Cressie 1999; Stroud, Müller, and Sansó 2001; Katzfuss and Cressie 2012)). In several of these strategies, adding a diagonal or otherwise sufficiently structured perturbation is not problematic. But direct covariance matrix-space methods come with significant drawbacks, such as the difficulty of ensuring that approximations for $\Sigma$ are positive definite while maintaining flexibility (see Chen and Stein (2021) for a notable exception, however). Not only is this necessary in the sense of being required for a valid distributional model, it is practically required because the Gaussian log-likelihood contains a log-determinant. And while this issue can be avoided by solving the score equations instead (see Stein, Chen, and Anitescu 2013) or using stochastic estimators for the log-determinant based on Lanczos quadrature (see, Ubaru, Chen, and Saad (2017), or the growing field of “Bayesian optimization” (Mockus 2012), these estimators can have difficulty being accurate enough for optimization.

The other primary thrust in approximating $\Sigma$ is to find sparse approximations to $\Sigma^{-1}$. Perhaps surprisingly, it is in some sense easier to obtain valid positive-definite matrices with this approach than it is with the hierarchical matrix approach, in part because they are often based on valid process approximations. A good example is the class of so-called Vecchia approximations (Vecchia 1988), which will be the specific methods that we will be using to showcase our estimation algorithm in...
this work, which effectively apply (compound-) Markovian-like assumptions that induce conditional independence. Vecchia’s approximation exploits the fact that any multivariate density can be expanded in terms of conditional densities, in particular that \( p(y_1, y_2, \ldots, y_n) = p(y_1) \prod_{i=2}^{n} p(y_i | y_1, \ldots, y_{i-1}) \), and attempts to approximate this quantity by instead conditioning on a subset of points that hopefully are of similar predictive power as all of the past observations, so that \( p(y_1, y_2, \ldots, y_n) \approx p(y_1) \prod_{i=2}^{n} p(y_i | y_{\sigma(i)}) \), where \( \sigma(j) \subseteq [1 \ldots i-1] \) and is typically \( O(1) \) in size. With that size constraint on the conditioning sets, evaluating the approximated log-likelihood can be done in linear complexity (see Stein, Chi, and Welty 2004; Katzfuss and Guinness 2021 for more details). While typically presented in terms of writing many small likelihoods as above, Vecchia approximations can also be considered through the lens of sparse approximations to precision matrices (Sun and Stein 2016; Finley et al. 2019) or sparse symmetric factors of precision matrices (Katzfuss and Guinness 2021; Schäfer, Katzfuss, and Owhadi 2021). Good examples of methods that more directly approximate the process, as opposed to the approximating the matrix in some purely algebraic way, would be the so-called Markov random field (MRF) models (Rue and Held 2005), which employ a graphical structure to approximate the original process in a way that directly creates sparsity in the precision matrix of the proxy process, and the closely related SPDE/stochastic finite element approach (Lindgren, Rue, and Lindström 2011; Girolami et al. 2021).

The issue of the perturbing noise \( \varepsilon \) in this latter paradigm is twofold. Most substantively, sparse approximations to precision matrices—including the Vecchia’s method introduced above—crucially depend on the screening effect, which is the phenomenon by which predictions depend very little on far-away measurements when conditioned on nearby measurements (Stein 2002, 2011). Additive white noise, for example, severely reduces the degree to which this phenomenon occurs (Stein 2011; Katzfuss and Guinness 2021), and so if such approximations are applied directly to the kernel with the added diagonal perturbation, their accuracy with respect to typical assessments such as the Kullback-Liebler (KL) divergence is significantly lowered (Katzfuss and Guinness 2021), although this admittedly does not necessarily imply that the resulting estimators one obtains by maximizing the worse likelihood approximation are in any sense “worse” (see Stein, Chi, and Welty (2004) for an example of this phenomenon). In our experience, however, particularly with singleton prediction sets (see Stein, Chi, and Welty (2004) for discussion), point estimates are indeed materially worsened in the sense of being farther from the MLE and having a lower terminal log-likelihood.

The second issue is more practical: if the log-likelihood requires a log-determinant and solving a linear system, it is difficult to get around the requirement for a matrix factorization of some perturbation of \( \mathbf{\Omega} \). As will be discussed further in the next section, the typical perturbation is \( \mathbf{\Omega} + R^{-1} \), which needs to be factorized for the log-determinant at minimum. In the case of Vecchia approximations, the current state-of-the-art provides methods for directly assembling Cholesky (or, more generally, symmetric) factors of (permutations of) \( \mathbf{\Omega} \), so that \( \mathbf{\Omega} = \mathbf{L} \mathbf{L}^T \). When there is not perturbative noise, then, one can directly assemble the factorization and avoid computing it with a sparse linear algebra library. But in the case of perturbative noise, methods such as Katzfuss and Guinness (2021) and Schäfer, Katzfuss, and Owhadi (2021) both unfortunately require first the assembly of \( \mathbf{\Omega} = \mathbf{L} \mathbf{L}^T \) and then the refactorization of \( (\mathbf{L} \mathbf{L}^T) + R^{-1} \) or some similar matrix, bringing us back to the original problem. Sparse matrix factorizations have several issues: for one, it is challenging to choose permutations of rows and columns that leads to optimal sparsity of the factors (Saad 2003). While in practice the blow-up of nonzero elements is not always a concern, it can also easily happen if one is not careful. The second issue pertains primarily to optimization, and addressing this problem is another focus of this work: derivatives of symmetric factors of precision matrices with respect to kernel parameters are challenging to work with. The simplest explanation for this is that the derivative of Cholesky factor of a matrix requires several matrix-matrix operations that are not easy to avoid (Murray 2016), which even if one disregards the issue of “fill-in,” which refers to zero entries becoming nonzero as the result of an algorithm, are computationally expensive, particularly if one wishes to compute Hessian matrices of the log-likelihood.

In this work we present a new method based on the EM algorithm (Dempster, Laird, and Rubin 1977) and stochastic trace estimation for dealing with this issue that is significantly broader and can be applied with no sparse matrix factorizations inside of optimization routines, meaning their derivatives will not be required. This is not only a practical gain in the sense of avoiding expensive operations (and operations that can potentially harm complexity, as will be discussed later), but it also enables effective automatic differentiation inside optimization steps, making it possible to perform second-order optimization of log-likelihoods whose derivatives are, at the least, very difficult to program efficiently by hand. This is in contrast to the methods mentioned above, which either perform parameter estimation using derivative-free methods or, more commonly, opt to use Bayesian methods to perform parameter estimation. The tradeoff is that we exchange a single optimization problem for an iterative fixed-point problem, which has the potential to be more burdensome to solve. By doing so, however, it is possible to compartmentalize matrix operations with \( \mathbf{\Omega} \) and \( \mathbf{R} \) in such a way that one can avoid the need to factorize \( \mathbf{\Omega} + \mathbf{R}^{-1} \) inside an optimization routine, which most other methods that solve straight optimization problems (including the ones that will be detailed below) require for their objective functions. As we will demonstrate, this tradeoff can be more than worthwhile.

Before moving to a more specific comparison of our method with its most directly related alternatives, we emphasize again that, while we apply this method exclusively to Vecchia approximations in this work, none of the tools are actually specific to Vecchia approximations or to a specific structure of \( \mathbf{R} \) beyond something that admits fast solves and, ideally at least, log-determinants. They have the potential to be useful in the SPDE framework (see Lindgren, Rue, and Lindström (2011) for examples), the MRF framework (Rue and Held 2005), “nearest-neighbor” Gaussian processes (Datta et al. 2016; Finley et al. 2019), and more, where in at least some cases analytical methods to obtain symmetric factors are not available, or where
direct methods for solving linear systems with the approximated matrices are best avoided.

1.1. Comparison with Existing Methods

Since this article will largely focus on the example setting of Vecchia approximations, we first discuss the state-of-the-art for that specific approximation. While there are many tools and software options for the sparse approximation of precision matrices, this work is specifically focused on dealing with the problem of approximating only $\Sigma^{-1}$ for covariance matrices that are given as $\Sigma + R$. It is always an option to simply ignore the reduced approximation quality and incorporate measurement noise into $\Sigma$, so that one approximates $(\Sigma + R)^{-1}$ with a sparse matrix, but as discussed above this can significantly reduce the quality of the approximation. To our knowledge, there are only two likelihood-based methodologies for directly attempting to address this problem.

The most direct comparison would be to the “sparse general Vecchia” (SGV) method presented in (Katzfuss and Guinness 2021). The idea of the SGV approximation is to condition on a prudently selected combination of noise-polluted measurements ($y_j = z_j + e_j$) and unobserved noise-free measurements ($z_j$), and then to integrate out the unobserved values (note the difference in our notation vs. the notation in Katzfuss and Guinness 2021, however). The scheme that they propose for deciding when to condition on observed values or unobserved values that need to be integrated out leads to precision matrices with valuable theoretical guarantees on sparsity structure. As an approximation of the probability distribution, it also performs meaningfully better than a naïve application of Vecchia using the kernel with the perturbation included in terms of Kullback-Liebler divergence (Katzfuss and Guinness 2021).

The other recent work on this topic is Schäfer, Katzfuss, and Owhadi (2021), which provides nice theoretical guarantees on sparsity structure. As an approximation of the probability distribution, it also performs meaningfully better than a naïve application of Vecchia using the kernel with the perturbation included in terms of Kullback-Liebler divergence (Katzfuss and Guinness 2021).

2. Estimating Covariance Parameters Using the EM Algorithm

The main idea of our method is to treat the problem of the model in 1 as a missing data problem, where the “missing” data is the data without the perturbative noise $e$, and to use the EM algorithm to estimate parameters (Dempster, Laird, and Rubin 1977). The primary observation we make in this section is to provide a specific form of the “E function” that is written in terms of standard likelihoods with just $\Sigma$ or $R$ and an additional trace term that is particularly well-suited to stochastic approximation. Using the notation introduced above, consider $Z$ to be the latent process, with a known covariance function $\mathcal{K}_0(x, x')$ but unknown parameters. For generality, we will use $R$ to denote the covariance matrix of finite-dimensional samples of the noise $e$, as the methods discussed here apply in much more general settings than $R$ diagonal. We will assume in this work, however, that $R$ is full rank. For the duration of this section, we write this section completely independently of any specific method for approximating $\Sigma$. In the interest of readability, we will simply write $\Sigma^{-1}$ for the precision with the understanding that applications will substitute the scalable $\tilde{\Sigma} \approx \Sigma^{-1}$. All of the linear algebra written here is exactly true regardless of what $\Sigma$ looks like so long as the same matrices are used consistently in all places.

The crux of the EM algorithm, at least in this setting, is preparing the “E function,” which is the expectation of the joint log-likelihood of the observed and missing data (in this case $y$ and $z$) under the conditional law of $z | y$ using some fixed parameters $\theta_0$. By design, the EM algorithm guarantees that any improvement in the E function gives at least that much improvement to the marginal log-likelihood of $y$ (Dempster, Laird, and Rubin 1977), and so this naturally gives rise to an iterative process in which one prepares an E function with parameters $\theta_0$, optimizes or improves said E function over parameters $\theta$, and then repeats the process with the updated $\theta_0 \leftarrow \theta$. While computing the expected joint log-likelihood of $y$ and $z$ in the interest of scalability may seem counter-intuitive, the following proposition, whose elementary proof is provided in Appendix A, provides a particularly convenient form for the E function.

**Proposition 1.** The expected joint negative log-likelihood (“E function”) of $(y, z)$ at parameters $\theta$ under the law with $z | y$ with covariance function $\mathcal{K}_0$ is given by

$$
\ell_{\mathcal{E}}(\theta_0) = \frac{1}{2} \text{tr} \left[ (\Sigma(\theta)^{-1} + R(\theta)^{-1}) (\Sigma(\theta_0)^{-1} + R(\theta_0)^{-1})^{-1} \right] \\
+ \ell_{\mathcal{L}}(\theta_0)(\hat{z}(\theta_0)) + \ell_{\mathcal{R}}(\theta)(y - \hat{z}(\theta_0)),
$$

where $\ell_{\mathcal{L}}(u)$ is the standard Gaussian negative log-likelihood for a vector $u$ and covariance matrix $\Sigma$ and $\hat{z}(\theta_0) := (\Sigma(\theta_0)^{-1} + R(\theta_0)^{-1})^{-1} R(\theta_0)^{-1} y$.

The full process for estimating parameters $\theta$ thus reduces to solving the fixed-point problem $F(\theta) = \theta$, where $F(\theta_0) = \arg \min_\theta \ell_{\mathcal{E}}(\theta)$ (where evaluation of $F$ is sometimes called an “M step”). If one strictly maximizes the E function described here and performs direct Picard iteration to solve the fixed point problem, this is exactly the standard EM algorithm. If one simply improves the E function, perhaps by a fixed number of Newton iterations, this procedure is a case of the generalized EM algorithm (Dempster, Laird, and Rubin 1977; Neal and Hinton 1998). The original theory from the EM algorithm in Dempster, Laird, and Rubin (1977) gives that, in this setting, any local minimizer of the marginal negative log-likelihood

$$
\mathcal{L}(\theta | y) = \log |\Sigma(\theta) + R(\theta)| + y^T (\Sigma(\theta) + R(\theta))^{-1} y
$$

is a fixed point of the above function $F$. The primary point of this additional complexity is that this framework gives the practitioner the option to approximate only $\Sigma(\theta)^{-1}$, as opposed
to \((\Sigma(\theta) + R(\theta))^{-1}\), and in doing so hopefully use much more accurate approximations and get better estimates as a result. We provide the following further guarantee that the estimates one obtains from this EM iteration corresponds to solving a set of unbiased estimating equations if the approximation to \(\Sigma(\theta)^{-1}\) gives unbiased estimating equations in the noiseless case, which is trivially satisfied by Vecchia approximations.

**Theorem 1.** Let \(\hat{\Omega}(\theta) \approx \Sigma(\theta)^{-1}\) be an approximated precision matrix with full rank that is differentiable with respect to \(\theta \in \Theta\) and such that

\[
E_\theta \nabla_\theta \left\{ -\log \left( \hat{\Omega}(\theta)^{-1} + R(\theta)^{-1} \right) + y^T (\hat{\Omega}(\theta)^{-1} + R(\theta)^{-1})^{-1} y \right\} = 0,
\]

(3)

where \(Z\) is a Gaussian process whose finite-dimensional covariance matrices are \(\Sigma(\theta)\). Then if \(Y = Z + \epsilon\) as in (1) and \(R(\theta)\) is indexed by nonoverlapping parameters (so that either \(\partial_\theta \hat{\Omega}(\theta) = 0\) or \(\partial_\theta R(\theta) = 0\) for all \(\theta\) and \(\Theta\) is open, the following two facts hold:

1. Any local minimizer of the approximated negative log-likelihood

\[
\log \left( \hat{\Omega}(\theta)^{-1} + R(\theta)^{-1} \right) + y^T (\hat{\Omega}(\theta)^{-1} + R(\theta)^{-1})^{-1} y
\]

(4)

is a fixed point of the EM iteration that also uses \(\hat{\Omega}(\theta)\) in place of \(\Sigma(\theta)^{-1}\).

2. The gradient of (4) provides unbiased estimating equations for \(\theta\).

The proof of this theorem is provided via a sequence of lemmas in Appendix A. If one were to know a priori that their approximated log-likelihood were truly unimodal as well as some technical conditions on the curvature of the log-likelihood surface detailed in Theorem 4 of Dempster, Laird, and Rubin (1977) in its subsequent discussion, conclusion 1 of the theorem could likely be strengthened to convergence to a global minimizer. Unfortunately, even for simple models (like the Matérn) and exact linear algebra unimodality is not provable. We have seen no evidence of multiple minimizers for such models, however, and so there is good reason to expect that the EM iterations will converge to the true minimizer of the approximated log-likelihood.

The value of this theorem is that it justifies a term-wise approximation of covariance components that is beneficial in settings like this one. If one is able to decompose a covariance matrix into multiple terms that individually satisfy the condition of yielding unbiased estimating equations, this theorem shows that their combination itself will yield unbiased estimating equations. In the specific case of Vecchia approximations, one may in the past have been tempted to simply include the measurement error in the covariance model and use a standard Vecchia formulation to keep the nice theoretical properties that estimator has. But this theorem demonstrates that this is not necessary, and one can work with a better approximation to the log-likelihood without giving up the guarantees of solving unbiased estimating equations. As we will show next, a stochastic approximation for the associated E function can be computed in almost exactly the same way as a standard Vecchia approximation, providing a way to work exclusively with these component approximations inside of M steps.

### 2.1. Symmetrized Hutchinson-Type Trace Estimation

While it may seem problematic, the matrix that appears in the trace term of the E function, given by

\[
(\Sigma(\theta)^{-1} + R(\theta)^{-1})(\Sigma(\theta_0)^{-1} + R(\theta_0)^{-1})^{-1},
\]

(5)

has several favorable properties that make stochastic trace estimation an appealing option. Hutchinson-type stochastic trace estimation (Hutchinson 1990) is based on the fact that \(E v^T A v = \text{tr}(A)\) for any random vector \(v\) such that \(E v = 0\) and \(E v v^T = I\). If one can draw samples from the law of \(v\), then, the empirical estimator \(\hat{\text{tr}} = \frac{1}{n} \sum_{j=1}^n v_j^T A v_j\), often also called the “sample average approximation” (SAA) in the stochastic trace estimation literature, will be an unbiased estimator for \(\text{tr}(A)\) (Hutchinson 1990; Avron and Toledo 2011; Stein, Chen, and Anitescu 2013).

The main challenge of these estimators is that the variance of the estimator may depend on matrix properties like (but not limited to) the condition number \(\kappa(A)\) or its diagonal concentration, so for a large and poorly conditioned matrix (like a large covariance matrix) the performance can be unsuitable for something as precise as maximum likelihood estimation. A significant gain can be made within the framework of Hutchinson-type estimators, however, by “symmetrizing”; a common fact about traces is that \(\text{tr}(AB) = \text{tr}(BA)\), and so by extension if \(A = LL^T\) then \(\text{tr}(AB) = \text{tr}(L^T BL) = \text{tr}(LBL^T)\). Hutchinson estimators for the Gaussian score equations that use this “symmetrized” form for the trace, for example, have variances that are at worst equal to their un-symmetrized counterparts (Stein, Chen, and Anitescu 2013), but have been demonstrated to provide a significant gain in practice (Geoga, Anitescu, and Stein 2020, 2021; Beckman et al. 2022). Further, in the notation of the inline example, if \(B\) is positive definite, then \(LBL^T\) is also positive definite, and in that case the error bounds of Hutchinson-type estimators are improved further still (Roosta-Khorasani and Ascher 2015).

The primary exploitable property of the matrix (5) is that there is good reason to believe that it is well-concentrated on the diagonal. There are multiple popular options for the distribution of \(v\), with Rademacher\((\frac{1}{2})\) (“random signs”) and unit Gaussian entries being two examples. While the variance of estimators is always smaller with Rademacher\((\frac{1}{2})\) entries, Gaussian-based estimators can have better concentration properties (Avron and Toledo 2011), and the best choice of sampling vectors is matrix- or purpose-dependent. A particularly good use case for Rademacher entries, however, is when the matrix whose trace is being computed is concentrated on the diagonal. For a general matrix \(A\), the variance of \(v^T A v\) when \(v\) has entries that are random signs is given by

\[
\text{var} \left( v^T A v \right) = 2 \sum_{j \neq k} M^2_{jk} = 2(\|A\|^2 - \sum_{j=1}^n A^2_{j,j}),
\]

compared with \(2\|A\|^2\) when \(v\) has Gaussian entries (Hutchinson 1990; Avron and Toledo 2011). While in general both of these variances can be very large, it is clear that when \(A\) is nearly diagonal the variance reduction can be substantial. And so for a matrix that looks like \(\Phi(\theta)\Phi(\theta)^{-1}\) that is smooth with respect to \(\theta\) and with \(\|\theta - \theta_0\|\) small, it is reasonable to expect the variance of this estimator, even before symmetrization, to be...
small. For this reason, Rademacher(\(\frac{1}{2}\)) entries perform very well in the setting of this particular problem.

This particular setting is also one where pre-computation can be used to great effect. Specifically, by observing that \(\theta_0\) doesn’t change for an entire M step, it is possible and clearly beneficial to pre-compute \(\{\Sigma(\theta_0)^{-1} + R(\theta_0)^{-1}\}v_j\) and reuse those pre-applied SAA vectors for the duration of the M step. This computation can also be done in an inversion-free setting by using iterative methods, further cutting down the necessary matrix operations to perform estimation. Even better, if one is willing and able to compute a symmetric factorization \(\Sigma(\theta_0)^{-1} + R(\theta_0)^{-1} = W(\theta_0)W(\theta_0)^T\), where \(W(\theta_0)\) is a symmetric factor that admits a fast solve, then one can pre-compute the vectors \(\hat{v}_j = W(\theta_0)^{-1}v_j\) to use symmetrized trace estimators that come with potentially enhanced accuracy. Unless an iterative method has been used to compute \(\hat{z}(\theta_0)\), that factorization likely has already been computed. The generic stochastic E function in the symmetrized form can thus be computed as

\[
\ell_{\theta_0}^v(\theta) \approx \frac{1}{2S} \sum_{j=1}^{S} \left[ \hat{v}_j^T (\Sigma(\theta)^{-1} + R(\theta)^{-1}) \hat{v}_j \right] + \hat{\ell}_\Sigma(\hat{z}(\theta_0)) + \ell(\theta_0)(y - \hat{z}(\theta_0)),
\]

and the un-symmetric form can be computed as

\[
\ell_{\theta_0}^u(\theta) \approx \frac{1}{2S} \sum_{j=1}^{S} \left[ v_j^T (\Sigma(\theta)^{-1} + R(\theta)^{-1}) v_j \right] + \hat{\ell}_\Sigma(\hat{z}(\theta_0)) + \ell(\theta_0)(y - \hat{z}(\theta_0)),
\]

where \(\hat{v}_j = (\Sigma(\theta_0)^{-1} + R(\theta_0)^{-1})^{-1}v_j\). In this form, one can substitute a suitable approximation for \(\Sigma(\theta)^{-1}\) and obtain a stochastic E function that can be evaluated in the same complexity as the relevant operations with the approximation to \(\Sigma(\theta)^{-1}\) (assuming that the relevant operations with \(R\) are not a computational bottleneck). Crucially, evaluating this expression requires no new matrix operations or factorizations when compared to a standard log-likelihood approximation using a sparse approximation \(\tilde{\Theta} \approx \Sigma^{-1}\).

For the specific setting of Vecchia approximations, one can do even better and evaluate these quadratic forms without ever assembling a \(O(n)\)-sized matrix to approximate \(\Sigma(\theta)^{-1}\), instead using the standard evaluation using sums of small conditional likelihoods that is trivial to parallelize and automatically differentiate. Specifically, let the standard Vecchia approximation to \(\ell_\Sigma(\theta)\) be given by \(2\ell(\theta) = \ell_{\theta_0}^v(\theta) + \text{det}(\theta)\), where the two terms correspond to the sum of small determinants and quadratic forms, respectively. Then simply by moving back and forth between the standard form and the precision matrix form of the Vecchia approximation (see Katzfuss and Guinness (2021) for full definitions), (6) can be further improved to give

\[
\ell_{\theta_0}^{\text{Vec}}(\theta) = \ell_{\theta_0}^v(\theta) + \text{det}(\theta) + \sum_{j=1}^{S} \ell(\theta_0)(S^{-1/2}\hat{v}_j) + \ell(\theta_0)(y - \hat{z}(\theta_0)),
\]

This specialized form of (6) gives a fully symmetrized and unbiased estimator for \(\ell_{\theta_0}^v(\theta)\) that can be computed in a single pass over each conditioning set and is matrix-free with respect to \(\Sigma\).

In summary, the cost of evaluating the stochastic estimator for the E function is effectively equal to the cost of evaluating a standard Vecchia likelihood for however many iid samples as one has SAA vectors. An important point to consider when analyzing the additional computational burden that this trace estimation poses is that the actual matrix operations of factorizing and computing the quadratic forms is not necessarily the most costly part of evaluating Vecchia-approximated likelihoods. Particularly for more complex covariance functions, the kernel evaluations required to assemble each conditional covariance are likely to dominate, and so in that sense the burden of extra quadratic forms is unlikely to be problematic. As a basic example, the difference in runtime cost between using \(S = 5\) and \(S = 150\) sampling vectors in the problem setting of the next section (a standard four-parameter Matérn model) is a factor of 1.4, not a factor of 30.

An important topic that requires discussion is the problem of picking the number \(M\) of sampling vectors \(\{v_j\}_{j=1}^{M}\). Answering this question precisely and in full generality is difficult if not impossible since it will depend on specific aspects of the problem like the covariance function, dimension, and sampling scheme. But with that said, there are always several easy empirical diagnostics to assess whether or not the variability of the stochastic trace is affecting estimates in a problematic way. One option would be to simply add more sampling vectors to your existing collection and perform an additional M step with the same \(\theta_0\) and compare the results. In the supplemental material we offer a demonstration of this diagnostic for the problem setting of the next section, with the results strongly indicating that even a very small number of sampling vectors gives very stable estimates. As a rule of thumb, we would ultimately suggest starting with approximately 70, if for no other reason than the fact that it perhaps provides a nice balance of variance reduction and the diminishing returns of the \(S^{-1/2}\) decay rate in standard deviation.

We close this section with a pseudo-code style summary of the algorithm to estimate parameters \(\theta\) given in Algorithm 1, which for maximum clarity is written in terms of \(\tilde{\Theta} \approx \Sigma^{-1}\) directly.

3. Demonstrations

3.1. A Comparison with SGV

In this section, we compare our estimation strategy to the one provided in Katzfuss and Guinness (2021) and made available in the GPVechcia R package. The following experiment was run 50 times:

1. Simulate 15,000 points in two dimensions at random spatial locations in \([0, 1]^2\) of a Matérn process, parameterized as \(K_\theta(x, x') = \sigma^2 M_v(\rho^{-1} \|x - x'\|) + \eta^2 I_{\|x = x'\|}\), where \(M_v\) is the Matérn correlation function given by

\[
M_v(t) = \frac{2^{1-v}}{\Gamma(v)} (\sqrt{2vt})^v K_v(\sqrt{2vt}),
\]
1 Code for the EM procedure and a kernel-agnostic Vecchia likelihood, as well as all of the scripts used to generate the results in this work, are available as a Julia language package at https://github.com/cgeoga/Veccia.jl.

\[ \Gamma \text{ is the gamma function, and } K_\nu \text{ is the modified second-kind Bessel function of order } \nu. \text{ All simulations used fixed parameters } \theta_{\text{true}} = (\sigma^2, \rho, \nu, \eta^2) = (10, 0.025, 2.25, 0.25). \]

2. Estimate the parameters using the SGV approximation via the vecchia_estimate function in GPVecchia, specified with the maximin ordering (Guinness 2018) and 10 nearest-neighbor conditioning points.

3. Estimate the parameters using our EM procedure\(^1\) with the exact same Vecchia approximation specification, initializing our EM iteration at the estimator obtained by using the standard Vecchia approximation with the covariance function that includes the nugget. For the EM procedure, allow up to 30 iterations using 72 SAA vectors with full symmetrization (although all terminated in approximately 10–15 iterations). For each M step, optimize using true gradients and Hessians of \( \ell_{\theta_0}(\theta) \) obtained via automatic differentiation (Revels, Lubin, and Papamarkou 2016; Geoga et al. 2023).

All computations were run on a computer with 16 GiB of DDR4 RAM and an Intel i5-11600K processor, using six threads (the number of physical cores). For context, for data of this size and models of this parameter count, a single gradient of the exact likelihood takes approximately 2 min to compute using ForwardDiff.jl (Revels, Lubin, and Papamarkou 2016), even if one additionally exploits manual derivative rules so that all matrix operations use the heavily optimized LAPACK and OpenBLAS libraries. Even worse, evaluating the Hessian requires more than 16 GiB of RAM if all second partial derivative matrices are computed at once, and if only some were computed at a time the cost to reassemble derivative matrices as necessary would make performance even worse. While a big enough computer with enough memory could of course solve this estimation problem exactly, this setting is already well into the regime where approximate methods are useful. The results of the estimation study are summarized in Figure 1, which shows scatterplots of pairs \(( \hat{\theta}_{\text{EM}} - \theta_{\text{true}}, \hat{\theta}_{\text{SGV}} - \theta_{\text{true}})\), and several patterns immediately stand out.

The estimates for the squared scale \( \sigma^2 \) agree reasonably well, but the range parameter \( \rho \) is at least slightly positively biased by SGV, a problem that does not occur with the EM estimators. Similarly, the error variance \( \eta^2 \) seems negatively biased in the case of SGV. By far the biggest problem, however, is that SGV-based estimates very consistently underestimate the smoothness parameter \( \nu \), to the point where every single estimate is below 2.25. In contrast, the EM-based estimator, while perhaps slightly more variable, does not display any bias. While certain parameters are difficult to estimate and cannot be estimated consistently under fixed domain asymptotics (Stein 1999), the smoothness parameter \( \nu \) in general can be estimated well in this setting, and so considering how persistent this bias is, it seems reasonable to conclude that this bias is an artifact of the SGV approximation. As a final observation to this point, retrying the estimation in the SGV case even when initializing on the exact true value of \( \nu \) gives the same result, indicating that the implied likelihood surface is clearly not flat in that region.

Figure 2 shows a histogram of the difference in exact negative log-likelihoods (assembled and computed with exact covariance matrices) of the two estimates for each trial. In 49/50 trials, the difference, computed as \( \ell_{\Sigma(\hat{\theta}_{\text{EM}})+R(\hat{\theta}_{\text{EM}})}(\mathbf{y}) - \ell_{\Sigma(\hat{\theta}_{\text{SGV}})+R(\hat{\theta}_{\text{SGV}})}(\mathbf{y}) \), is negative, indicating that the final likelihood of the EM-based estimators is superior. Considering the summary shown in Figure 1, this is unsurprising, but the magnitude of these differences can be interpreted in several different ways. On the one hand, for 15,000 points, a difference of \( \approx 15 \) log-likelihood units is very small, suggesting that the two methods performed similarly. On the other hand, however, a difference of 15 units has the potential to be very meaningful, as the log-likelihood surface can be very flat in sub-regions of the domain corresponding to parameters that have similar interpolation properties but are very distant from the MLE in a pointwise sense (Stein 1999). While that does not appear to have occurred here, a difference of 15 units absolutely merits serious attention, and the consistently better likelihood values of the EM-based estimators is, in our opinion, meaningful for this reason. If nothing else, any asymptotic theory based on MLEs, regardless of whether or not it is applicable to MLEs computed in this setting of spatial dependence, would absolutely not apply to an estimator whose likelihood is sufficiently worse that it is not even approximately equal to the MLE in finite computer precision. With all of that said, however, the EM-based estimators can take as much as twice as long to compute compared to the SGV-based estimators, with runtimes ranging between 30 sec and 2 min compared to the consistent 1 min of SGV.

As a final comparison, we consider the problem of predicting the process at the center point \( x_c = (1/2, 1/2) \) using 5000

---

**Algorithm 1**: An overview of the procedure for estimating parameters \( \theta \) using the EM algorithm and an approximate precision matrix \( \hat{\Sigma}(\theta) \approx \Sigma(\theta)^{-1} \).

- **Initialize**: Draw SAA vectors \( \{v_j\}_{j=1}^M \), with \( v_{j,k} \sim \text{Rademacher}(1/2) \).
- **while** \( \left| \theta_{j+1} - \theta_j \right| > \epsilon \) **do**
  - Compute conditional expectation \( \hat{\varepsilon}(\theta_j) := (\hat{\Sigma}(\theta_j) + R(\theta_j)^{-1})^{-1} R(\theta_j)^{-1} y \).
  - **if** Symmetrizing **then**
    - factorize \( \hat{\Sigma}(\theta_j) + R(\theta_j)^{-1} = W(\theta_j)W(\theta_j)^T \).
    - Compute pre-(half)-solved SAA vectors \( \hat{v}_j = W(\theta_j)^{-1} v_j, \ j = 1, \ldots, S \).
    - Optimize or improve the symmetrized stochastic E function (6) to obtain \( \theta_{j+1} \).
  - **else**
    - Compute pre-solved SAA vectors \( \hat{v}_j = (\hat{\Sigma}(\theta_j) + R(\theta_j)^{-1})^{-1} v_j, \ j = 1, \ldots, S \).
    - Optimize or improve the non-symmetrized stochastic E function (7) to obtain \( \theta_{j+1} \).
  - **end**
- **end**
Figure 1. Summary of 50 trials of the simulation study described above, where each point on the \((x, y)\) planes shows the difference of the EM-derived and SGV-derived point estimates, respectively, from the true parameter value. Clear bias is visible in both the range \(\rho\) and smoothness \(\nu\) for the SGV estimator, and is particularly pronounced in the latter.

Figure 2. A histogram showing the difference in the exact negative log-likelihood between EM- and SGV-based parameters for the 50 trials of the simulation study, with negative values indicating a superior terminal likelihood for the EM-based estimator.

Figure 3. A comparison of the prediction performance for the value \(z_c\) located at the center point of the domain, \((1/2, 1/2)\). Each point has \((x, y)\) coordinates given by \(|\hat{z}_c(\theta_{true}) - \hat{z}_c(\hat{\theta})|\), where \(\hat{\theta}\) denotes the estimated parameters using the EM method (x coordinate) and the SGV method (y coordinate).

nearest neighbors, with the predictand denoted \(z_c\). While not obviously the best method, it is standard practice to perform prediction by estimating model parameters and then treating them as the truth for subsequent operations like predicting. Figure 3 provides a comparison of the absolute difference of the true conditional mean from the plug-in prediction using estimated parameters for each of the 50 trials performed above, demonstrating the nontrivial effect that the discrepancy in the estimated parameters can have on even the most standard prediction problems. In this comparison, the predictions made with
the EM-based estimators are almost uniformly better than the S GV-based estimates in the sense of absolute difference from the true conditional mean. And while assessing the significance of these improvements is not trivial, for a very rough sense of their size, the prediction error variance of the process at \( x_c \) under the true law is approximately 0.0423 without the nugget. The mean absolute error between the conditional means was 0.0156 for the S GV-based predictions and 0.0062 for the EM-based predictions, and so we see that their difference of 0.0095 is nearly a quarter the size of the prediction error variance, which is not trivial.

### 3.2. An Application to Doppler LIDAR Data

We now briefly show an application of this method to a real dataset using both a nonstationary covariance model for the process \( Z \) as well as for the measurement error process \( \varepsilon \), so that \( R \) is now a nonconstant diagonal matrix. The US Department of Energy’s Atmospheric Radiation Measurement (ARM) program provides high-resolution and high-frequency meteorological measurements of a wide variety of quantities and at several locations throughout the United States (Stokes and Schwartz 1994). The Southern Great Plains (SGP) observatory is the largest site, and among many other quantities it offers highly resolved vertical wind profiles using Doppler LIDAR technology (Newsom 2012; Muradyan and Coulter 2020). For a more complete introduction and an example of a statistical study of this data, see Geoga, Anitescu, and Stein (2021). Figure 4 gives a visualization of the data, highlighting in particular that the process’ scale and measurement error both clearly change with altitude.

The model we will fit here, unlike in Geoga, Anitescu, and Stein (2021), is not concerned with estimating the height of the atmospheric boundary layer (ABL). We instead use a simple model for the space-time process given by

\[
Y(t,x) = \sigma(x)Z(t,x) + \eta(x)\varepsilon(t,x),
\]

where \( Z \) is a standard Matérn model with a full geometric anisotropy parameterized in terms of its inverse Cholesky factor, so that the geometric anisotropy matrix \( \mathbf{\Gamma}^{-1} = \mathbf{WW}^T \) is represented by parameters \( W_{1,1} > 0 \), \( W_{1,2} \), and \( W_{2,2} > 0 \), and \( \varepsilon(t,x) \) is a standard normal iid noise whose standard deviation is modulated by the spatial dependence of \( \eta(x) \). In both cases, spatial dependence is modeled simply with \( \sigma(x) = \sum_{j=1}^3 w_j(x)\sigma_j \) (and similarly for \( \eta(x) \)), where \( w_j \) are normalized weights based on the distance from knot points, placed at 0.2, 0.8, and 1.2 km. In total, the model has 10 parameters.

The results of the estimation on the approximately 32,000 measurements using (i) independent blocks of size 80 (two full vertical profiles), a natural first method to try for trivial scalability and estimates from unbiased estimating equations, (ii) a naïve Vecchia approximation that simply includes the nugget, and (iii) the EM refinement of (ii), both computed with chunks of size 20 and three past chunks for conditioning, are shown in Table 1. The naïve Vecchia method can also be considered as a representative of the many Vecchia-based estimation strategies that simply include the nugget in the kernel and primarily differ in terms of the optimization strategy, for example using Fisher scoring (Guinness 2021) or BFGS (Saha and Datta 2018). Because all of the R libraries for this estimation problem that we have encountered use hard-coded covariance functions in the C++ internals for the sake of speed, fitting this exact model to this data with a preexisting library is challenging. But the estimates here were computed using true gradients and Hessians and the high-quality Ipopt optimizer (Wächter and Biegler 2006), so there is good reason to believe that these estimates are at least as good as what one would get using one of the many other libraries based on the same underlying Vecchia approximation. In all cases, the data are ordered as a vector time series broken into two pieces (approximately, measurements below and then above 0.6m), and conditioning sets for each chunk were chosen from the most recent past. As can be seen, the true likelihood of the data under this model is nontrivially improved by the EM refinement, which considering that it retains the theoretical guarantees with regard to bias thus gives an objectively better estimator. Interestingly, while the block diagonal estimator is of course faster and would still be faster even for much larger blocks, for this particular dataset estimates with larger blocks happened to have lower terminal exact likelihoods.

As a final comment on this application, we remind the reader that this covariance model, which still is in some ways inflexible, required 10 parameters. A practitioner equipped with an \( O(n) \) likelihood approximation may still struggle to fit a model like this one if they are forced to use gradient-free optimizers like Nelder-Mead, especially considering that many natural parameterizations for covariance models give individual parameters complex dependence relationships with others, making likelihood surfaces particularly challenging to work with. The method of this article is much more amenable to derivative-based optimization by virtue of providing an E function that is easily automatically differentiated, and the accompanying software is unique among its alternatives in that a practitioner can trivially provide their own covariance functions—written in the same language as the rest of the software, no less—and imme-
EM + Vecchia − Naïve Vecchia space approximation or introduces problematic bias in the estimation, severely reduces the accuracy of the precision likelihoods have been approximated in the precision space but ing parameter estimation of Gaussian process models whose We have introduced in this work a new method for perform-

4. Discussion

We have introduced in this work a new method for perform-
ing parameter estimation of Gaussian process models whose likelihoods have been approximated in the precision space but contain some kind of polluting noise which, if not handled with special attention, severely reduces the accuracy of the precision-space approximation or introduces problematic bias in the esti-
mates. This work focused specifically on Vecchia’s approxima-
tion (Vecchia 1988) and additive measurement error, but it applies with equal directness to any method that provides sparse approximations for precision matrices and any polluting noise whose covariance matrix at least admits a fast matrix-vector product. This is a broad category, including Markov random fields (Rue and Held 2005), nearest neighbor Gaussian processes (Datta et al. 2016; Finley et al. 2019), finite-element based SPDE methods (Lindgren, Rue, and Lindström 2011), and surely many others.

The method we propose here is potentially advantageous over its alternatives in several ways. For one, its estimators provably correspond to the solution of unbiased estimating equations, unlike any of its Vecchia- and likelihood-based alternative besides simply ignoring that the nugget ruins screening and using standard Vecchia approximations anyway. Second, at least in the case of Vecchia approximations, it removes all (numerical) sparse matrix factorizations from the M steps in which optimization that benefits from derivative information is performed. While not obviously impossible, a careful implementation of the gradient and Hessian of the likelihood (or expected likelihood) that is written in terms of Cholesky factors for the precision would be very difficult to put together in a way that preserves performance, parallelizability, and computational complexity. Practically speaking, the E function given here that is optimized in each M step is the first objective function in the setting of perturbed Vecchia approximations for which second derivatives are conveniently computable, either by hand or by automatic differen-
tiation. In fact, to our knowledge this is the first publication that uses true Hessian matrices of any Vecchia approximation, with the closest prior work using expected Fisher information matrices (Guinness 2021). Moreover, it makes use of the newly available automatic derivatives for $K_r$ provided in Geoga et al. (2023), which can be composed at no extra effort with the automatic derivatives of the E function itself and allow derivative based estimation of smoothness parameters. It is worth noting that Bayesian sampling-based methods sidestep the optimization problem and can also make use of approximate likelihoods (Finley et al. 2019), although with unclear impacts on the result-
ing posteriors.

There are many questions that this work does not answer. Primarily, this method can only be as good as the original approximation $\hat{\Omega}(\theta) \approx \Sigma(\theta)^{-1}$, but particularly for nonadap-
tive approximation schemes like Vecchia’s the quality of this approximation can vary. For one example, separable covariance functions that are popular in fields like computer experiments can give rise to processes that do not screen well (Stein 2011). For kernels that are smooth away from the origin, a property that is closely connected to screening, many of the algebraic approximation tools discussed in the introduction exploit the rank deficiency of off-diagonal matrix blocks. But that level of rank deficiency changes with the dimension of the process (see Ambikasaran et al. (2016) for an example), making those algorithms less performant. We expect this reduced approximation accuracy to carry over to precision-based methods like Vecchia approximations. Another natural question to ask is how one obtains standard errors for the resulting estimators. There is a straightforward relationship between the Hessian of the log-likelihood for the data and derivatives of the E and M functions (Dempster, Laird, and Rubin 1977), but considering that we are using an approximated likelihood, this Hessian is almost certainly not even asymptotically the precision of the MLE. Considering this additional complication, we leave a more careful investigation of uncertainty quantification in this setting to a future project. With regard to efficiency, in the symmetrized case using exact covariance matrices Stein, Chen, and Anitescu (2013) gives a bound on the efficiency cost of $1 + S^{-1}$, which is quite small, and the supplemental information shows similarly optimistic if not better results in the setting of this work. The efficiency cost of Vecchia approximations themselves, however, is an open area of study with only a few special cases having any theory at all, and so this work cannot make any claims about the efficiency of resulting estimators. Finally, we note that in the case of restricted maximum likelihood estimation (REML), it is possible for the supremum of the likelihood function to occur as the range tends to infinity (Stein 2022), which may pose problems for iterative methods like the EM algorithm. Considering that the log-likelihood surface for some spatial data and models can be very flat on certain portions of the domain, like when range or inverse range parameters tend to infinity or zero (Gu, Wang, and Berger 2018), sufficiently poor initialization may also be problematic.

Lastly, we observe that there are many potential improve-
ments to our approach here that are not discussed in this work. For example, many more accurate methods for trace approximation exist outside of the Hutchinson paradigm, like the peeling method (Lin, Lu, and Ying 2011), first applied to GPs in Minden et al. (2017). Adaptive Hutchinson- or Krylov-type methods also

| Model              | $\ell_{\text{exact}}(\theta)$ | $\sigma_1$ | $\sigma_2$ | $\sigma_3$ | $W_{1,1}$ | $W_{1,2}$ | $W_{2,1}$ | $W_{2,2}$ | $\nu$ | $\eta_1$ | $\eta_2$ | $\eta_3$ |
|--------------------|-------------------------------|------------|------------|------------|-----------|-----------|-----------|-----------|-------|------------|------------|------------|
| Indep. Blocks      | $-27317$                      | 0.92       | 0.36       | 0.54       | 8.30      | 0.17      | 773.36    | 0.61      | 0.001 | 0.01       | 0.10       |            |
| Naïve Vecchia      | $-31898$                      | 1.20       | 0.48       | 0.03       | 12.26     | 0.71      | 1069.83   | 0.99      | 0.001 | 0.0003     | 0.05       |            |
| EM + Vecchia       | $-31916$                      | 1.12       | 0.47       | 0.04       | 14.41     | 0.64      | 1272.12   | 1.06      | 0.006 | $2 \times 10^{-5}$ | 0.05       |            |

The lowest (and best) negative log-likelihood is emphasized in bold text.
exist (Meyer et al. 2021; Persson, Cortinovis, and Kressner 2022; Chen and Hallman 2022), which may offer improvements in some settings without requiring so much additional machinery. Moreover, many extensions, improvements, and generalizations of the EM algorithm exist (see Liu and Rubin 1994) for one of many examples), and it is conceivable that at least some of those extensions may provide significant benefit in this setting.

Supplementary Materials

The supplemental materials provide proofs of Proposition 1 and Theorem 1 and a demonstration of the effect of varying the number of sampling vectors in the stochastic trace as discussed at the end of Section 2.

Acknowledgments

The authors thank Lydia Zoells for her careful copyediting.

Funding

This material was based upon work supported by the US Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) under Contract DE-AC02-06CH11347.

ORCID

Christopher J. Geoga ORCID: https://orcid.org/0000-0002-5060-4799

References

Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D., and O’Neil, M. (2016), “Fast Direct Methods for Gaussian Processes,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 38, 252–265. [603,611]

Anitescu, M., Chen, J., and Wang, L. (2012), “A Matrix-Free Approach for Solving the Parametric Gaussian Process Maximum Likelihood Problem,” SIAM Journal on Scientific Computing, 34, A240–A262. [603]

Avron, H., and Toledo, S. (2011), “Randomized Algorithms for Estimating the Trace of an Implicit Symmetric Positive Semi-Definite Matrix,” Journal of the ACM, 58, 8:1–8:34. [606]

Beckman, P. G., Geoga, C. J., Stein, M. L., and Anitescu, M. (2022), “Scalable Computations for Nonstationary Gaussian Processes,” Technical Report arXiv:2206.05220, arXiv:2206.05220 [stat] type: article. [606]

Chen, J., and Stein, M. L. (2021), “Linear-Cost Covariance Functions for Gaussian Random Fields,” Journal of the American Statistical Association, 118, 147–164. [603]

Chen, T., and Hallman, E. (2022), “Krylov-Aware Stochastic Trace Estimation,” Technical Report arXiv:2205.01736, arXiv:2205.01736 [cs, math] type: article. [612]

Cressie, N., and Johannesson, G. (2008), “Fixed Rank Kriging for Very Large Spatial Data Sets,” Journal of the Royal Statistical Society, Series B, 70, 209–226. [603]

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016), “Hierarchical Near-Neighbor Gaussian Process Models for Large Geostatistical Datasets,” Journal of the American Statistical Association, 111, 800–812. [604,611]

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), “Maximum Likelihood from Incomplete Data via the EM Algorithm,” Journal of the Royal Statistical Society, Series B, 39, 1–22. [604,605,606,611]

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., and Banerjee, S. (2019), “Efficient Algorithms for Bayesian Near-Neighbor Gaussian Processes,” Journal of Computational and Graphical Statistics, 28, 401–414. [604,611]

Geoga, C. J., Anitescu, M., and Stein, M. L. (2020), “Scalable Gaussian Process Computations Using Hierarchical Matrices,” Journal of Computational and Graphical Statistics, 29, 227–237. [603,606]

——— (2021), “Flexible Nonstationary Spatiotemporal Modeling of High-Frequency Monitoring Data,” Environmetrics, 32, e2670. [606,610]

Geoga, C. J., Marin, O., Schanen, M., and Stein, M. L. (2023), “Fitting Matérn Smoothness Parameters Using Automatic Differentiation,” Statistics and Computing, 33, 48. [608,611]

Girolami, M., Febrianto, E., Yin, G., and Cirak, F. (2021) , “The Statistical Finite Element Method (Stafem) for Coherent Synthesis of Observation Data and Model Pedictions,” Computer Methods in Applied Mechanics and Engineering, 375, 113533. [604]

Gu, M., Wang, X., and Berger, J. O. (2018), “Robust Gaussian Stochastic Process Emulation,” The Annals of Statistics, 46, 3038–3066. [611]

Guinness, J. (2018), “Permutation and Grouping Methods for Sharpening Gaussian Process Approximations,” Technometrics, 60, 415–429. [605,606]

——— (2021), “Gaussian Process Learning via Fisher Scoring of Vecchia’s Approximation,” Statistics and Computing, 31, 1–8. [610,611]

Hutchinson, M. F. (1990), “A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines,” Communications in Statistics – Simulation and Computation, 19, 433–450. [606]

Katzfuss, M., and Cressie, N. (2012), “Bayesian Spatio-Temporal Smoothing for Very Large Datasets,” Environmetrics, 23, 94–107. [603]

Katzfuss, M., and Guinness, J. L. (2021), “A General Framework for Vecchia Approximations of Gaussian Processes,” Statistical Science, 36, 124–141. [604,605,607]

Kaufman, C. G., Schervish, M. J., and Nychka, D. W. (2008), “Covariance Tapering for Likelihood-based Estimation in Large Spatial Data Sets,” Journal of the American Statistical Association, 103, 1545–1555. [603]

Lin, L., Lu, J., and Ying, L. (2011), “Fast Construction of Hierarchical Matrix Representation from Matrix-Vector Multiplication,” Journal of Computational Physics, 230, 4071–4087. [611]

Lindgren, F., Rue, H., and Lindström, J. (2011), “An Explicit Link Between Gaussian Fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach,” Journal of the Royal Statistical Society, Series B, 73, 423–498. [604,611]

Litvinenko, A., Sun, Y., Genton, M., and Keyes, D. (2019), “Likelihood Approximation with Hierarchical Matrices for Large Spatial Datasets,” Computational Statistics & Data Analysis, 137, 115–132. [603]

Liu, C., and Rubin, D. (1994), “The ECME Algorithm: A Simple Extension of EM and ECM with Faster Monotone Convergence,” Biometrika, 81, 633–648. [612]

Meyer, R. A., Musco, C., Musco, C., and Woodruff, D. P. (2021), “Hutch++: Optimal Stochastic Trace Estimation,” in Proceedings of Symposium on Simplicity in Algorithms (SOSA), pp. 142–155, Society for Industrial and Applied Mathematics. [612]

Minden, V., Damle, A., Ho, K., and Ying, L. (2017), “Fast Spatial Gaussian Process Maximum Likelihood Estimation via Skeletonization Factorizations,” Multiscale Model & Simulation, 15, 1584–1611. [611]

Mockus, J. (2012), Bayesian Approach to Global Optimization: Theory and Applications, Dordrecht: Springer. [603]

Muradhan, P., and Coulter, R. L. (2020), Radar Wind Profiler (RWP) and Radio Acoustic Sounding System (RASS) Instrument Handbook (DOE/SC-ARM-TR-044). DOE Office of Science Atmospheric Radiation Measurement (ARM) Program, United States. [610]

Murray, I. (2016), “Differentiation of the Cholesky Decomposition,” arXiv preprint arXiv:1602.07527. [604]

Neal, R. M., and Hinton, G. E. (1998), “A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants,” in Learning in Graphical Models, pp. 355–368, Springer. [605]

Newsom, R. K. (2012), “Doppler lidar (DL) Handbook,” Technical Report DOE/SC-ARM/TR-101, DOE Office of Science Atmospheric Radiation Measurement (ARM) Program (United States). [610]

Persson, D., Cortinovis, A., and Kressner, D. (2022), “Improved Variants of the Hutch++ Algorithm for Trace Estimation,” SIAM Journal on Matrix Analysis and Applications, 43, 1162–1185. [612]

Revels, J., Lubin, M., and Papamarkou, T. (2016), “Forward-Mode Automatic Differentiation in Julia,” arXiv:1607.07892 [cs.MS]. [608]
Roosta-Khorasani, F., and Ascher, U. (2015), “Improved Bounds on Sample Size for Implicit Matrix Trace Estimators,” *Foundations of Computational Mathematics*, 15, 1187–1212. [606]

Rue, H., and Held, L. (2005), *Gaussian Markov Random Fields: Theory and Applications*, New York: Chapman and Hall/CRC. [604,611]

Saad, Y. (2003), *Iterative Methods for Sparse Linear Systems*, Philadelphia, PA: SIAM. [604]

Saha, A., and Datta, A. (2018), “Brisc: Bootstrap for Rapid Inference on Spatial Covariances,” *Stat*, 7, e184. [610]

Schafer, F., Katzfuss, M., and Owhadi, H. (2021), “Sparse Cholesky Factorization by Kullback–Leibler Minimization,” *SIAM Journal on Scientific Computing*, 43, A2019–A2046. [604,605]

Stein, M. L. (1999), *Interpolation of Spatial Data: Some Theory for Kriging*, New York: Springer. [608]

——— (2002), “The Screening Effect in Kriging,” *The Annals of Statistics*, 30, 298–323. [604]

——— (2011), “2010 Rietz Lecture: When Does the Screening Effect Hold?” *The Annals of Statistics*, 39, 2795–2819. [604,611]

——— (2022), “Local Variogram Models with Negative Inverse “Range” Parameters,” *Spatial Statistics*, 48, 100633. [611]

Stein, M. L., Chen, J., and Anitescu, M. (2013), “Stochastic Approximation of Score Functions for Gaussian Processes,” *The Annals of Applied Statistics*, 7, 1162–1191. [603,606,611]

Stein, M. L., Chi, Z., and Welty, L. J. (2004), “Approximating Likelihoods for Large Spatial Data Sets,” *Journal of the Royal Statistical Society*, Series B, 66, 275–296. [604]

Stokes, G. M., and Schwartz, S. E. (1994), “The Atmospheric Radiation Measurement (ARM) Program: Programmatic Background and Design of the Cloud and Radiation Test Bed,” *Bulletin of the American Meteorological Society*, 75, 1201–1222. [610]

Stroud, J. R., Müller, P., and Sansó, B. (2001), “Dynamic Models for Spatiotemporal Data,” *Journal of the Royal Statistical Society*, Series B, 63, 673–689. [603]

Sun, Y., and Stein, M. L. (2016), “Statistically and Computationally Efficient Estimating Equations for Large Spatial Datasets,” *Journal of Computational and Graphical Statistics*, 25, 187–208. [604]

Ubaru, S., Chen, J., and Saad, Y. (2017), “Fast Estimation of $\text{tr}(f(A))$ via Stochastic Lanczos Quadrature,” *SIAM Journal on Matrix Analysis and Applications*, 38, 1075–1099. [603]

Vecchia, A. V. (1988), “Estimation and Model Identification for Continuous Spatial Processes,” *Journal of the Royal Statistical Society*, Series B, 50, 297–312. [603,611]

Wikel, C. K., and Cressie, N. (1999), “A Dimension-Reduced Approach to Space-Time Kalman Filtering,” *Biometrika*, 86, 815–829. [603]

Wächter, A., and Biegler, L. T. (2006), “On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming,” *Mathematical Programming*, 106, 25–57. [610]