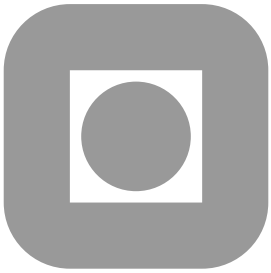
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Parameter Estimation in High Dimensional Gaussian Distributions

by

Erlend Aune, Daniel Simpson and Jo Eidsvik

PREPRINT
STATISTICS NO. 5/2012



NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL http://www.math.ntnu.no/preprint/statistics/2012/S5-2012.pdf
Erlend Aune has homepage: http://www.math.ntnu.no/∼erlenda
E-mail: erlenda@math.ntnu.no
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Abstract

In order to compute the log-likelihood for high dimensional Gaussian models, it is necessary to compute the determinant of the large, sparse, symmetric positive definite precision matrix. Traditional methods for evaluating the log-likelihood, which are typically based on Choleksy factorisations, are not feasible for very large models due to the massive memory requirements. We present a novel approach for evaluating such likelihoods that only requires the computation of matrix-vector products. In this approach we utilise matrix functions, Krylov subspaces, and probing vectors to construct an iterative numerical method for computing the log-likelihood.

1 Introduction

In computational and, in particular, spatial statistics, increasing possibilities for observing large amounts of data leaves the statistician in want of computational techniques capable of extracting useful information from such data. Large datasets arise in many applications, such as modelling seismic data acquisition (Buland and Omre, 2003); analysing satellite data for ozone intensity, temperature and cloud formations (McPeters et al., 1996); or constructing global climate models (Lindgren et al., 2011). Most models in spatial statistics are based around multivariate Gaussian distributions, which means that random vector $\mathbf{x} = (x_1, \ldots, x_n)^T$ has probability density function

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\eta}) = (2\pi)^{-n/2} \det(\mathbf{Q}_{\boldsymbol{\eta}})^{1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{Q}_{\boldsymbol{\eta}}(\mathbf{x}-\boldsymbol{\mu})\right),$$

where the mean vector is $\boldsymbol{\mu}$, and the precision matrix $\mathbf{Q}_{\boldsymbol{\eta}}$ is the inverse of the covariance matrix, which depends on the parameters $\boldsymbol{\eta}$. For short, we write $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}_{\boldsymbol{\eta}}^{-1})$. In this paper, we assume that the precision matrix is sparse, that is, most of its entries are zero. For our purposes this sparseness arises from a Markov property on the random vector $\mathbf{x}$, which gives computational advantages (Rue and Held, 2005). Moreover, the sparse structure also has strong physical and statistical motivations (Lindgren et al., 2011). We note that Rue and Tjelmeland (2002) showed that it is possible to approximate general Gaussian random fields on a lattice by multivariate Gaussians with sparse precision matrices.

Throughout this paper, we will consider the common Gauss-linear model, in which our data $\mathbf{y} = (y_1, \ldots, y_{n_y})^T$ is a noisy observation of a linear transformation of a true random field, that is

$$\mathbf{y} = \mathbf{A}_{\boldsymbol{\theta}}\mathbf{x} + \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}}^{-1}), \tag{1}$$

where the matrix $\mathbf{A}_{\boldsymbol{\theta}}$ connects the true underlying field $\mathbf{x}$ to observations. We assume that $\mathbf{A}_{\boldsymbol{\theta}}$ and $\mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}}$ are sparse matrices. In the simplest case they are diagonal, or block diagonal. Under this Gauss-linear model assumptions the conditional distribution of $\mathbf{x}$, given $\mathbf{y}$, is Gaussian with $\mathbf{x}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}, \mathbf{Q}_{\mathbf{x}|\mathbf{y}}^{-1})$, where $\mathbf{Q}_{\mathbf{x}|\mathbf{y}} = \mathbf{Q}_{\boldsymbol{\eta}} + \mathbf{A}_{\boldsymbol{\theta}}^T\mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}}\mathbf{A}_{\boldsymbol{\theta}}$ and $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}} = \mathbf{Q}_{\mathbf{x}|\mathbf{y}}^{-1}(\mathbf{Q}_{\boldsymbol{\eta}}\boldsymbol{\mu} + \mathbf{A}_{\boldsymbol{\theta}}^T\mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}}\mathbf{y})$. Estimating the parameters, $\boldsymbol{\eta}, \boldsymbol{\theta}$, in the frequentist way amounts to maximising the following likelihood

$$p(\mathbf{y}|\boldsymbol{\eta}, \boldsymbol{\theta}) \propto \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\eta})}{p(\mathbf{x}|\mathbf{y}, \boldsymbol{\eta}, \boldsymbol{\theta})}. \tag{2}$$

In the Bayesian setting, we look at the posterior distribution of model parameters, $p(\boldsymbol{\eta}, \boldsymbol{\theta}|y)$, which decomposes similarly, and we often need to compute the mode of this distribution. In both cases, we minimise the function $\Phi(\boldsymbol{\eta}, \boldsymbol{\theta}) = -2\log(f(\boldsymbol{\eta}, \boldsymbol{\theta}))$ for $f = p(\mathbf{y}|\boldsymbol{\eta}, \boldsymbol{\theta})$ or $f = p(\boldsymbol{\eta}, \boldsymbol{\theta}|y)$. These

expressions involve the log-determinant of matrices. When we evaluate (2) at the conditional mean $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}} = \boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{\eta}, \boldsymbol{\theta})$, the likelihood is available as

$$\begin{aligned} 2\log p(\mathbf{y}|\boldsymbol{\eta}, \boldsymbol{\theta}, \sigma^2, \lambda^2) =& n_y \log(2\pi) + \log\det \mathbf{Q}_{\boldsymbol{\eta}} + \log\det \mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}} - \log\det(\mathbf{Q}_{\boldsymbol{\eta}} + \mathbf{A}_{\boldsymbol{\theta}}^T \mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}} \mathbf{A}_{\boldsymbol{\theta}}) \\ & - (\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}} - \boldsymbol{\mu})^T \mathbf{Q}_{\boldsymbol{\eta}} (\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}} - \boldsymbol{\mu}) - (\mathbf{y} - \mathbf{A}\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}})^T \mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}} (\mathbf{y} - \mathbf{A}\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}). \end{aligned} \tag{3}$$

Thus, the main computational requirement is the evaluation of three log-determinants, where $\log\det \mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}}$ usually is trivial to compute.

We consider the situations when n and n_y are very large, say 10^6. In such high dimensions the direct determinant evaluations of the terms in (3) often become infeasible due to computational costs and storage limitations. For instance, the standard method of computing the determinant through the Cholesky factor is in most situations impossible due to enormous storage requirements. We suggest to use ideas from numerical linear algebra to overcome this problem, and present methods for likelihood evaluation or Bayesian computations that are useful for massive datasets. Our approach relies on fast evaluation of sparse matrix-vector products.

Previous approaches have tried to circumvent the determinant evaluation by constructing approximate likelihood models. A determinant-free approach is investigated in Fuentes (2007), based on estimated spectral densities. Pseudo-likelihood methods (Besag, 1974), composite likelihood and block composite likelihood (Eidsvik et al., 2011) combine subsets of the data to build an approximate likelihood expression. What these methods generally have in common is that they change the statistical model; i.e. they make simplifying assumptions about the model to reduce the computing dimensions. Our approach differs from these in that we do not approximate the likelihood model, but rather approximate the log-determinants expressions directly.

In Section 2 we outline the main concepts behind our log-determinant evaluation and the different challenges involved. Section 3 presents solutions to these different challenges, using a number of results from numerical linear algebra, complex analysis and graph theory. Results are shown for real and synthetic datasets in Section 4.

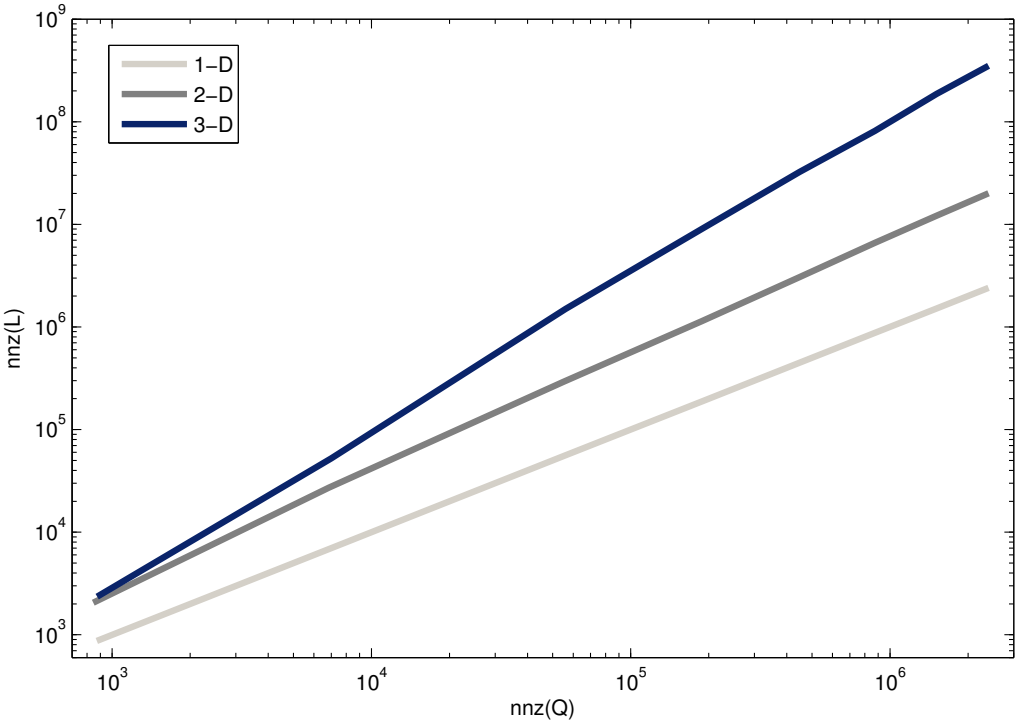
2 Log-determinant evaluations

Precision and covariance matrices are characterised by being symmetric, positive definite; that is $\mathbf{Q} = \mathbf{Q}^T$ and for all $\mathbf{z} \in \mathbb{R}^n$, $\mathbf{z}^T\mathbf{Q}\mathbf{z} > 0$. For this class of matrices, the log-determinant can be found through the Cholesky factor of $\mathbf{Q}$ in the following manner: Let $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is lower triangular. Then $\log\det \mathbf{Q} = 2\sum_i \log L_{ii}$. This is the most common way to compute the log-determinant. It takes only a few lines of code using a library for computing the Cholesky factor, such as CHOLMOD (Davis and Hager, 1999; Chen et al., 2008).

If $\mathbf{Q}$ is dense, computing $\mathbf{L}$ is an $\mathcal{O}(n^3)$ operation, and this quickly becomes infeasible for large n. If $\mathbf{Q}$ is sparse, much lower computational complexities may be obtained. In particular, if $\mathbf{x}$ is a one dimensional random field, such as an random walk or characterised through some stochastic differential equation, the computational complexity for computing $\mathbf{L}$ is $\mathcal{O}(n)$. Similarly, for a 2-D Markovian field, the complextiy is $\mathcal{O}(n^{3/2})$ and for a 3-D Markovian field $\mathcal{O}(n^2)$ (Rue and Held, 2005). These order terms are obtained after reordering the elements in the precision matrix. The fill-in is defined by the number of extra non-zero terms in $\mathbf{L}$, compared with $\mathbf{Q}$. This fill-in becomes large for higher dimensional processes. In Figure 1 we plot the number of non-zero entries of $\mathbf{L}$ versus that of $\mathbf{Q}$ on a log-scale. The Cholesky factor (second axis) grows quickly in 3-D, causing

memory requirements to explode. The precision matrices used to display this figure come from discretization of the Laplacian in 1-D, 2-D and 3-D.

Figure 1: Loglog-plot of nonzero elements in the precision matrix $\mathbf{Q}$ (first axis) versus nonzero elements in the Cholesky factor $\mathbf{L}$ (second axis). The precision matrices are constructed from a discretized Laplacian in 1-D, 2-D and 3-D.

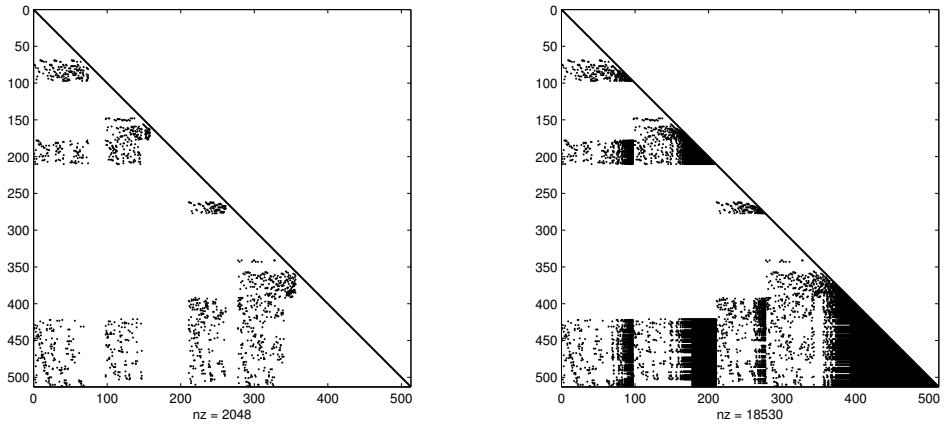


In Figure 2 an illustration of fill-in in the Cholesky factor is depicted. Here, the precision matrix $\mathbf{Q}$ of the 3-D Laplacian is used (lower triangular part of $\mathbf{Q}$ shown in left display). The lower triangular Cholesky factor $\mathbf{L}$ (right display) is obtained using METIS reordering (Karypis and Kumar, 1999).

In this paper we suggest methods to overcome the prohibitive storage requirements of the Cholesky approach by using ideas from different areas of numerical mathematics, namely

- a matrix identity stating that log-determinants are equal to the tr $\log \mathbf{Q}$, where $\log \mathbf{Q}$ is the matrix logarithm,
- Cauchy's integral formula along with rational approximations for computing the log matrix times a vector (Higham, 2008),
- Krylov subspace methods for solving linear systems (Saad, 2003),
- stochastic probing vectors (Hutchinson, 1989; Bekas et al., 2007; Tang and Saad, 2010).

Figure 2: Illustration of fill-in for a 3-D Laplacian. The black dots indicate the non-zero structure of matrices. The lower triangular part of the precision matrix (left) is very sparse, with 2048 non-zero elements. In contrast, the Cholesky factor (right) contains 18530 non-zero elements.



We next outline these main concepts for evaluating log-determinants. Section 3 presents several useful extensions for practical use.

It appears that approximating the determinant of a large sparse matrix to sufficient accuracy is a hard problem. Nevertheless, several approximating techniques exist in the literature, the most useful of which is the approximation developed in Hutchinson (1989). This is the method extended in this paper.

The Hutchinson estimator was originally developed for calculating the trace of the inverse of a matrix, but the method easily generalises to our situation by the following observation (Bai and Golub, 1997):

$$\log\det\mathbf{Q} = \mathrm{tr}\,\log\mathbf{Q}. \tag{4}$$

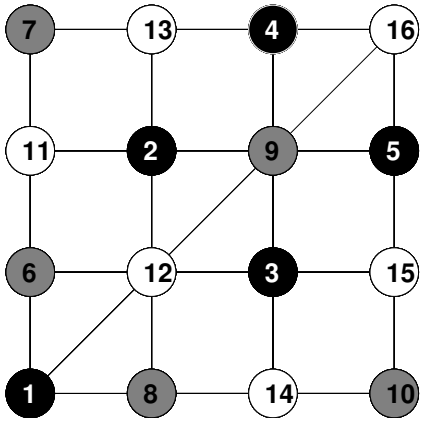
In the symmetric positive definite case, this identity is proved noting that $\det\mathbf{Q} = \prod_{i=1}^{n}\lambda_i$ where $\{\lambda_i\}$ are the eigenvalues of $\mathbf{Q}$ and that $\log\mathbf{Q} = \boldsymbol{U}\log(\boldsymbol{\Lambda})\boldsymbol{U}^T$ with $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$ and $\boldsymbol{U}$ contains the eigenvectors of $\mathbf{Q}$. Furthermore, $\mathrm{tr}\,(\boldsymbol{U}\log(\boldsymbol{\Lambda})\boldsymbol{U}^T) = \mathrm{tr}\,(\boldsymbol{U}\boldsymbol{U}^T\log\boldsymbol{\Lambda}) = \mathrm{tr}\,\log\boldsymbol{\Lambda}$, which gives the identity.

For practical implementation of this result we note the following;

$$\mathrm{tr}\,\log\mathbf{Q} = \sum_{j=1}^{n}\mathbf{e}_j^T\log(\mathbf{Q})\mathbf{e}_j, \tag{5}$$

where $\mathbf{e}_j = (0,\ldots,1,\ldots,0)^T$ and the 1 entry is in position j. The unit vectors extract the diagonal of $\log\mathbf{Q}$ in (5). From this we can recover the Hutchinson estimator by introducing probing vectors $\mathbf{v}_j$ as follows: Let $\mathbf{v}_j$, $j = 1,\ldots,s$ be vectors with random entries at element k defined by $P(v_j^k = 1) = 1/2$,

Figure 3: Illustration of 1-distance colouring. Nodes sharing an edge cannot have the same colour.



$P(v_j^k = -1) = 1/2$, independently for all $k = 1, \ldots, n$. Next, let

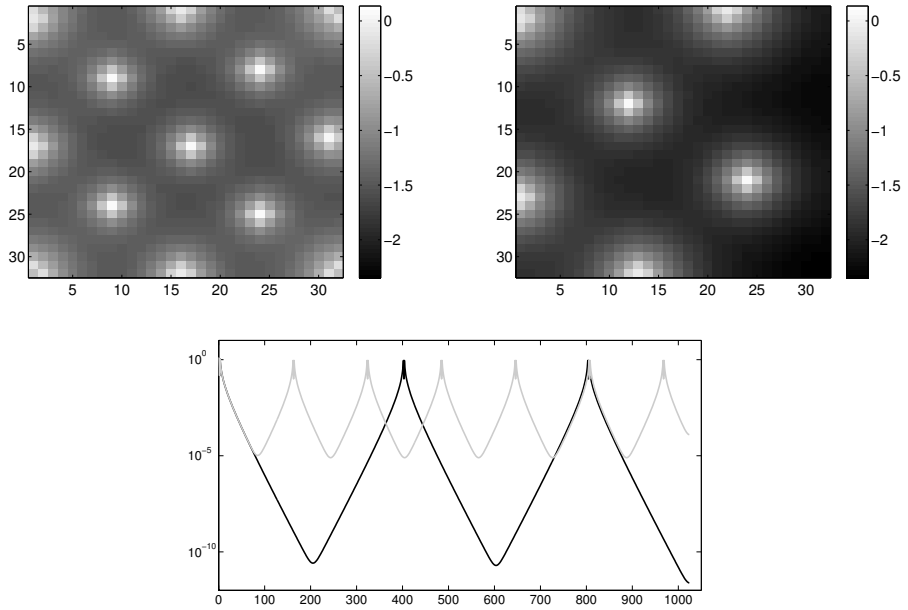
$$\text{tr} \log \mathbf{Q} \approx \frac{1}{s} \sum_{j=1}^{s} \mathbf{v}_j^T \log(\mathbf{Q}) \mathbf{v}_j. \tag{6}$$

It is possible to compute confidence regions for the estimator in (6), using either Monte Carlo techniques or the Hoeffding inequality (Bai and Golub (1997)). This can give guidelines for choosing $s < n$. The memory requirements are low, but since this is a Monte Carlo method, the Hutchinson estimator requires a large s to be sufficiently accurate.

One method for keeping the number of probing vectors to a reasonable number is to choose the $\mathbf{v}_j$s in a clever way, so that we require far fewer vectors than a Monte Carlo method. In recent publications, Bekas et al. (2007) and Tang and Saad (2010) explored the use of probing vectors for extracting the diagonal of a matrix or its inverse. Bekas et al. (2007) extract the diagonal of a sparse matrix under mild conditions. Tang and Saad (2010) relies on an approximate sparsity pattern of $\mathbf{Q}^{-1}$, determined by a power of the original matrix, i.e. $\mathbf{Q}^p$, $p = 2, 3, \ldots,$. That it is always true for large enough p can be seen using polynomial approximation. It turns out that if the sparsity structure of $\mathbf{Q}^{-1}$ can be approximated by that of $\mathbf{Q}^p$, then a set of probing vectors can be computed that takes this into account by using a colouring of the adjacency graph of $\mathbf{Q}^p$.

In this paper, we are considering Gaussian random vectors that have an approximate *Markov property*, which, equivalently, means that their precision matrices are approximately sparse. We can, therefore, associate with each precision matrix a graph, such as the one shown in Figure 3. We can use this graph structure, and the idea that $\mathbf{Q}^{-1}$ or $\log(\mathbf{Q})$ can be well approximated by a matrix with the same sparsity structure as $\mathbf{Q}^p$ to design a good set of probing vectors. Ideally, we would chose $\mathbf{v}_j \equiv \mathbf{e}_j$, however, as that is not possible, we will relax our requirements and chose a set of probing vectors that are sums of $\mathbf{e}_j$s. In order to not lose too much accuracy with this approximation, we need to make sure that the non-zero elements of $\mathbf{v}_j$ are sufficiently separated in some appropriate sense. Using the fact that our desired matrix function is well approximated by

Figure 4: Illustration of $\log(\mathbf{Q})\mathbf{v}_i$ for different probing vectors. Right: $\log(\mathbf{Q})\mathbf{v}_i$ in the situation with few probing vectors. Left: Situation with more probing vectors. Bottom: The same computation for the 1-D problem.



$\mathbf{Q}^p$, Tang and Saad (2010) suggested that a good choice of probing vectors would have the property that if both the kth and ℓth element of $\mathbf{v}_j$ were non-zero, then the (k,ℓ)-entry of $\mathbf{Q}^p$ is zero. A set of probing vectors with this property can be constructed using a graph colouring of $\mathbf{Q}^p$.

A neighbourhood colouring of the graph induced by $\mathbf{Q}^p$ associates with each node a colour, c, such that no adjacent nodes have the same colour. While constructing good graph colourings is generally a difficult problem, sufficiently good colourings can be generated easily using greedy algorithms (Culberson, 1992). Figure 3 illustrates the concept with three colours inducing three probing vectors. Here, the probing vectors are defined by $v^1_{1,2,3,4,5} = 1, v^2_{6,7,8,9,10} = 1, v^3_{11,12,13,14,16} = 1$, with the remaining entries equal to zero.

A heuristic method suggested in Tang and Saad (2010) is to find the power, p in $\mathbf{Q}^p$ by solving $\mathbf{Q}\mathbf{x} = \mathbf{e}_j$ and setting $p = \min\{d(l,j)||x_l| < \epsilon\}$ where $d(\cdot,\cdot)$ defines the graph distance. In our case, we may compute $\log(\mathbf{Q})\mathbf{e}_j$ and apply the same heuristic. Figure 4 illustrates how ones in a probing vector influence neighbors. This is illustrated on a grid, where the size is 32×32, i.e. $n = 1024$. We discuss some issues with using this kind of probing vectors in Section 3.4, and propose a potential remedy. Note that the probing vectors need not be stored, but may be computed cheaply on the fly. If we pre-compute them, they are sparse, and do not need much storage. Since what we need for each probing vector is $\mathbf{v}_j^T \log(\mathbf{Q})\mathbf{v}_j$, we observe that the computation is highly parallel with low communication costs. Each node gets one probing vector, and computes $\mathbf{v}_j^T \log(\mathbf{Q})\mathbf{v}_j$ and sends back the result. In essence, this leads to linear speedup in the amount of processors available with proportionality close to unity.

The procedure described above requires the evaluation of $\log(\mathbf{Q})\mathbf{v}_j$. The matrices we consider

have real positive spectrum, and it is possible to evaluate $\log(\mathbf{Q})\mathbf{v}_j$ through Cauchy's integral formula,

$$\log(\mathbf{Q})\mathbf{v}_j = \oint_\Gamma \log(z)(z\mathbf{I} - \mathbf{Q})^{-1}\mathbf{v}_j dz, \tag{7}$$

where Γ is a suitable curve enclosing the spectrum of $\mathbf{Q}$ and avoiding branch cuts of the logarithm. Discretizing this integral leads to a rational approximation of $\log(\mathbf{Q})\mathbf{v}$ of the following form

$$\log(\mathbf{Q})\mathbf{v}_j \approx f_N(\mathbf{Q})\mathbf{v}_j = \sum_{l=1}^{N} \alpha_l(\mathbf{Q} - \sigma_l\mathbf{I})^{-1}\mathbf{v}_j, \;\; \alpha_l, \sigma_l \in \mathbb{C}, \tag{8}$$

where $N \in \{8, \ldots, 20\}$ in our case, and α_l, σ_l, $l = 1, \ldots, N$ are integration weights and shifts respectively.

In Davies and Higham (2005) it is shown that direct quadrature on (7) can be extremely inefficient, but through clever conformal mappings, Hale et al. (2008) developed midpoint quadrature rules that converge rapidly for increasing number of quadrature points. The maps needed depend on the extremal eigenvalues of the matrix $\mathbf{Q}$ and therefore need to be estimated. For the quadrature rules resulting from these mappings, the following theorem holds

Theorem 1. *(Hale et al., 2008) Let $\mathbf{Q}$ be a positive definite matrix with eigenvalues in $[\lambda_{min}, \lambda_{max}]$, then the N-point discretization formula developed in Hale et al. (2008) (equation 3.2) converges at the following rate*

$$\|\log \mathbf{Q} - f_N(\mathbf{Q})\| = \mathcal{O}(e^{-2\pi N/(\log(\lambda_{max}/\lambda_{min})+6)}) \tag{9}$$

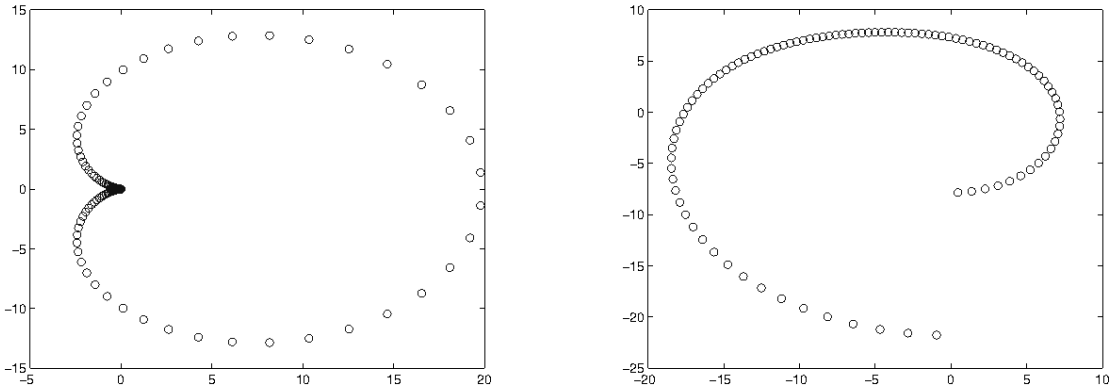
with f_N as in (8).

By the inequality $\|\log(\mathbf{Q})\mathbf{v} - f_N(\mathbf{Q})\mathbf{v}\| \leq \|\log \mathbf{Q} - f_N(\mathbf{Q})\|\|\mathbf{v}\|$, the theorem holds for functions of a matrix times a vector as well. This theorem can be used to determine the needed number of terms, N, in (8) required to achieve a certain accuracy. An example of the contour and shift produced by this method is illustrated in Figure 5. The conformal maps required for computing this quadrature rule, require the evaluation of the Jacobi elliptic functions, $\mathrm{sn}(\cdot|k), \mathrm{cn}(\cdot|k)$ and $\mathrm{dn}(\cdot|k)$ for complex arguments. These functions are in general difficult to compute. We use an approach similar to that in Driscoll (2009) to compute these.

The approximation of $\log(\mathbf{Q})\mathbf{v}$ in (8) is based on solving a family of shifted linear systems. The method of choice for computing $f_N(\mathbf{Q})\mathbf{v}_j$ is problem dependent, but in high dimensions, we usually have to rely on iterative methods, such as Krylov methods. Conjugate gradients (CG) is the most famous such method for solving $\mathbf{Q}\mathbf{x} = \mathbf{v}$, for a sparse $\mathbf{Q}$. This method solves for $\mathbf{x}$ by iteratively computing $\mathbf{Q}\mathbf{w}$, many times, for different $\mathbf{w}$. Generally, a Krylov subspace, $\mathcal{K}_k(\mathbf{Q}, \mathbf{v})$ is defined by $\mathcal{K}_k(\mathbf{Q}, \mathbf{v}) = \mathrm{span}\{\mathbf{v}, \mathbf{Q}\mathbf{v}, \mathbf{Q}^2\mathbf{v}, \ldots, \mathbf{Q}^k\mathbf{v}\}$, see e.g. Saad (2003). The Krylov method of choice is highly dependent on the condition number $\mathfrak{K}(\mathbf{Q}) = \lambda_{max}/\lambda_{min}$ of $\mathbf{Q}$, and the performance can often be improved by preconditioning the matrix $\mathbf{Q}$. The following inequality (Saad (2003)) illustrates the convergence of CG-type methods based on the condition number $\mathfrak{K}$:

$$\|\mathbf{x}^* - \mathbf{x}_k\|_\mathbf{Q} \leq 2\left(\frac{\sqrt{\mathfrak{K}} - 1}{\sqrt{\mathfrak{K}} + 1}\right)^k \|\mathbf{x}^* - \mathbf{x}_0\|_\mathbf{Q} \tag{10}$$

Figure 5: Contours (left) and shifts (right) for f_N in (8), with $N = 200$.



where $\mathbf{x}^*$ is the true solution of $\mathbf{Q}\mathbf{x}^* = \mathbf{y}$ and $\mathbf{x}_k$ is the kth approximation coming from the CG-method.

If the condition number $\mathfrak{K}(\mathbf{Q})$ is not too large, there are Krylov methods that are particularly well suited to compute the approximation in (8). These methods are based on the fact that $\mathcal{K}_k(\mathbf{Q}, \mathbf{v}) = \mathcal{K}_k(\mathbf{Q} - \sigma_l\mathbf{I}, \mathbf{v})$. This means that we can obtain the coefficients for the shifted systems in (8) without computing new matrix-vector products, see Jegerlehner (1996) and Frommer (2003) for details. We have employed the method CG-M in Jegerlehner (1996) for our implementation. One possible difficulty in employing the method is that we have complex shifts - this is remedied by using a variant, Conjugate Orthogonal CG-M (COCG-M), which entails using the conjugate symmetric form $(\overline{\mathbf{v}}, \mathbf{y}) = \mathbf{v}^T\mathbf{y}$ instead of the usual inner product $(\mathbf{v}, \mathbf{y}) = \overline{\mathbf{v}}^T\mathbf{y}$ in the Krylov iterations. See van der Vorst and Melissen (1990) for a description of the COCG method. In practice, little complex arithmetic is needed, since the complex, shifted coefficients are computed from the real ones obtained by the CG method used to solve $\mathbf{Q}\mathbf{v} = \mathbf{y}$. Note that for large $\mathfrak{K}$, this particular method may have poor convergence behaviour and it is difficult to precondition this method. In these cases, one is better of by solving the shifted systems in (8) in sequence using good preconditioners for $\mathbf{Q} - \sigma_l\mathbf{I}$.

This method also allows for an elegant way of computing the generalised log-determinant if we have an essentially intrinsic (singular) precision matrix. That is, the evaluation of

$$\widetilde{\log\det}\,\mathbf{Q} = \sum_{\substack{\lambda_i \in \sigma(\mathbf{Q}) \\ \lambda_i > 0}} \log \lambda_i. \tag{11}$$

To do this, we will need to implicitly deflate the eigenvectors corresponding to the zero eigenvalues. More, specifically, if $\boldsymbol{u}_j$, $j = 1, \ldots, r$ are the eigenvectors of $\mathbf{Q}$ associated with zero eigenvalues, we orthogonalise the probing vectors $\mathbf{v}_i$ to these eigenvectors by a Gram-Schmidt process and use these new probing vectors for computing $\widetilde{\log\det}\,\mathbf{Q}$. While we need accurate approximations to these eigenvectors for this procedure to work, they are often known from the modelling assumptions (see, for example, Chapter 3 in Rue and Held (2005)).

It is also possible to use this technique if we have a small cluster of eigenvalues that are very different (on a relative scale) to the other eigenvalues. Then we use the same approach as above, but we include the eigenvalues in our determinant evaluation, which leads to $\log\det\mathbf{Q} =$

$(\log\det\mathbf{Q})_{probe} + \sum_{j=1}^{r}\log\lambda_j$. While this approach has sound theory, one has to be careful so that round-off errors due to loss of orthogonality do not start to dominate. One remedy is to orthogonalise current estimator of $f_N(\mathbf{Q})\mathbf{v}_j$ in the Krylov method to the known eigenvectors at regular intervals. The cost of this orthogonalisation small.

3 Tools for effective approximation of the log-determinant

The method outlined in Section 2 can be used as a black-box procedure for well-conditioned matrices, for which COCG-M only requires a few iterations to converge. For not so well-conditioned matrices, the method is not a particularly fast one; solving hundreds of linear systems for computing one determinant approximation can be very time consuming, and therefore we should make the effort of tuning the method to the application at hand. Indeed, if it is possible to solve one set of shifted systems using fewer Krylov iterations, we should do so. Additionally, if we are able to shave off some probing vectors for a sufficiently accurate approximation, we should do so as well.

In the following subsections we propose various extensions of the basic methodology presented in Section 2 to facilitate special problems that may arise. These tricks are useful both for evaluating the potential of the approach and in practical implementations. First, we give some general advice on using the proposed log-determinant approximations. This advice also apply when using the numerous extensions proposed.

The most obvious way to reduce the number of Krylov iterations for convergence, is if $\mathbf{Q}$ is on some product form, $\mathbf{Q} = \prod_{i=1}^{K}\mathbf{Q}_i$. If there are repeated factors in the product, $i_j, j = 1,\ldots,J$, we note that $\log\det\prod_{j=1}^{J}\mathbf{Q}_{i_j} = J\log\det\mathbf{Q}_{i_1}$, and the conditioning of $\mathbf{Q}_{j_1}$ is better than that of the product. Additionally, some matrices may have determinants that are easy to compute, such as diagonal or tridiagonal matrices and can be separated from the approximation.

To shave off some probing vector, start using the approach above, looking at $\log(\mathbf{Q})\mathbf{e}_j$ for some js to find a k-distance colouring that is sufficient. Then compute the log-determinant, for a worst case scenario $\mathbf{Q}$ use a $(k-1)$-distance colouring for the probing vectors and see if the resulting determinant is (almost) the same as for the k-distance version. If they are, use the $(k-1)$-distance colouring instead, which should decrease the number of probing vector by a significant amount.

There are many situations in which $\mathbf{Q}$ does not depend on parameters, but rather is fixed. When this is the case it is obviously a good idea to precompute $\log\det\mathbf{Q}$ to high accuracy and use this stored value in the optimisation routine. In this case, only one or two determinant evaluations per iteration in the optimisation routine is needed, which greatly reduces the computational effort needed to find an optimum.

3.1 Off-diagonal compression using time-frequency transforms

The most common matrix functions have the property that the elements of $f(\mathbf{Q})$ decay quickly as they get farther from the diagonal (Benzi and Golub, 1999; Benzi and Razouk, 2007). However, the rate of decay often depends on the basis - that is, the elements of $f(\mathbf{W}\mathbf{Q}\mathbf{W}^{-1}) = \mathbf{W}f(\mathbf{Q})\mathbf{W}^{-1}$ may decay faster than those of $f(\mathbf{Q})$. In our context the rate of decay is very important: the faster the diagonal elements decay, the smaller we can take p. Therefore, the efficiency of our method is intimately tied to the decay properties of $f(\mathbf{Q})$ and, in this section, we consider some options for finding a good basis $\mathbf{W}$.

In particular, we can change the basis through a wavelet transform. The continuous wavelet

transform of a function $g \in L^2(\mathbb{R})$ is defined by through shifts and scalings of a mother wavelet $\phi \in L^2(\mathbb{R})$, namely $\phi_{u,s}(t) = \frac{1}{\sqrt{s}}\phi((t-u)/s)$, by

$$Wg(u,s) = \int_{\mathbb{R}} g(t)\overline{\phi}_{u,s}(t)dt, \tag{12}$$

provided that $\int_{\mathbb{R}} \phi(t)dt = 0$ and that $\int_{\mathbb{R}} |\hat{\phi}(\omega)|^2/\omega \; d\omega < \infty$. This transform can be discretized and has a fast version if g has compact support, called the fast wavelet transform. In the discretized setting, this corresponds to a basis change with another orthonormal basis (i.e. $\boldsymbol{W}^{-1} = \boldsymbol{W}^T$). This can also be generalised to multiple dimensions and on general manifolds. An introduction to wavelets can be found in Mallat (1998). Now, the properties of this transform that are interesting for our setting is exactly this: if the off-diagonal entries of $f(\mathbf{Q})$ possess some smoothness, which they almost always will when $\mathbf{Q}$ corresponds to a spatial prior, the entries in the transformed basis will have good decay. This is essentially the compression property of wavelet bases. While we consider wavelets here, the approach naturally extends to other transforms which may compress the off-diagonal entries and at the same time has a fast transform and inverse transform. Examples include curvelet transforms and Gabor frames (see Gröchenig (2001) and Candés et al. (2006)).

In our setting, we are not interested in using this as a preconditioner for solving linear systems, as is done in e.g. Chan et al. (1997), but rather to find a basis in which we need fewer probing vectors to make a sufficiently good log-determinant approximation. Since $\boldsymbol{W}(\mathbf{Q} - \sigma\mathbf{I})^{-1}\boldsymbol{W}^T = (\boldsymbol{W}\mathbf{Q}\boldsymbol{W}^T - \sigma\mathbf{I})^{-1}$ we do not need to modify our rational approximations to facilitate this new basis. The probing vectors do, however, need to be computed with respect to the new basis, which may be difficult to facilitate in a computationally efficient way. An empirical observation, however, suggests that it may be possible to use the probing vectors computed from the original precision matrix.

To illustrate how the decay behaviour may change, we compute $\log(\mathbf{Q})\boldsymbol{e}_{256}$ and $\log(\boldsymbol{W}\mathbf{Q}\boldsymbol{W}^T)\boldsymbol{e}_{256}$ for a 1-D lag-2 random walk model using the Daubechies 2 wavelet (the compact wavelet with fewest non-zero entries with vanishing moments of order 0, 1 and 2). The result is illustrated in Figure 6. In this RW-2 model, the $\log\det\mathbf{Q} = 2.8347 \cdot 10^3$, the 1-distance colouring in the wavelet approximation, corresponding to 27 colours gives $\log\det\mathbf{Q} \approx 2.8318 \cdot 10^3$, while the 17-distance colouring (30 colours) in the original basis gives $\log\det\mathbf{Q} \approx 2.6893 \cdot 10^3$. In the original basis, we require a 169-distance colouring, corresponding to 172 colours in order to match the approximation accuracy in the wavelet basis.

Now, computing $\boldsymbol{W}\mathbf{Q}\boldsymbol{W}^T\mathbf{v}$ for arbitrary $\mathbf{v}$s can be done without forming the matrix $\boldsymbol{W}\mathbf{Q}\boldsymbol{W}^T$ by using the fast wavelet transform, and while we need to form an approximation to $\boldsymbol{W}\mathbf{Q}\boldsymbol{W}^T$ in order to compute the probing vectors, it will be faster to use the fast wavelet transform in the matrix vector product case. This certainly suggests that for specific problems, where $\log\mathbf{Q}$ has some smoothness in its off-diagonal terms, this may be an approach to pursue.

3.2 Nodal nested dissection reordering and recursive computation

When computing the log-determinant of a precision matrix using the Cholesky approach, we should always do a fill-in reordering of the precision matrix before computing the Cholesky factor. In effect, we then compute $\text{chol}(\boldsymbol{P}\mathbf{Q}\boldsymbol{P}^T)$, where $\boldsymbol{P}$ is a permutation matrix. A particularly well-suited reordering is the METIS nodal nested dissection reordering (Karypis and Kumar (1999)). Figure 7 illustrates a typical sparsity structure coming from employing this reordering.

While this type of reordering allows for fill-in that is close to minimal, it also allows for recursive computation of the log-determinant via a nested Schur-complement technique. Take the following

Figure 6: Decay behaviour of wavelet basis versus normal basis.

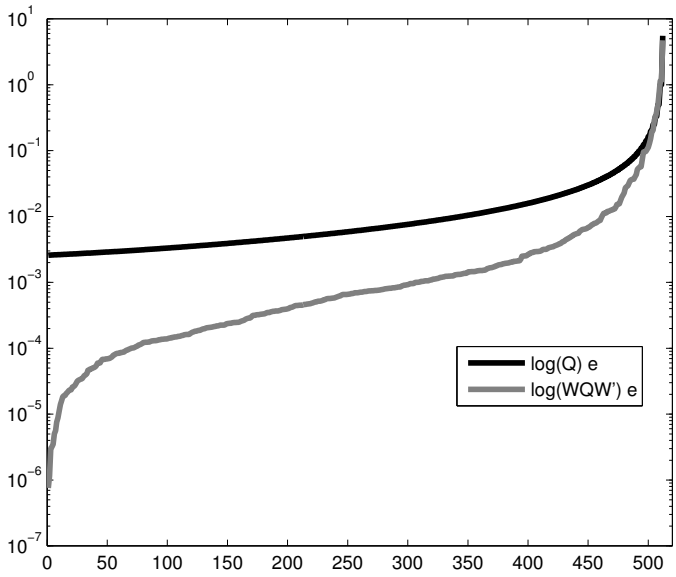
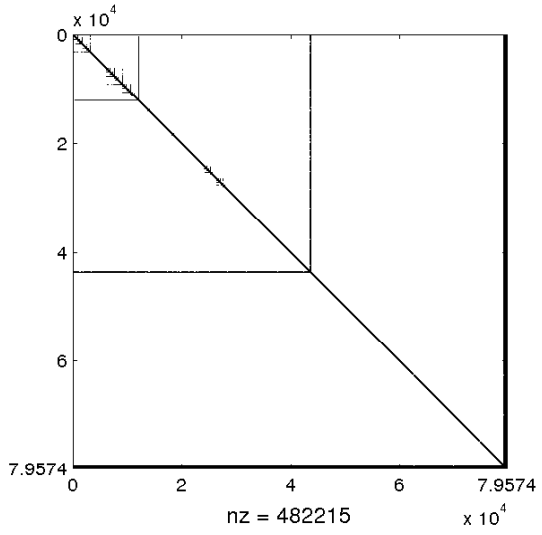


Figure 7: Example of nested dissection reordering. The non-zero elements are very centered near the diagonal, except for a small number of variables that are coupled with all predecessors.



block matrix, corresponding to the general block form of a matrix that has undergone nodal nested dissection reordering,

$$\mathbf{Q} = \begin{pmatrix} \mathbf{A}_1 & \boldsymbol{F}_{1,1} & \boldsymbol{O} & \boldsymbol{F}_{2,1} \\ \boldsymbol{F}_{1,1}^T & \boldsymbol{B}_1 & \boldsymbol{O} & \boldsymbol{F}_{2,2} \\ \boldsymbol{O} & \boldsymbol{O} & \mathbf{A}_2 & \boldsymbol{F}_{2,3} \\ \boldsymbol{F}_{2,1}^T & \boldsymbol{F}_{2,2}^T & \boldsymbol{F}_{2,3}^T & \boldsymbol{B}_2 \end{pmatrix}, \tag{13}$$

and let $\boldsymbol{F}_2 = \begin{pmatrix} \boldsymbol{F}_{2,1}^T & \boldsymbol{F}_{2,2}^T & \boldsymbol{F}_{2,3}^T \end{pmatrix}^T$, $\mathbf{Q}_1 = \begin{pmatrix} \mathbf{A}_1 & \boldsymbol{F}_{1,1} & \boldsymbol{O} \\ \boldsymbol{F}_{1,1}^T & \boldsymbol{B}_1 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \mathbf{A}_2 \end{pmatrix}$ and let the block Schur complements be $\boldsymbol{S}_1 = \boldsymbol{B}_1 - \boldsymbol{F}_{1,1}^T \mathbf{A}_1^{-1} \boldsymbol{F}_{1,1}$ and $\boldsymbol{S}_2 = \boldsymbol{B}_2 - \boldsymbol{F}_2^T \mathbf{Q}_1^{-1} \boldsymbol{F}_2$. Then we can compute the log-determinant of $\mathbf{Q}$ in the following recursive manner,

$$\log\det \mathbf{Q} = \log\det \mathbf{Q}_1 + \log\det \boldsymbol{S}_2 = \log\det \mathbf{A}_1 + \log\det \boldsymbol{S}_1 + \log\det \mathbf{A}_2 + \log\det \boldsymbol{S}_2. \tag{14}$$

This obviously extends to arbitrary levels of recursion, say k. The key elements in this recursive way of computing the log-determinant are 1) we can use Krylov methods to compute $\boldsymbol{F}_2^T \mathbf{Q}_1^{-1} \boldsymbol{F}_2$ and its upper level counterparts. This requires the solution of some linear systems that do not need to be stored. 2) $\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_k$ are typically low dimensional, and we can use direct methods for computing their log-determinants, and 3) we can use the determinant approximations of the previous section for computing $\log\det \mathbf{A}_i$, and the condition numbers and the distance colourings required for the $\mathbf{A}_i$s are typically much smaller than for the original system.

The question then is: when do we need to use this recursive approach rather than using the matrix function approach directly? The obvious situation in which to apply this extension is when, after reordering the matrix $\mathbf{Q}$, the last block matrix $\boldsymbol{B}_k$ is very small, and the conditioning of $\mathbf{Q}_1$ is much better than the original $\mathbf{Q}$. Then this approach should be orders of magnitude faster than using the direct approximation on $\mathbf{Q}$. Another situation is when it is very hard to compute $\log\det \mathbf{Q}$ or $\log\det \mathbf{Q} - \log\det(\mathbf{Q} + \lambda^2 \mathbf{A}^T \mathbf{A})$. Then this approach may be prudent, in this case to increase the accuracy of the log-determinant approximations without them taking much more time.

3.3 Subtractive cancellation for log-determinants

Suppose that the quantity of computational interest is given by

$$f(\boldsymbol{\eta}, \lambda^2) = \log\det(\mathbf{Q}_{\boldsymbol{\eta}}) - \log\det(\mathbf{Q}_{\boldsymbol{\eta}} + \lambda^2 \boldsymbol{K}) + q(\boldsymbol{\eta}, \lambda^2) \tag{15}$$

for some well conditioned $\boldsymbol{K}$, and where $q(\boldsymbol{\eta}, \lambda^2)$ is shorthand for the quadratic forms and potential prior distributions involving $\boldsymbol{\eta}, \lambda^2$. This happens when we have noisy observations of a Gaussian field of some sort and we want to find the posterior distribution or compute the maximum likelihood estimate for $\boldsymbol{\eta}, \lambda^2$, as in (3). When computing $f(\boldsymbol{\eta})$, it appears that $\log\det(\mathbf{Q}_{\boldsymbol{\eta}})$ is over-/underestimated while $\log\det(\mathbf{Q}_{\boldsymbol{\eta}} + \lambda^2 \boldsymbol{K})$ is under-/overestimated comparatively, so that the relative error in $\log\det(\mathbf{Q}_{\boldsymbol{\eta}}) - \log\det(\mathbf{Q}_{\boldsymbol{\eta}} + \lambda^2 \boldsymbol{K})$ is greater than for each of the quantities individually. There also appears to be additional effects, as can be seen for the case where $\kappa = 0.001, \lambda^2 = 0.5$. In the numerical litterature, this is known as subtractive cancellation. This effect may lead to problems in optimisation procedures where this difference needs to be computed several and wildly different $\boldsymbol{\eta}, \lambda^2$. In computational terms, it essentially means that we will need more probing vectors to accurately compute this difference than to accurately compute its individual ingredients.

Table 1: Relative accuracy for log-determinants of precision matrices, perturbations of these and their differences.

	$\log\det \mathbf{Q}_\kappa^2$	$\log\det(\mathbf{Q}_\kappa^2 + \lambda^2\mathbf{I})$	Difference
$\kappa = 0.001, \lambda^2 = 0.1$	1.01411	0.99997	0.75638
$\kappa = 0.005, \lambda^2 = 0.1$	1.00714	0.99996	0.85185
$\kappa = 0.01, \lambda^2 = 0.1$	1.00468	0.99996	0.89247
$\kappa = 0.05, \lambda^2 = 0.1$	1.00098	0.99995	0.96623
$\kappa = 0.001, \lambda^2 = 0.05$	1.01410	0.99980	0.68790
$\kappa = 0.005, \lambda^2 = 0.05$	1.00714	0.99980	0.79878
$\kappa = 0.01, \lambda^2 = 0.05$	1.00468	0.99980	0.84818
$\kappa = 0.05, \lambda^2 = 0.05$	1.00098	0.99984	0.94338
$\kappa = 0.001, \lambda^2 = 0.5$	1.01411	1.00001	0.87264
$\kappa = 0.005, \lambda^2 = 0.5$	1.00714	1.00001	0.92890
$\kappa = 0.01, \lambda^2 = 0.5$	1.00468	1.00001	0.95090
$\kappa = 0.05, \lambda^2 = 0.5$	1.00098	1.00001	0.98751

To illustrate this effect, we utilise a 2-D Matérn field with indirect observation on each discretization point; i.e. we discretize

$$(\kappa - \triangle)x = \mathcal{W} \tag{16}$$

and add upon it i.i.d noise to obtain our observations. Here $\triangle = \sum_i \partial/\partial x_i^2$ is the Laplacian operator on $\mathbb{R}^d$. If $\mathbf{Q}_\kappa^2$ denotes the discretized precision obtained from (16), the perturbed matrix becomes $\mathbf{Q}_\kappa^2 + \lambda^2\mathbf{I}$. In Table 1 we can see this effect, and this is typical for these type of models. Better conditioning of both matrices (corresponding to higher κ and λ^2) leads to less subtractive cancellation and worsening of the conditioning leads to greater amounts of subtractive cancellation. Specifically, when $\kappa = 0.001, \lambda^2 = 0.05$, the differences of the log-determinants are too inaccurate to perform optimisation on the parameters, and we need to have sufficient accuracy in the entire range of possible values for the parameters for an optimisation routine to stably find the correct optimum.

3.4 Random sign flipping of entries of probing vectors for monotone covariance functions

In spatial modeling, it is common to in advance know if the precision matrix induces a monotone, decreasing function off the diagonal of the covariance matrix. This is the case for Matérn type covariance functions and many other used in a wide range of spatial models (see e.g. Cressie (1993)). It may also be known in graphical models that the correlation of nodes remain positive throughout the graph. In this particular setting, it is possible to refine the probing vectors in order to achieve greater accuracy with fewer vectors. To see this, note that if $\boldsymbol{u}_k = -\boldsymbol{e}_k$,

$$\boldsymbol{u}_k^T f_N(\mathbf{Q})\boldsymbol{u}_k = \boldsymbol{e}_k^T f_N(\mathbf{Q})\boldsymbol{e}_k, \tag{17}$$

and we could have replaced all probing vectors with their negatives and recovered the same approximation. Now, let $\mathbf{v}_k, k = 1, \ldots, s$ be the probing vectors computed with the graph colouring

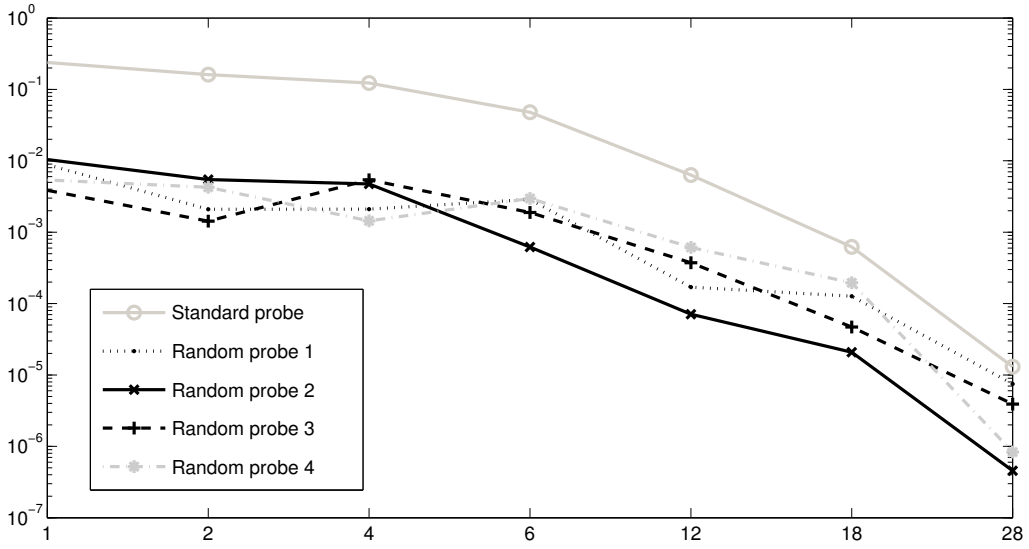
Table 2: Relative accuracy for log-determinants of precision matrices, perturbations of these and their differences. Now using random flipping in probing vectors.

	$\log\det \mathbf{Q}_\kappa^2$	$\log\det(\mathbf{Q}_\kappa^2 + \lambda^2\mathbf{I})$	Difference
$\kappa = 0.001, \lambda^2 = 0.1$	1.00262	1.00008	0.95061
$\kappa = 0.005, \lambda^2 = 0.1$	1.00227	1.00008	0.95446
$\kappa = 0.01, \lambda^2 = 0.1$	1.00200	1.00009	0.95766
$\kappa = 0.05, \lambda^2 = 0.1$	1.00113	1.00011	0.96962
$\kappa = 0.001, \lambda^2 = 0.05$	1.00262	1.00020	0.93674
$\kappa = 0.005, \lambda^2 = 0.05$	1.00227	1.00021	0.94097
$\kappa = 0.01, \lambda^2 = 0.05$	1.00200	1.00021	0.94470
$\kappa = 0.05, \lambda^2 = 0.05$	1.00113	1.00024	0.95969
$\kappa = 0.001, \lambda^2 = 0.5$	1.00262	0.99989	0.97432
$\kappa = 0.005, \lambda^2 = 0.5$	1.00227	0.99989	0.97679
$\kappa = 0.01, \lambda^2 = 0.5$	1.00200	0.99990	0.97871
$\kappa = 0.05, \lambda^2 = 0.5$	1.00113	0.99991	0.98539

approach, and let some of the entries of $\mathbf{v}_k$ be flipped to -1. We propose the following approach: If $\mathbf{v}_j(i) = 1$, set $\mathbf{v}_j(i) = -1$ with probability $1/2$. We motivate this heuristic as follows: Given a non-zero entry in a probing vector, e.g. $\mathbf{v}_j(i) = 1$, then surrounding ones in the same probing vector will all contribute positively or negatively to the entry so that $(f_N(\mathbf{Q})\mathbf{v}_j)(i) = f_N(\mathbf{Q})_{ii} + \epsilon$, where ϵ denotes errors accumulating from nearby ones. If, however, some of the surrounding ones are flipped to minus one, some of this error will cancel locally. Moreover, since we are interested in the sum of many quadratic forms $\mathbf{v}_j^T f_N(\mathbf{Q})\mathbf{v}_j$, a global cancellation also occurs. One can see this approach as a synthesis of the original Hutchinson estimator (Hutchinson (1989)), in which the vectors have entries $+1$ or -1 with probability $1/2$ and the basic probing approach in Tang and Saad (2010). It appears that this synthesis greatly improves upon the accuracy of the log-determinant approximations, and it also seems to partially remove the effect of subtractive cancellation. To illustrate this, we use the same model as in Section 3.3, and reproduce Table 1 and give them in Table 2. We also note that producing this table requires a 4-distance colouring, whereas the previous one required a 14-distance colouring, so using randomised entries in the probing vectors both reduces the number of probing vectors required and eliminates some of the subtractive cancellation.

Even though the heuristic suggested above does not immediately carry over to precision matrices inducing oscillating covariance functions, it appears that using this randomised approach still gives better approximations than not using it. We illustrate this by using the a stationary covariance function that oscillates and induces a sparse precision matrix. In Figure 8, we see the effect of using randomised probing vectors vs. the standard ones. Considering these observations, it becomes quite clear that randomly flipping entries in the probing vectors should be the default behaviour for computing these log-determinant approximations. It may be that in some cases, it is possible to compute the optimal distribution of $+1$ and -1 in the probing vectors, but how to do this is not obviously clear in all situations. The randomised version is therefore a good default choice. The observations made here also suggests that randomised probing vectors is compatible with the wavelet approach discussed in Section 3.1.

Figure 8: Standard versus random probing vectors for an oscillating covariance function.



3.5 Model variants with precision $(\boldsymbol{K} + \kappa^2\boldsymbol{C})\boldsymbol{B}_1^{-1}(\boldsymbol{K} + \kappa^2\boldsymbol{C}) \cdots \boldsymbol{B}_k^{-1}(\boldsymbol{K} + \kappa^2\boldsymbol{C})$

When doing optimisation using high-dimensional determinant approximations, it is important to use whatever structure that is available in order to speed up computations. The approach outlined in this article is not always fast, and if it is possible optimise some aspects of computation for special models, we should do so.

In particular, for precision matrices of the kind

$$\mathbf{Q} = (\boldsymbol{K} + \kappa^2\boldsymbol{C})\boldsymbol{B}_1^{-1}(\boldsymbol{K} + \kappa^2\boldsymbol{C}) \cdots \boldsymbol{B}_k^{-1}(\boldsymbol{K} + \kappa^2\boldsymbol{C}), \tag{18}$$

which for instance arises in the SPDE approach in Lindgren et al. (2011), it is possible to compute the partial derivative with respect to κ^2 at almost no extra cost. To see this, note the following calculations

$$\begin{aligned}
&\frac{\partial}{\partial\kappa^2} \log\det(\boldsymbol{K} + \kappa^2\boldsymbol{C})\boldsymbol{B}_1^{-1}(\boldsymbol{K} + \kappa^2\boldsymbol{C}) \cdots \boldsymbol{B}_k^{-1}(\boldsymbol{K} + \kappa^2\boldsymbol{C}) \\
&\quad = \frac{\partial}{\partial\kappa^2} \left((k+1)\log\det(\boldsymbol{K} + \kappa^2\boldsymbol{C}) + \sum_{j=1}^{k} \log\det \boldsymbol{B}_j \right) \\
&\quad = (k+1)\ \mathrm{tr}\left((\boldsymbol{K} + \kappa^2\boldsymbol{C})^{-1}\boldsymbol{C}\right).
\end{aligned} \tag{19}$$

First note that the matrix vector products, $(\boldsymbol{K} + \kappa^2\boldsymbol{C})\mathbf{v}_i$ are exactly those needed to compute the log-determinant. The trace approximation then follows directly from the diagonal inverse approximation in Tang and Saad (2010). If $\boldsymbol{C}\mathbf{v}_i$ is relatively cheap to compute, as happens if $\boldsymbol{C}$ is, e.g. diagonal, this partial derivative comes at essentially no extra cost.

Similarly, if we have an observation matrix, $\mathbf{A}$ after which i.i.d. noise is added, we need to compute $\log\det(\mathbf{Q}+\eta^2\mathbf{A}^T\mathbf{A})$. Then we obtain the following partial derivatives

$$\frac{\partial}{\partial\kappa^2}\log\det(\mathbf{Q}+\eta^2\mathbf{A}^T\mathbf{A}) = \text{tr}\left((\mathbf{Q}+\eta^2\mathbf{A}^T\mathbf{A})^{-1}(k+1)\boldsymbol{C}(\boldsymbol{K}+\kappa^2\boldsymbol{C})^k\prod_{i=1}^{k}\boldsymbol{B}_i^{-1}\right) \tag{20}$$

by the cyclic property of the trace operator for symmetric matrices, and

$$\frac{\partial}{\partial\eta^2}\log\det(\mathbf{Q}+\eta^2\mathbf{A}^T\mathbf{A}) = \text{tr}\left((\mathbf{Q}+\eta^2\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A}\right). \tag{21}$$

In computations, the matrix vector product $(\mathbf{Q}+\eta^2\mathbf{A}^T\mathbf{A})\mathbf{v}$ needs to be computed for the determinant approximation. Hence $\mathbf{A}^T\mathbf{A}\mathbf{v}$ needs to be cheap if the second expression is to compute at low costs. The first of these, however, is a bit more complicated, but observe that if $k=1$ and $\boldsymbol{B}_1^{-1}$ is diagonal, we can have $(\boldsymbol{K}+\kappa^2\boldsymbol{C})\mathbf{v}$ from (19), provided that the probing vectors are equal, and by definition the $\boldsymbol{B}_1^{-1}\mathbf{v}$ is cheap to compute. Hence it is possible to compute the gradient in an optimisation routine on the fly while computing the objective function at little extra cost, and the computational requirements for a Newton-type algorithm is easily decreased to a fraction between 1/2 and 1/3 compared to the one where finite differences are used for gradient computations. We also note that these observations are compatible with the wavelet compression approach discussed in Section 3.1, and the random flipping of entries in probing vectors in 3.4.

4 Examples

In this section we apply the approximate log-determinant methods to parameter estimation on three examples. The examples are chosen to emphasise both the nice properties and challenges that occur in practical implementations. In the notation here, we assume that $\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},(\sigma^2/\lambda^2)\mathbf{Q}_{\boldsymbol{\eta}}^{-1})$, and $\boldsymbol{y}=\boldsymbol{A}_{\boldsymbol{\theta}}\boldsymbol{x}+\sigma\,\mathcal{N}(\mathbf{0},\mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}}^{-1})$, with essentially $\boldsymbol{A}_{\boldsymbol{\theta}}=\boldsymbol{A}$, $\mathbf{Q}_{\boldsymbol{\eta}}=\mathbf{Q}_{\kappa^2}$, $\mathbf{Q}_{\boldsymbol{\epsilon},\boldsymbol{\eta}}=\boldsymbol{I}$ in subsequent sections. This corresponds to a SPDE model with iid observations on top of it.

We compare the estimates using the approach explored in the previous sections with those obtained by a block composite likelihood approach, see Eidsvik et al. (2011). The main idea behind composite likelihoods is to replace the computationally demanding likelihood expression with several block-type expressions. Each term requires less memory and computational time. Thus, rather than working with the full likelihood function $\log p(\mathbf{y}|\boldsymbol{\eta},\boldsymbol{\theta},\sigma^2,\lambda^2)$, which in the Gaussian setting is given by (3), the block composite likelihood approach adds up Gaussian composite terms built from block interactions.

In essence, partition the domain $\mathcal{D}$ into pairwise disjoint subdomains; $\mathcal{D}_i\cap\mathcal{D}_j=\emptyset, i\neq j$ and $\bigcup_{i=1}^{M}\mathcal{D}_i=\mathcal{D}$. Thereafter, assume that the only interaction terms are between neighboring blocks. Let $\mathbf{y}_k$ and $\mathbf{y}_l$ be the data in domains $\mathcal{D}_k$ and $\mathcal{D}_l$. Then, the block composite likelihood is available by

$$\begin{aligned} 2\log p_{CL}(\mathbf{y}|\boldsymbol{\eta},\boldsymbol{\theta},\sigma^2,\lambda^2) &= \sum_{k=1}^{M-1}\sum_{l>k}2\log p(\mathbf{y}_{kl}|\boldsymbol{\eta},\boldsymbol{\theta},\sigma^2,\lambda^2) \\ &= \sum_{k=1}^{M-1}\sum_{l>k}\left(\log\det\mathbf{Q}_{y,kl}-(\mathbf{y}_{kl}-\boldsymbol{\mu}_{y,kl})^T\mathbf{Q}_{y,kl}(\mathbf{y}_{kl}-\boldsymbol{\mu}_{y,kl})\right) \end{aligned} \tag{22}$$

where $\boldsymbol{\mu}_{y,kl}$ is the mean of block variables $(\mathbf{y}_k^T, \mathbf{y}_l^T)^T$ and $\mathbf{Q}_{y,kl}^{-1} = \text{Cov}(\mathbf{y}_k, \mathbf{y}_l)$ is the covariance matrix for this block pair.

The maximum composite likelihood estimators are the parameter values $(\boldsymbol{\eta}, \boldsymbol{\theta}, \sigma^2, \lambda^2)$ that optimize expression (23). Theoretical considerations and computational approaches for this block composite likelihood model can be found in Eidsvik et al. (2011).

4.1 3-D Matérn field with direct and indirect observations

In spatial statistics, it is fairly common to assume that an underlying spatial field comes from the Matérn family or to use a Gaussian prior coming from the same family. The underlying field or prior field is then described by the fractional Laplacian

$$(\kappa^2 - \triangle)^{\alpha/2} x = \mathcal{W}, \tag{23}$$

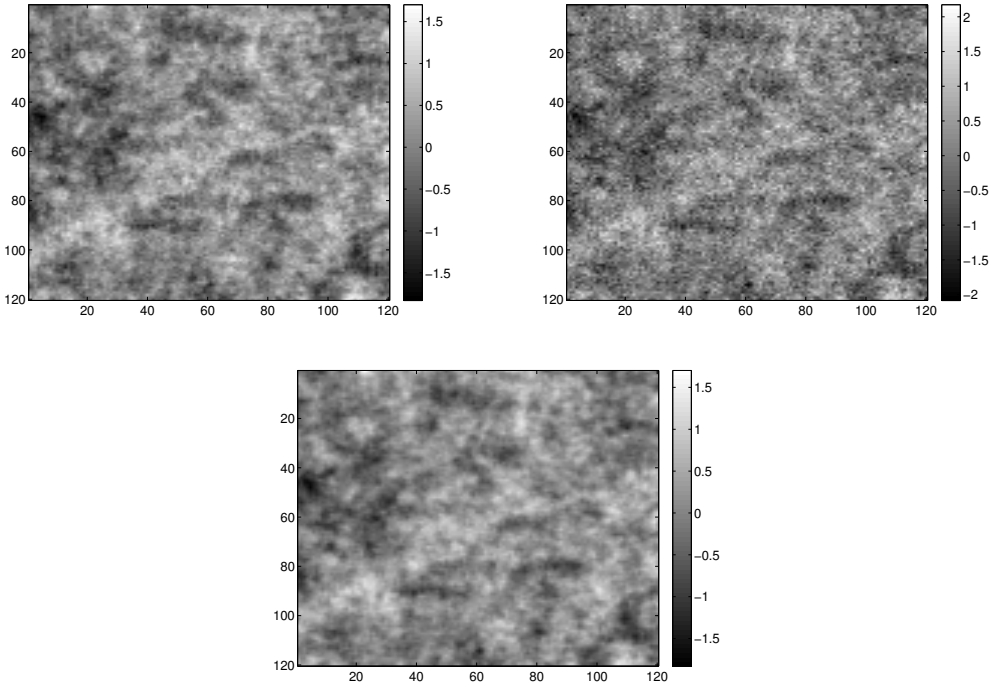
where $\mathcal{W}$ is spatial white noise. The connection of this representation and Matérn covariance functions, can be found in Lindgren et al. (2011). We mention that from a physical point of view, the same article makes good arguments for using Neumann boundary instead of imposing artificial boundary conditions corresponding to completely unchanging marginal variances at each site. If we observe x directly, we have direct observations, and we only need to compute one log-determinant for the likelihood evaluation and we avoid the previously discussed effects of subtractive cancellation. If we have an observation process on top of x, we are in the setting of (3), and we have the problem described in Section 3.3. In Figure 9 we see a slice of the direct observations and a slice of the corresponding indirect observations, as well as a reconstruction from the indirect observations.

In our example, we assume that we gradually observe more sites of the total field, from 15^3 sites to 120^3 sites.

Table 3: Estimation of precision parameters in a Matérn field with respect to distance colouring and observed part of field. This is for the situation with direct observations.

	4-distance		8-distance		16-distance	
	τ^2	κ^2	τ^2	κ^2	τ^2	κ^2
15^3	0.98362	0.23740	1.00158	0.17137	1.00675	0.15188
30^3	0.97116	0.18467	0.99003	0.11396	0.99783	0.08355
60^3	0.97135	0.18442	0.99147	0.10676	1.00067	0.06988
120^3	0.96716	0.18112	0.98759	0.10131	0.99741	0.06152

Figure 9: Direct (left) and indirect (right) observations of Matérn field, and a reconstructed field (bottom).



4.1.1 Direct observations

For the rare case where direct (non-noisy) observations are available, the log-likelihood for the Gaussian represents the objective function, presuming no prior information on the parameters to estimate is available. In this setting, two parameters need to be estimated, κ^2, representing the range and a scaling parameter τ^2 representing the level of the realisation. In Table 3, we see the effect of using different distance colourings of the precision matrix and observing smaller and larger parts of the field. The true parameters were $\tau^2 = 1, \kappa^2 = 0.05$.

Table 4: Estimation of precision parameters in a Matérn field with respect to distance colouring and observed part of field. This is for the situation with indirect observations.

	4/5-distance			8/9-distance			16/17-distance		
	λ^2	κ^2	σ^2	λ^2	κ^2	σ^2	λ^2	κ^2	σ^2
15^3	0.0822	0.2714	0.0966	0.1179	0.1423	0.1040	0.1286	0.10587	0.1055
30^3	0.0667	0.2431	0.0925	0.0941	0.1193	0.1001	0.1042	0.07595	0.1020
60^3	0.0633	0.2559	0.0906	0.0906	0.1189	0.0984	0.0989	0.06943	0.1007
120^3	0.0601	0.2603	0.0894	0.0861	0.1191	0.0972	0.0968	0.06520	0.0994

4.1.2 Indirect observations

Suppose that the discretized field $\mathbf{x}$ generated by (23) has the observation process $\mathbf{y} = \mathbf{x} + \sigma^2\boldsymbol{\epsilon}$ attached to it, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then we optimise (3) for the parameters $\boldsymbol{\eta} = (\kappa^2, \tau^2, \sigma^2)$ in the corresponding precision matrices. In addition to generating a table of the estimated parameters for different distance colourings, we compare them with parameters obtained from the block composite likelihood method. In order to obtain comparable accuracy between for the two log-determinant evaluations in (3), we needed to use a larger distance colouring for the perturbed matrix, reflected by n_1/n_2 in Table 4. In addition, we use the reparametrisation $\lambda^2 = \tau^2\,\sigma^2$, since this is beneficient in the optimisation routine. The true parameters are $\sigma^2 = 0.1, \lambda^2 = 0.1$ and $\kappa^2 = 0.05$, and we see that we recover these well by observing more of the field and using more probing vectors. In order to compare our results with those resulting from the block composite likelihood procedure, some care must be taken: the parameter we estimate in this model are not completely equivalent to those coming from using covariance functions. In our case, we have the SPDE from the previous section, given by (23) for $\alpha = 2$ and the corresponding exponential covariance function in three dimensions,

$$C(r) = \gamma^2 e^{-r/\phi}. \tag{24}$$

In particular, the marginal variance parameter, γ^2 for the field is estimated in the composite likelihood approach, while in the SPDE model, using the natural Neumann boundary conditions, the entries of $\mathbf{Q}_{ii}^{-1}$ differ, depending on how far i is from the boundary. Now, $\mathrm{tr}\,(\mathbf{Q}^{-1})/n$ gives a natural estimate for the marginal variance parameter for the overall field and should be comparable to that coming from the composite likelihood approach. Similarly, the range parameter ϕ has its relative in the parameter κ^2, but here there is also no direct correspondence. A natural surrogate in this case is the correlation length, ℓ, which can be computed from the probing vectors. The parameter σ^2 is directly comparable between the two models.

A comparison of estimates achieved by approximating the log-determinant and the composite likelihood estimation is shown in Table 5. The estimates are very similar. It appears as if the composite likelihood returns slightly larger range values ℓ and signal to noise γ^2, while the measurement noise level σ^2 is a little smaller. For the log-determinant approach there seems to be a monotone trend in all the parameters when observing more of the field. This is a desirable property that does not seem to hold for the composite likelihood approach. In Table 5, there is a bad value for the correlation distance in dimensions 15^3 - this is a consequence of the discretization being so coarse that it is impossible to properly adjudicate it.

Table 5: A comparison of block composite likelihood and the determinant approximation. The γ relates to signal to noise ratio, the ℓ is the correlation range parameter and σ is the measurement noise standard deviation.

	Comp. lik.			log-det approx.		
	γ^2	ℓ	σ^2	γ^2	ℓ	σ^2
15^3	0.455^2	9.86	0.302^2	0.485^2	27.3	0.325^2
30^3	0.464^2	11.3	0.309^2	0.467^2	10.7	0.319^2
60^3	0.453^2	11.2	0.309^2	0.437^2	10.5	0.317^2
120^3	0.450^2	10.7	0.306^2	0.425^2	10.3	0.315^2

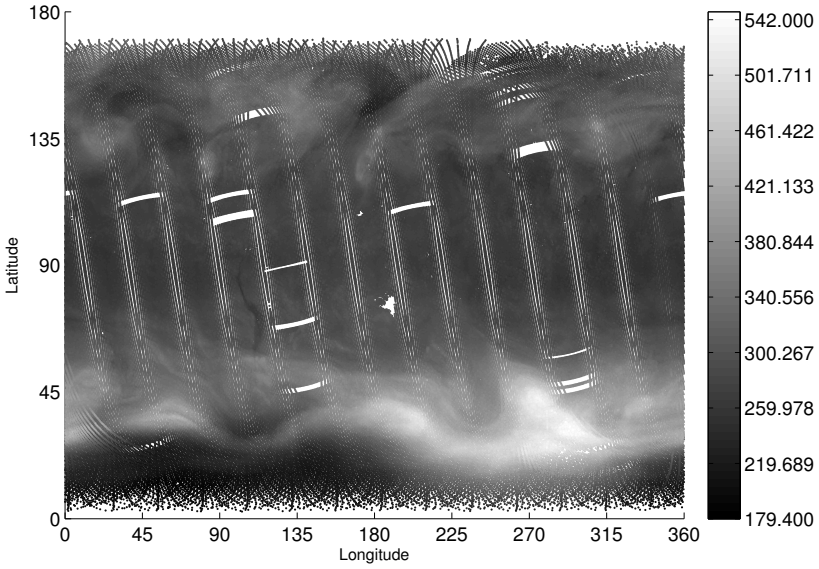
4.2 Ozone column estimation

In this example, we analyse total column ozone (TCO) data acquired from an orbiting satelite. This is a popular dataset that has been analysed in Cressie and Johannesson (2008) using a fixed rank Kriging approach and in Eidsvik et al. (2011) using the block composite likelihood method. The dataset has also been modeled using a nested SPDE approach in Bolin and Lindgren (2011). What is special about this dataset is that it is 1) on the sphere and 2) since the data is acquired along the transects of the satelite and therefore a rather special sampling pattern is obtained.

We use the SPDE approach as in the previous sections, only this time on a sphere. We use a uniform triangulation on the sphere for discretizing the SPDE. This gives us a different observation process than in the previous section: the observed data is given by $\boldsymbol{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}$, where the matrix $\mathbf{A}$ interpolates from the uniform triangulation on the sphere to the observation pattern given by the satelite. Our discretization consists of 324 002 points on the sphere, and we have 173 405 observations. Since the observations are not snapshots of the globe at a given time, we also get temporal effects in the data. We do not model this temporal effect. An illustration of the observations is given in Figure 10.

The estimated parameters are $\lambda^2 \approx 0.0117^2$, $\kappa^2 \approx 1.61^2$ and $\sigma^2 \approx 5.015^2$. Using the same tricks as in the last section, this converts to, $\gamma^2 \approx 55.3$, $\ell \approx 10\,567$ km. In comparison, the block composite likelihood model, with 15×15 blocks in latitude and longitude, gives the $\gamma^2 \approx 73.6^2$, $\ell \approx 7028$ km, and $\sigma^2 \approx 4.7^2$, which is not very dissimilar. We mention that the blocking in the block composite likelihood may not be sufficient to capture all large scale variations.

Figure 10: Total column ozone observations (dots) acquired along satellite orbits. There are 173 405 measurements in total.



5 Discussion

In this paper, we have presented a new method for performing statistical inference on Gaussian models that are too large for conventional techniques to work. Focussing on the problem of computing likelihoods for large Gaussian random vectors, we have shown that by combining a number of approximation techniques, we can evaluate the likelihood of truly large models in a reasonable amount of time with reasonable accuracy. In particular, we have shown that a combination of function approximation, graph theory, wavelet methods, modern numerical linear algebra, and problem specific tricks are necessary when a problem is so large that Cholesky factorisations are no longer feasible. This explosion of complexity, which indicative of the difficulty of the problem, comes at the advantage that *we can actually solve these models*, which is not possible using standard techniques. Furthermore, when combined with the work of Simpson et al. (2008); Simpson (2008); Aune et al. (2012), this work completes a suite of methods for performing statistical calculations with large Gaussian random vectors using Krylov subspace methods.

5.1 Extensions and future work

An article that inspired this work in many ways is the work on using probing vectors for finding the diagonal of the matrix inverse (Tang and Saad (2010)). In this approach, the entire diagonal is wanted - not just its sum - and hence a variant expression is needed, namely

$$\operatorname{diag} f(\mathbf{Q}) = \left(\sum_{j=1}^{n} \mathbf{v}_j^T \odot f(\mathbf{Q})\, \mathbf{v}_j \right) \oslash \left(\sum_{j=1}^{n} \mathbf{v}_j \odot \mathbf{v}_j \right), \tag{25}$$

where $\oslash$ and $\odot$, respectively, are elementwise division and multiplication of vectors. Using the same probing vectors as those needed for the determinant, and $f(t) = t^{-1}$ will then yield an estimate for the diagonal of the inverse of the precision matrix, i.e. the marginal variances. Note that it is much easier to compute the diagonal of the inverse using probing vectors, since in this case we are dealing with the matrix inverse. Preconditioning can therefore be applied directly, and in since we need to compute $\mathbf{Q}^{-1}\mathbf{v}$ for quite some vectors, traditionally expensive preconditioners can be worth it. Specifically, combination of AINV (see. Huckle and Grote (1997)) and wavelet compression may be well suited for extracting this diagonal. In this situation, we get a dual benefit from the wavelet compression: it may both improve upon the AINV preconditioner and decrease the number of probing vectors needed.

The marginal variances together with the log-determinant (required for optimisation) are components needed in doing inference by the integrated nested Laplace approximation (INLA) (Rue et al. (2009)), and the approach given in this paper may be a way to extend the INLA approximation to larger models than can be handled with the current direct methods.

Another potential application of (25) is the computation of communication in graphical models, such as social networks and networks of oscillators (Estrada et al. (2011)). Using the matrix exponential or relatives as the map, the diagonal of this is a measure for self-communicability or subgraph centrality, which is used in analysis of complex networks. Naturally, a matrix-vector type method is needed for the action $\exp(\alpha\,\mathbf{Q})\,\mathbf{v}$, and an innovative approach for this can be found in Al-Mohy and Higham (2011). This approach is well suited for computing the matrix exponential times several probing vectors in parallel.

Using rational approximations for the square-root or inverse square-root with random vectors ($\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$) with Krylov is another venue which has been pursued in Aune et al. (2012) and Simpson et al. (2008). In these articles, the authors demonstrate that in circumstances where the Cholesky factorisation is impossible to compute due to memory constraints, using rational approximations with Krylov methods is a good substitute, and also show that it is competitive in other circumstances.

In some cases, we may be interested in other entries of $f(\mathbf{Q})$ than its diagonal ones. For $f(t) = t^{-1}$, we obtain correlations between specific nodes and for $f = \exp$ we can obtain an estimate of the communicability between two nodes in an undirected graph. Looking at (25), we note that if we change $\mathbf{v}_j^T \odot f(\mathbf{Q})\,\mathbf{v}_j$ to $\boldsymbol{w}_j \odot f(\mathbf{Q})\,\mathbf{v}_j$, it may be possible to extract other entries of the matrix $f(\mathbf{Q})$. The question that remains is how to choose $\{\boldsymbol{w}_i\}_{j=1}^{kn}$ corresponding to the set $\{\boldsymbol{v}_j\}_{j=1}^{n}$. A heuristic that may help in forming the set of $\boldsymbol{w}_i$s is that if $\boldsymbol{v}_j$ is given, the corresponding $\boldsymbol{w}_i$s should be those corresponding to the neighbours of the nonzero entries of $\boldsymbol{v}_j$. We do not pursue this idea in here, but it is an interesting topic for future research.

5.2 Software

The software package KRYLSTAT by Aune, E. contains an implementation of the log-determinant approximation outlined in Section 2 with the extension in Section 3.4. For ease of use, MATLAB (MATLAB (2010)) wrappers for the relevant functions are included. It also contains implementations of one of the sampling procedures found in Aune et al. (2012) and a refined version of the marginal variance computations found in Tang and Saad (2010). The package can be found on `http://www.math.ntnu.no/~erlenda/KRYLSTAT/`.

References

Al-Mohy, A. and Higham, N. (2011). Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators. Technical report, University of Manchester.

Aune, E., Eidsvik, J., and Pokern, Y. (2012). Iterative Numerical Methods for Sampling from High Dimensional Gaussian Distributions. *Statistics and Computing*, Submitted.

Bai, Z. and Golub, G. (1997). Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices. *Annals of Numerical Mathematics*, 4:29–38.

Bekas, C., Kokiopoulou, E., and Saad, Y. (2007). An estimator for the diagonal of a matrix. *Applied numerical mathematics*, 57(11-12):1214–1229.

Benzi, M. and Golub, G. (1999). Bounds for the entries of matrix functions with applications to preconditioning. *BIT Numerical Mathematics*, 39(3):417–438.

Benzi, M. and Razouk, N. (2007). Decay bounds and $\mathcal{O}(n)$ algorithms for approximating functions of sparse matrices. *Electronic Transactions on Numerical Analysis*, 28:16–39.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36:192–236.

Bolin, D. and Lindgren, F. (2011). Spatial models generated by nested stochastic partial differential equations, with an application to global ozone mapping. *The Annals of Applied Statistics*, 5(1):523–550.

Buland, A. and Omre, H. (2003). Bayesian linearized AVO inversion. *Geophysics*, 68:185–198.

Candés, E., Demanet, L., Donoho, D., and Ying, L. (2006). Fast discrete curvelet transforms. *Multiscale Modeling Simulation*, 5(3):861– 899.

Chan, T., Tang, W., and Wan, W. (1997). Wavelet sparse approximate inverse preconditioners. *BIT Numerical Mathematics*, 37(3):644–660.

Chen, Y., Davis, T., Hager, W., and Rajamanickam, S. (2008). Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)*, 35(3):22.

Cressie, N. (1993). *Statistics for Spatial Data*. Wiley.

Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for large spatial datasets. *Journal of the Royal Statistical Society, Series B*, 70:209–226.

Culberson, J. (1992). Iterated greedy graph coloring and the difficulty landscape. Technical report, University of Alberta.

Davies, P. I. and Higham, N. J. (2005). *QCD and Numerical Analysis III*, chapter Computing f(A)b for Matrix Functions f, pages 15–24. Springer-Verlag.

Davis, T. and Hager, W. (1999). Modifying a sparse Cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 20(3):606–627.

Driscoll, A. (2009). The Schwarz Christoffel Toolbox available at `http://www.math.udel.edu/~driscoll/software/SC`. Electronic.

Eidsvik, J., Shaby, B. A., Reich, B. J., Wheeler, M., and Niemi, J. (2011). Estimation and prediction in spatial models with block composite likelihoods using parallel computing. Technical report, NTNU.

Estrada, E., Hatano, N., and Benzi, M. (2011). The physics of communicability in complex networks. *Arxiv preprint arXiv:1109.2950*.

Frommer, A. (2003). BiCGStab (l) for families of shifted linear systems. *Computing*, 70(2):87–109.

Fuentes, M. (2007). Approximate likelihood for large irregularly spaced spatial data. *Journal of the American Statistical Association*, 102(477):321–331.

Gröchenig, K. (2001). *Foundations of time-frequency analysis*. Birkhauser.

Hale, N., Higham, N. J., and Trefethen, L. N. (2008). Computing A^α, $\log(A)$ and related matrix functions by contour integrals. *SIAM Journal of Numerical Analysis*, 46:2505–2523.

Higham, N. J. (2008). *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Huckle, M. and Grote, M. (1997). Parallel preconditioning with sparse approximate inverses. *Siam J. Sci. Comput*, 18(3):838–853.

Hutchinson, M. (1989). A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Commun. Stat. Simula.*, 18:1059–1076.

Jegerlehner, B. (1996). Krylov space solvers for shifted linear systems. *arXiv:hep-lat/9612014v1*.

Karypis, G. and Kumar, V. (1999). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359.

Lindgren, F., Lindstrøm, J., and Rue, H. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: The SPDE approach. *Journal of the Royal Statistical Society, Series B*, 73:423–498.

Mallat, S. (1998). *a wavelet tour of signal processing*. Academic Press.

MATLAB (2010). *Version 7.11.0 (R2010b)*. The MathWorks Inc., Natick, Massachusetts.

McPeters, R., Aeronautics, U. S. N., Scientific, S. A., and Branch, T. I. (1996). *Nimbus-7 Total Ozone Mapping Spectrometer (TOMS) data products user's guide*. NASA, Scientific and Technical Information Branch.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields*. Chapman & Hall.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392.

Rue, H. and Tjelmeland, H. (2002). Fitting Gaussian Markov Random Fields to Gaussian Fields. *Scandinavian Journal of Statistics*, 29:31–49.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems, 2nd Ed.* SIAM.

Simpson, D. (2008). *Krylov subspace methods for approximating functions of symmetric positive definite matrices with applications to applied statistics and anomalous diffusion.* PhD thesis, School of Mathematical Sciences, Queensland Univ of Tech.

Simpson, D., Turner, I., and Pettitt, A. (2008). Fast sampling from a Gaussian Markov random field using Krylov subspace approaches. Technical report, School of Mathematical Sciences, Queensland Univ of Tech.

Tang, J. and Saad, Y. (2010). A Probing Method for Computing the Diagonal of the Matrix Inverse. Technical report, Minnesota Supercomputing Institute for Advanced Computational Research.

van der Vorst, H. and Melissen, J. (1990). A Petrov-Galerkin type method for solving Axk= b, where A is symmetric complex. *Magnetics, IEEE Transactions on*, 26(2):706–708.