Mathematical Geology, Vol. 35, No. 1, January 2003 (© 2003)

Efficient Computation of Linearized Cross-Covariance and Auto-Covariance Matrices of Interdependent Quantities[1]

Wolfgang Nowak,[2] Sascha Tenkleve,[2] and Olaf A. Cirpka[2]

In many geostatistical applications, spatially discretized unknowns are conditioned on observations that depend on the unknowns in a form that can be linearized. Conditioning takes several matrix–matrix multiplications to compute the cross-covariance matrix of the unknowns and the observations and the auto-covariance matrix of the observations. For large numbers n of discrete values of the unknown, the storage and computational costs for evaluating these matrices, proportional to n^2, become strictly inhibiting. In this paper, we summarize and extend a collection of highly efficient spectral methods to compute these matrices, based on circulant embedding and the fast Fourier transform (FFT). These methods are applicable whenever the unknowns are a stationary random variable discretized on a regular equispaced grid, imposing an exploitable structure onto the auto-covariance matrix of the unknowns. Computational costs are reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log_2 n)$ and storage requirements are reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

KEY WORDS: Toeplitz, circulant, embedding, spectral, FFT, convolution.

INTRODUCTION

In geostatistics, spatially distributed unknowns, such as soil and aquifer properties, the hydraulic head or solute concentrations, are commonly interpreted as realizations of random processes, characterized by their mean values and covariance functions. The same holds for time series analysis, e.g., in stochastic hydrology. Given observations of related quantities, information is inferred upon the unknowns by the process of conditioning, e.g., using iterative cokriging (e.g., Kitanidis, 1995) or through the generation of conditional realizations (e.g., Dietrich and Newsam, 1996). To infer information from the observations onto the unknowns, this class of methods utilizes the cross-covariance between the unknowns and the observations, and the auto-covariance among the observations.

[1] Received 7 January 2002; accepted 27 November 2002.

[2] Institut für Wasserbau, Universität Stuttgart, Pfaffenwaldring 61, 70569 Stuttgart, Germany; e-mail: wolfgang.nowak@iws.uni-stuttgart.de

0882-8121/03/0100-0053/1 © 2003 International Association for Mathematical Geology

This kind of conditioning requires several computationally expensive matrix–matrix multiplications. Consider n discrete values of the unknown arranged in the $n \times 1$ vector **s**, and m observations of a related quantity in the $m \times 1$ vector **t**. Vector **s** is characterized by its $n \times n$ covariance matrix $\mathbf{Q_{ss}}$. Given the relation $\mathbf{t} = \mathbf{f}(\mathbf{s})$ between the observations and the unknowns, a linearized sensitivity matrix $\mathbf{H} = \partial\mathbf{t}/\partial\mathbf{s}$ is computed, e.g., using the highly efficient adjoint-state method (Sun, 1994). According to linear error propagation and its quasi-linear extensions, the $n \times m$ cross-covariance matrix $\mathbf{Q_{st}}$ of the unknowns **s** and the observations **t**, and the auto-covariance matrix $\mathbf{Q_{tt}}$ of the observations **t** are given by (e.g., Schweppe, 1973, p. 368)

$$\mathbf{Q_{st}} = \mathbf{Q_{ss}}\mathbf{H}^T \tag{1}$$

$$\begin{aligned}\mathbf{Q_{tt}} &= \mathbf{H}\mathbf{Q_{ss}}\mathbf{H}^T + \mathbf{R} \\ &= \mathbf{H}\mathbf{Q_{st}} + \mathbf{R},\end{aligned} \tag{2}$$

with **R** denoting the auto-covariance matrix of the measurement errors, typically a diagonal matrix. There is a long list of other geostatistical methods that require similar products of $\mathbf{Q_{ss}}$ with matrices or vectors.

If a fine discretization of the unknown is required, the computational costs and storage requirements for these operations become restricting or even strictly inhibiting (Zimmerman and others, 1998). Typically, the resolution of the unknowns is dictated not by the geostatistical method, but rather by the applications that will process the values of the unknowns in further steps. Depending on the numerics involved in these applications, the resolution of the unknowns may easily rise up to the order of 10.000 to 1.000.000 discrete values.

The construction of $\mathbf{Q_{ss}}$ is a computation $\mathcal{O}(n^2)$, the matrix multiplication $\mathbf{Q_{st}} = \mathbf{Q_{ss}}\mathbf{H}^T$ is $\mathcal{O}(mn^2)$, and the computation of $\mathbf{Q_{tt}}$ via $\mathbf{Q_{tt}} = \mathbf{Q_{ts}}\mathbf{H}^T$ or $\mathbf{Q_{tt}} = \mathbf{H}\mathbf{Q_{ts}}\mathbf{H}^T$ is $\mathcal{O}(nm^2)$ or $\mathcal{O}(nm^2 + mn^2)$, respectively. Storage of $\mathbf{Q_{ss}}$ is $\mathcal{O}(n^2)$. To prevent memory overflow, single columns of $\mathbf{Q_{ss}}$ can be constructed, multiplied by **H** and deleted, a procedure that lets CPU time explode drastically.

These circumstances have heavily limited methods like cokriging that require explicit formulation of $\mathbf{Q_{sy}}$ and $\mathbf{Q_{yy}}$. In most cases, however, the unknown is a stationary random variable discretized on a regular and equispaced grid. This imposes symmetric Toeplitz (ST) or symmetric block Toeplitz with Toeplitz block (STT) structure onto $\mathbf{Q_{ss}}$ and reduces the storage requirements for $\mathbf{Q_{ss}}$ to $\mathcal{O}(n)$ (Zimmerman, 1989).

Toeplitz matrices can be embedded in larger circulant matrices (e.g., Dietrich and Newsam, 1993; Kozintsev, 1999; Nott and Wilson, 1997). Highly efficient algorithms for circulant matrices have been discovered as early as the middle of the twentieth century (Davis, 1979; Good, 1950) and applied and extended successfully ever since. The vast majority of these methods is based on the fast

Fourier transform (FFT) (Cooley and Tukey, 1965), computing the eigenvalues of circulant matrices in $\mathcal{O}(n \log_2 n)$ operations while only storing $\mathcal{O}(n)$ elements of the matrix. By now, a respectable toolbox of these so-called spectral methods is available for geostatistics, e.g., the generation of unconditional (Dietrich and Newsam, 1993) and conditional realizations (Dietrich and Newsam, 1996).

In this paper, we want to bring spectral methods for the efficient computation of cross-covariance and auto-covariance matrices and comparable matrix–matrix products to the attention of the geostatistical community. We summarize a method known in the field of signal processing as spectral convolution for the multiplication of circulant matrices with a vector (van Loan, 1992), and present an extension of this method for the evaluation of vector–matrix–vector products, and show how to use these methods for the matrix products discussed above.

Spectral convolution reduces the computational costs for computing $\mathbf{Q_{st}}$ from $\mathcal{O}(mn^2)$ to $\mathcal{O}(mn \log_2 n)$. The extension of this method to vector–matrix–vector products allows the computation of $\mathbf{Q_{tt}}$ in $\mathcal{O}(nm^2 + mn \log_2 n)$. Although the operation $\mathbf{Q_{tt}} = \mathbf{HQ_{st}}$ is $\mathcal{O}(nm^2)$ once $\mathbf{Q_{st}}$ has been computed, the direct evaluation of $\mathbf{Q_{tt}}$ without prior computation and storage of $\mathbf{Q_{st}}$ is useful in some cases, e.g., for the identification of structural parameters in Kitanidis (1995).

We have chosen to demonstrate the application to two-dimensional (2D) domains. The modifications for one and three dimensions are straightforward. One-dimensional (1D) problems lead to symmetric circulant (SC) matrices, while 2D problems lead to symmetric block circulant matrices with circulant blocks (SCC), and level 3 blocking structures will occur in three-dimensional (3D) applications.

This paper is organized as follows: In the next section, we provide some basic definitions of structured matrices. Then, we will explain when STT covariance matrices occur, and how to embed STT in SCC matrices. Finally, in the last section, we discuss the spectral methods for elementary vector–matrix and vector–matrix–vector products and how to divide the matrix–matrix products into these elementary operations.

TOEPLITZ AND CIRCULANT MATRICES

In the following, all quantities related to Toeplitz matrices are primed. An $n'_x \times n'_x$ ST matrix has the structure (Golub and van Loan, 1996, p. 193)

$$\mathbf{T} = \begin{bmatrix} t_0 & t_1 & \cdots & t_{n'_x-1} \\ t_1 & t_0 & & t_{n'_x-2} \\ \vdots & & \ddots & \vdots \\ t_{n'_x-1} & t_{n'_x-2} & \cdots & t_0 \end{bmatrix}. \tag{3}$$

The first row is given by the series $t_0 \,.\,.\, t_{n'_x-1}$. To construct the $(k+1)$th row, shift the kth row to the right by one, and fill the leading empty position with the kth element of the series $t_1 \,.\,.\, t_{n'_x-1}$. STT matrices have the same structure with the t_i replaced by $\mathbf{T}_i$, denoting ST blocks sized $n'_y \times n'_y$, so that the total size is $n'_y n'_x \times n'_y n'_x$. ST (STT) matrices are uniquely defined by their first (block)row.

In the following, unprimed quantities correspond to circulant matrices. SC matrices sized $2n_x \times 2n_x$ are defined as (Golub and van Loan, 1996, pp. 201–202)

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & \dots & c_{n_x} & \dots & c_1 \\ c_1 & c_0 & & c_{n_x-1} & & c_2 \\ \vdots & & \ddots & & & \vdots \\ c_{n_x} & c_{n_x-1} & & c_0 & & c_{n_x-1} \\ \vdots & & & & \ddots & \vdots \\ c_1 & c_2 & \dots & c_{n_x-1} & \dots & c_0 \end{bmatrix}. \tag{4}$$

The first row is given by the series $c_0 \,.\,.\, c_{n_x} \,.\,.\, c_1$ (the index runs from zero to n_x and down to one again). To construct the $(k+1)$th row, shift the kth row to the right by one, and fill the leading empty position with the last element of the kth row. In SCC matrices, the c_i are replaced by $\mathbf{C}_i$, which themselves are SC submatrices sized $2n_y \times 2n_y$, so the total size is $4n_y n_x \times 4n_y n_x$. SC matrices are completely defined by the first n_x entries of their first row, and SCC matrices are defined by the first n_y entries of the first n_x blocks each.

GENERAL PROCEDURE

The overall approach is as follows: (1) Define the unknown as a discretized random space variable such that its auto-covariance matrix $\mathbf{Q}_{\mathbf{ss}}$ has STT structure; (2) embed the STT matrix $\mathbf{Q}_{\mathbf{ss}}$ in a larger SCC matrix; then (3) compute all matrix–matrix products using spectral methods, and (4) extract the results from the embedded matrices. In the following, we discuss steps (1), (2), and (4) to give an overview over the entire procedure. Step (3) is discussed in a separate section.

Structure of Covariance Matrices

Consider a finite regular $n'_y \times n'_x$ grid with constant grid spacing d_x and d_y, sized $L'_x \times L'_y$, and $\mathbf{x}'$ and $\mathbf{y}'$ the $n'_y n'_x \times 1$ vectors with the x and y coordinates of its nodes. $\mathbf{s}'$ denotes a stationary Gaussian random space variable on the regular grid with zero mean and covariance function $R(h)$, in which h is the effective anisotropic separation distance.

The covariance matrix $\mathbf{Q}'_{ss}$ is given by $Q'_{kl} = Q'_{lk} = R\,(h_{kl})$, with h_{kl} being the corresponding separation between the points $(x_k,\ y_k)$ and $(x_l,\ y_l)$. If the global point numbering is ordered by y_k first and then by x_k, $\mathbf{Q}'_{ss}$ has STT structure with $n'_x \times n'_x$ blocks sized $n'_y \times n'_y$ (Zimmerman, 1989).

If the grid is the $n_y \times n_x$ unit cell of a periodic medium, both the unknown and the covariance function are periodic as well, and the resulting covariance matrix has SCC structure with $n_x \times n_x$ blocks sized $n_y \times n_y$.

Circulant Embedding

Before we deal with the mathematics of circulant embedding to convert STT to SCC matrices, we graphically describe the corresponding embedding of the random space variable. Consider that a finite domain Ω' may be interpreted as a subdomain of a larger virtual domain Ω. This process is referred to as embedding, Ω' is the embedded and Ω is the embedding domain. In order to maintain the statistical properties of Ω' in the embedding process, the mean value and the covariance function for all separation distances that are observable in Ω' must be identical for Ω.

For circulant embedding, the embedding has to be periodic, with periodic statistical properties, and Ω is the unit cell of the periodic embedding domain. For simplicity, we choose Ω twice the length and width of Ω' (Fig. 1). Since covariance functions are even functions by definition, we mirror the covariance function of Ω' to render it periodic (Fig. 2).

For our purposes, there is no need to actually generate the random space variable in the embedding domain, but only its SCC covariance matrix. For better understanding, it should be mentioned that even if the random variable was generated in the embedding domain, it would not be obtained by mirroring the finite domain, but by mirroring its covariance function. This automatically yields the variable periodic through perfect correlation each period length. A very detailed and graphic description of circulant embedding is given by Kozintsev, (1999).

The condensed mathematical description of this embedding on the matrix level is as follows: to embed ST in SC matrices, extend the series $t_0 \ldots t_{n'_x-1}$ by appending the elements $t_1 \ldots t_{n'_x-2}$ in reverse order to obtain a series $c_0 \ldots c_{n_x} \ldots c_1$, $n_x = n'_x - 1$, corresponding to mirroring the covariance function to render it periodic. To embed STT in SCC matrices, embed the ST blocks $\mathbf{T}_i$ in SC blocks $\mathbf{C}_i$, and then extend the series of the blocks to obtain a periodic series of blocks.

Embeddings of other size are allowed under certain circumstances that are beyond the scope of this paper. Larger embeddings extend the series with new elements from the covariance function before appending the reversed section. Smaller embeddings can be chosen if, e.g., the last $2k$ elements of the series are

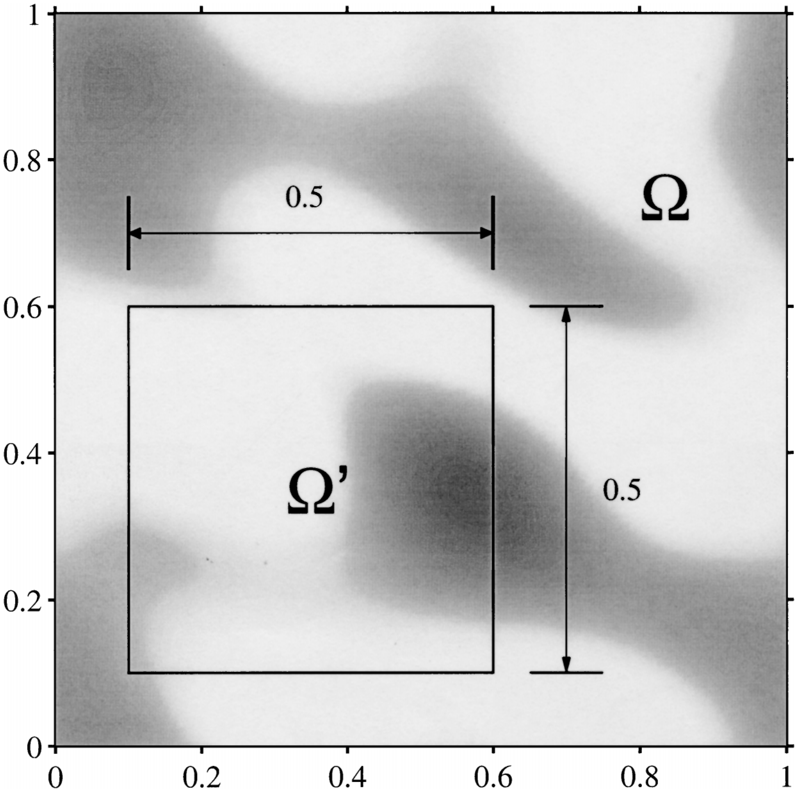

Figure 1. Finite domain Ω' embedded in the double-sized unit cell Ω of a periodic domain. Ω is not obtained by mirroring Ω', and thus is not symmetric.

constant, so that the last k elements serve as the mirrored image of the previous k elements.

As for the size of the embedding, two aspects are of special interest. First, for some other spectral methods, the resulting SCC matrix has to be nonnegative definite (Dietrich and Newsam, 1997; Nott and Wilson, 1997) or positive definite (Kozintsev, 1999). Second, choosing powers of two for n_y and n_x is especially suited for standard FFT algorithms, whereas newer FFT algorithms like the FFTW perform almost as well for n_y and n_x differing from powers of two (Frigo and Johnson, 1998).

To maintain the compatibility of matrix dimensions for multiplication, the Jacobian **H** has to undergo the same embedding by zero-padding all entries corresponding to the new entries in the SCC covariance matrix $\mathbf{Q}_{\text{ss}}$. Reshape the m rows sized $1 \times n'_y n'_x$ in to $n'_y \times n'_x$ matrices, pad them with zeros to obtain $n_y \times n_x$ matrices, and reshape them back to $1 \times n_y n_x$ rows to obtain the embedded version

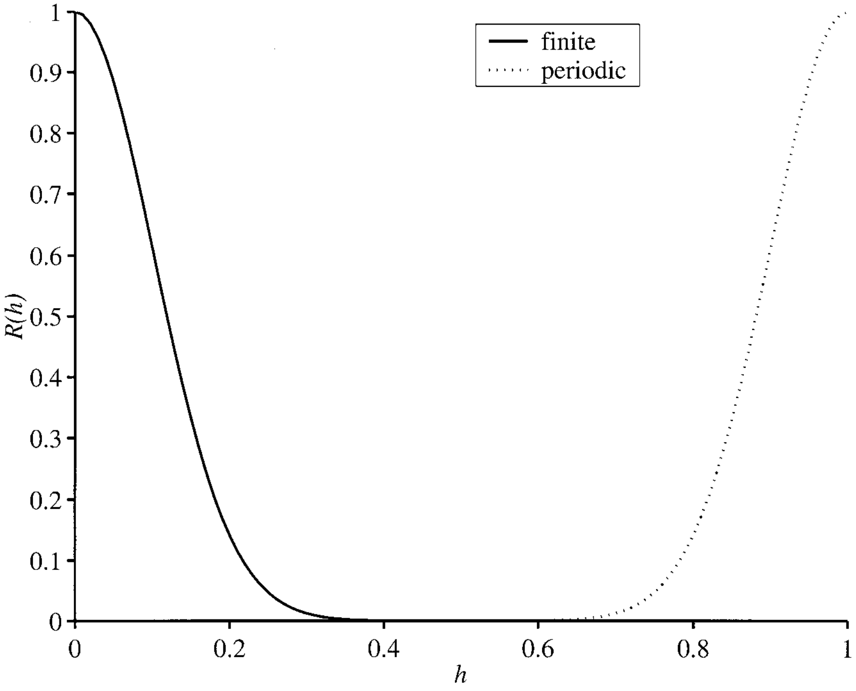

Figure 2. Covariance function $R(h)$ for finite domain (solid) and for periodic domain (dotted), example for 1D.

of **H** sized $m \times n_y n_x$. The zero padding is to suppress the influence of the new elements of $\mathbf{Q}_{\text{ss}}$ during matrix multiplication.

Extraction

The spectral result of $\mathbf{Q}_{\text{st}} = \mathbf{Q}_{\text{ss}}\mathbf{H}^T$ (sized $m \times n_y n_x$) corresponds to the embedded version of **H**. To extract the original-size $\mathbf{Q}_{\text{st}}$, reverse the process. $\mathbf{Q}_{\text{tt}}$ requires no extraction as it is sized $m \times m$.

MATRIX–MATRIX MULTIPLICATION

As a prerequisite of all spectral methods for circulant matrices, we recapitulate the diagonalization of SCC matrices first. Then, as matrix–matrix multiplications are a series of vector–matrix multiplications, we will discuss vector–matrix multiplication first and then return to matrix–matrix multiplications.

Eigenvalues

The diagonalization theorem (Dietrich and Newsam, 1993; Nott and Wilson, 1997) gives the eigenvalues of the $n_y n_x \times n_y n_x$ SCC matrix $\mathbf{Q}_{\mathrm{ss}}$:

$$\mathbf{Q}_{\mathrm{ss}} = \mathbf{F}^H \boldsymbol{\Lambda} \mathbf{F}, \tag{5}$$

with $\boldsymbol{\Lambda}$ denoting the diagonal matrix of eigenvalues. $\mathbf{F}$ is the 2D Fourier-matrix and $\mathbf{F}^H = \mathbf{F}^{-1}$ its Hermitian transpose (see Appendix). After multiplication by $\mathbf{F}$

$$\mathbf{F}\mathbf{Q}_{\mathrm{ss}} = \boldsymbol{\Lambda}\mathbf{F}. \tag{6}$$

One column of $\mathbf{Q}_{\mathrm{ss}}$ contains all information, thus consider the first column only:

$$\mathbf{F}\mathbf{Q}_{\mathrm{ss},1} = \boldsymbol{\Lambda}\mathbf{F}_1. \tag{7}$$

All entries of $\mathbf{F}_1$ equal $(n_x n_y)^{-\frac{1}{2}}$:

$$\boldsymbol{\lambda} = \sqrt{n_x n_y}\, \mathbf{F}\mathbf{Q}_{\mathrm{ss},1}, \tag{8}$$

where $\boldsymbol{\lambda}$ is an $n_y n_x \times 1$ vector of the eigenvalues, and $\mathbf{F}$ contains the eigenvectors (Barnett, 1990, pp. 350–354 for the 1D case). $\mathbf{F}\mathbf{Q}_{\mathrm{ss},1}$ is computed using 2D FFT (see Appendix). This renders the eigenvalue decomposition $\mathcal{O}(n \log_2 n)$ compared to the conventional $\mathcal{O}(n^3)$, with $n = n_y n_x$. The eigenvalues are computed once and then stored for all subsequent steps.

Matrix–Vector Multiplication

The product of $\mathbf{Q}_{\mathrm{ss}}$ and an $n_y n_x \times 1$ vector $\mathbf{u}$ can be simplified using Eq. (5):

$$\mathbf{Q}_{\mathrm{ss}}\mathbf{u} = (\mathbf{F}^H \boldsymbol{\Lambda} \mathbf{F})\mathbf{u} = \mathbf{F}^H \boldsymbol{\Lambda}(\mathbf{F}\mathbf{u}), \tag{9}$$

in which $\mathbf{F}\mathbf{u} = \mathbf{v}$ is executed via 2 D FFT. As $\boldsymbol{\Lambda}$ is diagonal, we can write

$$\mathbf{Q}_{\mathrm{ss}}\mathbf{u} = \mathbf{F}^H(\boldsymbol{\Lambda}\mathbf{v}) = \mathbf{F}^H[\lambda_1 v_1, \lambda_2 v_2, \ldots, \lambda_{n_x n_y} v_{n_x n_y}]^T, \tag{10}$$

using the inverse 2D FFT to compute $\mathbf{F}^H[\cdot]$. This procedure is known as spectral convolution (van Loan, 1992, pp. 205–209). For $\mathbf{u}^T\mathbf{Q}_{\mathrm{ss}}$, compute $(\mathbf{Q}_{\mathrm{ss}}\mathbf{u})^T$ as $\mathbf{Q}_{\mathrm{ss}} = \mathbf{Q}_{\mathrm{ss}}^T$.

Vector–Matrix–Vector Multiplication

For evaluating $\mathbf{u}_1^T \mathbf{Q}_{\mathbf{ss}} \mathbf{u}_2$ with $\mathbf{u}_1$ and $\mathbf{u}_2$ sized $n_y n_x \times 1$, the procedure is similar:

$$\begin{aligned}\mathbf{u}_1^T \mathbf{Q}_{\mathbf{ss}} \mathbf{u}_2 &= \mathbf{u}_1^T (\mathbf{F}^H \boldsymbol{\Lambda} \mathbf{F}) \mathbf{u}_2 = \mathbf{v}_1^H \boldsymbol{\Lambda} \mathbf{v}_2 \\ &= \sum_{k=1}^{n_x n_y} (v_1^*)_k \lambda_k (v_2)_k \end{aligned} \tag{11}$$

$\mathbf{v}_1 = \mathbf{F}\mathbf{u}_1$ and $\mathbf{v}_2 = \mathbf{F}\mathbf{u}_2$ are computed by 2D FFT and $\mathbf{v}_1^*$ is the complex conjugate of $\mathbf{v}_1$. For $\mathbf{u}_1 = \mathbf{u}_2$, this simplifies to the quadratic form (compare Nott and Wilson, 1997)

$$\mathbf{u}^T \mathbf{Q}_{\mathbf{ss}} \mathbf{u} = \sum_{k=1}^{n_x n_y} \lambda_k \left| v_k \right|^2 . \tag{12}$$

Matrix–Matrix Multiplications

Consider $\mathbf{H}$ an $m \times n_y n_x$ matrix. The computation of $\mathbf{Q}_{\mathbf{st}} = \mathbf{Q}_{\mathbf{ss}} \mathbf{H}^T$ can be split up into single vector–matrix multiplications:

$$\mathbf{Q}_{\mathbf{st},k} = \mathbf{Q}_{\mathbf{ss}} \mathbf{u}_k, \; k = 1..\, m, \tag{13}$$

where $\mathbf{u}_k$ is the transpose of the kth row of $\mathbf{H}$. Likewise, $\mathbf{Q}_{\mathbf{tt}} = \mathbf{H}\mathbf{Q}_{\mathbf{ss}}\mathbf{H}^T$ can be split up into m^2 subproblems:

$$\mathbf{Q}_{\mathbf{tt},kl} = \mathbf{u}_k^T \mathbf{Q}_{\mathbf{ss}} \mathbf{u}_l . \tag{14}$$

As $\mathbf{H}\mathbf{Q}_{\mathbf{ss}}\mathbf{H}^T$ is symmetric, only the upper triangle and the diagonal has to be computed.

DISCUSSION

To demonstrate the power of spectral methods for computing $\mathbf{Q}_{\mathbf{ss}}\mathbf{H}^T$ compared to standard methods, both methods were implemented and timed in MATLAB. As the standard method runs out of memory quite easily, an additional method was coded that exploits the STT structure of $\mathbf{Q}'_{\mathbf{ss}}$ to reduce storage. It subsequently generates the columns of $\mathbf{Q}_{\mathbf{ss}}$ by permutation, storing only one column at a time. The MATLAB codes were executed on a personal computer (1.5 GB RAM, 900 MHz AMD Athlon CPU) for problems with different numbers

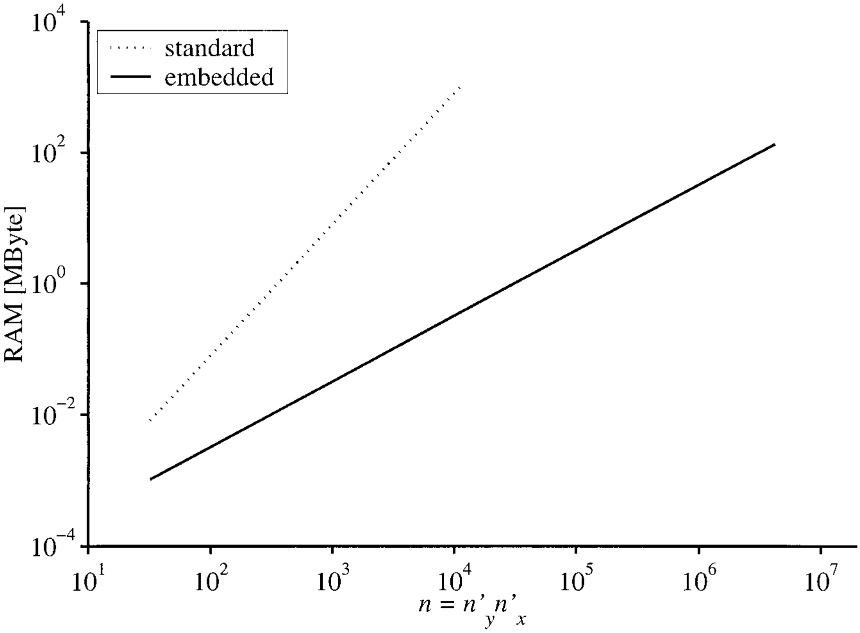

Figure 3. Log–Log plot for the comparison of memory consumption in MByte for the storage of $\mathbf{Q}_{ss}$ as a function of the number n of unknowns, $n = n_y n_x$, using the different methods.

of points $n = n'_y n'_x$. As exemplary problem, we chose a Gaussian covariance function with 20 times the correlation per domain length both in x and in y direction, so that $n_x = 1.3n'_x$ and $n_y = 1.3n'_y$ could be chosen as adequate embedding. The $m \times n_x n_y$ matrix $\mathbf{H}$ was generated randomly.

Figure 3 compares the storage requirements of the standard and the spectral method. On our reference computer, the standard method ran out of memory at $n = 2^{12} = 4.096$ and the columnwise standard method at $n = 2^{16} = 65.536$, while the FFT-based method could be applied to grids of up to $n = 2^{21} = 2.097.152$ without memory problems.

Figure 4 shows the comparison of CPU time for the matrix–matrix multiplication, including the generation of $\mathbf{Q}_{ss}$. Although the columnwise standard method has significant overhead for the permutation of the columns in $\mathbf{Q}_{ss}$, it is faster than the standard method by one order of magnitude as it only computes n entries of $\mathbf{Q}_{ss}$. Because of its embedding overhead, the FFT-based method is slower than both standard methods for small n, but outruns them for $n > 2^8 = 256$ as it is of lower order in n. At the upper limits, the columnwise standard method takes approximately 30 min for its maximum allowable problem size ($n = 2^{16}$ grid points) compared to 7 s for the FFT-based method, which only takes 8 min for its maximum resolution of $n = 2^{21}$.

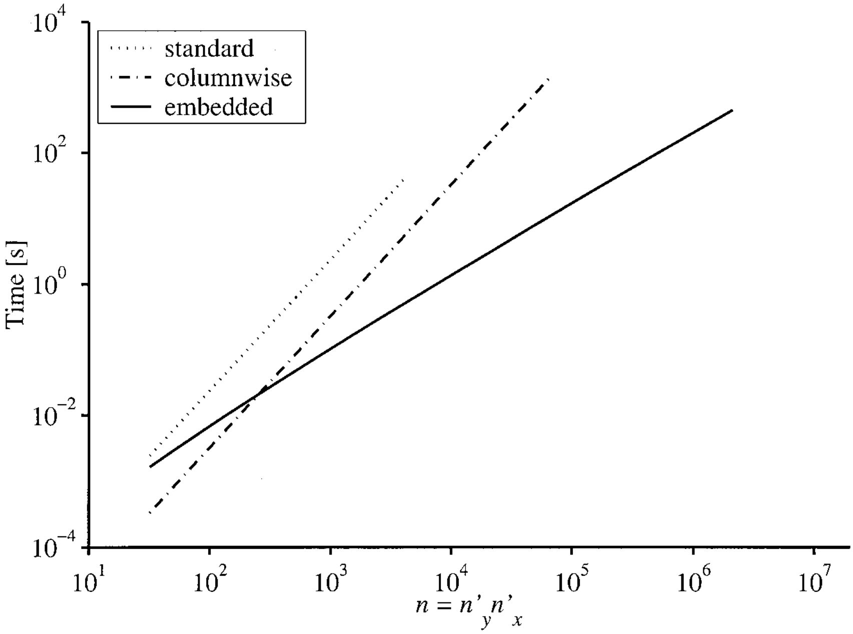

Figure 4. Log–Log plot for the comparison of CPU time in seconds required for the computation of $\mathbf{Q}_{ss}\mathbf{H}^T$ as a function of the number n of unknowns, $n = n_y n_x$, using the different methods on a reference computer (900 MHz AMD Athlon CPU).

Comparison to Related Spectral Methods

The spectral matrix multiplication of STT matrices is exact, and the embedding matrix does not have to be definite. Both properties are in contrast to many other spectral applications. Inverting SC and SCC matrices through spectral computation of its eigenvalues requires the embedding matrix to be positive definite (Nott and Wilson, 1997), and the decomposition $\mathbf{Q}_{ss} = \mathbf{S}^T\mathbf{S}$ via the square root of eigenvalues requires nonnegative definiteness (Dietrich and Newsam 1993, 1996). Further, the eigenvalues of ST(T) and SC(C) matrices only asymptotically approach each other (Gray, 1972), so that inversion and determinant from spectral methods are only asymptotically exact, affecting the method by Nott and Wilson, (1997).

CONCLUSIONS

We have presented an existing spectral method and an extension thereof to multiply symmetric Toeplitz matrices or symmetric block-Toeplitz matrices with

Toeplitz blocks with arbitrary matrices from the left-hand side, the right-hand side or both sides simultaneously, motivated by the high computational effort of standard methods for cross-covariance matrices $\mathbf{Q}_{\mathbf{st}}$ and auto-covariance matrices $\mathbf{Q}_{\mathbf{tt}}$ of observations **t** and unknowns **s**.

The spectral methods perform significantly better than standard methods: Storage requirements are reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, and computational costs are reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log_2 n)$. On the reference computer used in this work, the maximum allowable matrix size was $n = 2^{21}$ for the spectral method instead of $n = 2^{16}$ for an enhanced standard method, and the spectral method was faster by orders of magnitude for large n.

Up to now, computational costs and storage requirements have severely restricted geostatistical methods that require explicit computation of $\mathbf{Q}_{\mathbf{st}}$ and $\mathbf{Q}_{\mathbf{tt}}$, such as linear and iterative cokriging and the generation of conditional realizations. The spectral methods discussed in this work overcome these restrictions.

ACKNOWLEDGMENTS

This work was funded by the Emmy-Noether-program of the Deutsche Forschungsgemeinschaft (DFG) under grant Ci26/3-2. We thank Jaime Gomez-Hernandez for his helpful remarks improving the paper.

REFERENCES

Barnett, S., 1990, Matrices methods and applications (Oxford applied mathematics and computing science series): Clarendon Press, Oxford, 450 p.

Cooley, J., and Tukey, J., 1965, An algorithm for the machine calculation of complex fourier series: Math. Comp., v. 19, p. 297–301.

Davis, P., 1979, Circulant matrices, pure and applied mathematics: Wiley, New York, 250 p.

Dietrich, C., and Newsam, G., 1993, A fast and exact method for multidimensional gaussian stochastic simulations: Water Resourc. Res., v. 29, no. 8, p. 2861–2869.

Dietrich, C., and Newsam, G., 1996, A fast and exact method for multidimensional gaussian stochastic simulations: Extension to realizations conditioned on direct and indirect measurements: Water Resourc. Res., v. 32, no. 6, p. 1643–1652.

Dietrich, C., and Newsam, G., 1997, Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix: SIAM J. Sci. Comp., v. 18, no. 4, p. 1088–1107.

Frigo, M., and Johnson, S., 1998, FFTW: An adaptive software architecture for the FFT, In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, Seattle, WA, v. 3, pp. 1381–1384. http://www.fftw.org

Golub, G., and van Loan, C., 1996, Matrix computations, 3rd edn.: John Hopkins University Press, Baltimore, 694 p.

Good, I., 1950, On the inversion of circulant matrices: Biometrika, v. 37, p. 185–186.

Gray, R., 1972, On the asymptotic eigenvalue distribution of Toeplitz matrices: IEEE Trans. Inf. Theor., v. 18, no. 6, p. 725–730.

Kitanidis, P., 1995, Quasi-linear geostatistical theory for inversing: Water Resourc. Res., v. 31, no. 10, p. 2411–2419.

Kozintsev, B., 1999, Computations with gaussian random fields: PhD thesis, Institute for Systems Research, University of Maryland.

Nott, D., and Wilson, R., 1997, Parameter estimation for excursion set texture models: Signal Process., v. 63, p. 199–210.

Schweppe, F., 1973, Uncertain dynamic systems: Prentice-Hall, Englewood Cliffs, NJ, 563 p.

Sun, N.-Z., 1994, Inverse problems in groundwater modeling, Theory and applications of transport in porous media: Kluwer Academic, Dordrecht, The Netherlands.

van Loan, C., 1992, Computational frameworks for the Fast Fourier Transform: SIAM, Philadelphia, 273 p.

Zimmerman, D., 1989, Computationally exploitable structure of covariance matrices and generalized covariance matrices in spatial models: J. Stat. Comp. Sim., v. 32, no. 1/2, p. 1–15.

Zimmerman, D., de Marsily, G., Gotaway, C., and Marietta, M., 1998, A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow: Water Resourc. Res., v. 34, no. 6, p. 1373–1413.

APPENDIX

Definition 1 (2D DFT). The discrete 2D Fourier transform (DFT) of a rectangular $n_y \times n_x$ matrix **U** is the $n_y \times n_x$ matrix **V** with the entries:

$$V_{kl} = \frac{1}{\sqrt{n_x n_y}} \sum_{p=0}^{n_y-1} \sum_{q=0}^{n_x-1} U_{pq} \cdot \exp\left(-\frac{2\pi i pk}{n_y} - \frac{2\pi i ql}{n_x}\right) \quad k = 0..n_y - 1, \quad l = 0..n_x - 1. \tag{A1}$$

The inverse Fourier transformation is given by the same expression with opposite signs inside the exponential function.

Definition 2 (2D Fourier matrix). The 2D Fourier transformation can be formulated in matrix notation. Consider **F** an $n_y n_x \times n_y n_x$ matrix with the entries:

$$F_{kl} = \frac{1}{\sqrt{n_x n_y}} \exp\left(-\frac{2\pi i}{n_y}\left[\frac{l - (l \bmod n_x)}{n_x}\right] \times \left[\frac{k - (k \bmod n_x)}{n_x}\right]\right) \cdot \exp\left(-\frac{2\pi i}{n_x}(l \bmod n_x)(k \bmod n_x)\right) \tag{A2}$$
$$l, m = 0,\ 1, \ldots,\ n_x n_y - 1,$$

then the 2D Fourier transform **V** of an $n_y \times n_x$ matrix **U** is:

$$\mathbf{v} = \mathbf{F}\mathbf{u}, \tag{A3}$$

in which $\mathbf{u}$ and $\mathbf{v}$ are $n_y n_y \times 1$ vectors obtained from rearranging the matrices $\mathbf{U}$ and $\mathbf{V}$ column-wise. The inverse Fourier transformation corresponds to the inverse of the Fourier matrix. As $\mathbf{F}$ is unitary,

$$\mathbf{F}^{-1} = \mathbf{F}^{H} \tag{A4}$$

$$\mathbf{F}^{H}\mathbf{F} = \mathbf{I}\,, \tag{A5}$$

where the superscript H denotes the Hermitian transpose, i.e., the transpose of the complex conjugate.

The Fourier matrix and its inverse are for the derivations only. In the practical application, the 2D FFT algorithm is applied. $\mathbf{Fu}$ is evaluated by reshaping $\mathbf{u}$ into an $n_y \times n_x$ matrix $\mathbf{U}$, computing the 2D FFT $\mathbf{V} = \mathcal{F}(\mathbf{U})$, and reshaping $\mathbf{V}$ back to an $n_y n_x \times 1$ vector $\mathbf{v} = \mathbf{Fu}$, which is $\mathcal{O}(n \log_2 n)$ instead of $\mathcal{O}(n^2)$, $n = n_y n_x$.