High Performance Multivariate Spatial Modeling for Geostatistical Data on Manycore Systems

Mary Lai O. Salvaña, Sameh Abdulah, Huang Huang, Hatem Ltaief, Ying Sun, Marc G. Genton, and David E. Keyes

Abstract—Modeling and inferring spatial relationships and predicting missing values of environmental data are some of the main tasks of geospatial statisticians. These routine tasks are accomplished using multivariate geospatial models and the cokriging technique. The latter requires the evaluation of the expensive Gaussian log-likelihood function, which has impeded the adoption of multivariate geospatial models for large multivariate spatial datasets. However, this large-scale cokriging challenge provides a fertile ground for supercomputing implementations for the geospatial statistics community as it is paramount to scale computational capability to match the growth in environmental data coming from the widespread use of different data collection technologies. In this paper, we develop and deploy large-scale multivariate spatial modeling and inference on parallel hardware architectures. To tackle the increasing complexity in matrix operations and the massive concurrency in parallel systems, we leverage low-rank matrix approximation techniques with task-based programming models and schedule the asynchronous computational tasks using a dynamic runtime system. The proposed framework provides both the dense and the approximated computations of the Gaussian log-likelihood function. It demonstrates accuracy robustness and performance scalability on a variety of computer systems. Using both synthetic and real datasets, the low-rank matrix approximation shows better performance compared to exact computation, while preserving the application requirements in both parameter estimation and prediction accuracy. We also propose a novel algorithm to assess the prediction accuracy after the online parameter estimation. The algorithm quantifies prediction performance and provides a benchmark for measuring the efficiency and accuracy of several approximation techniques in multivariate spatial modeling.

Index Terms—Gaussian log-likelihood, geospatial statistics, high-performance computing, large multivariate spatial data, low-rank approximation, multivariate modeling/prediction.

1 INTRODUCTION

The convergence of high-performance computing (HPC) and big data brings great promise in accelerating and improving large-scale applications [1], [2] on climate and weather modeling [3], astronomy [4], transportation [5], and bioinformatics [6]. Climate and weather modeling, in particular, is one of the first applications of HPC for big data [7]. The need to improve climate and weather models has pushed for advances in environmental data collection technologies such as spaceborne, airborne, and ground sensors [8]. The volume of data coming from these sources is huge and increasing. For instance, NASA’s Earth Observing System Data and Information System (EOSDIS) is expected to archive more than 37 petabytes of data in 2020 [9]. By 2022, the yearly increase is projected at 47.7 petabytes.

Environmental datasets, such as climate and weather variables, are often recorded from different spatial locations, and thus indexed by \( s \in \mathbb{R}^d, d \geq 1 \), where \( s \) is the location of the measurement. Usually there are multiple variables measured at each location, such as temperature, humidity, wind speed, and atmospheric pressure. These colocated variables may or may not depend on each other and on the variables at other locations.

A main concern when dealing with environmental datasets is missing data on one or few variables. For instance, when using environmental variables as inputs to climate and weather models, the gaps in areas with no measurements caused by poor atmospheric conditions or defective sensors, to name a few, need to be filled [10], [11]. Multivariate geospatial statistics can interpolate environmental variables at unsampled locations by modeling the multivariate spatial dependencies. While every variable of interest can be modeled and predicted separately, it has been shown that more accurate predictions can be produced when modeling dependent variables jointly [12], [13].

Modeling the variables as realizations from a multivariate Gaussian random field is the cornerstone of multivariate geostatistics. Multivariate random fields are the equivalent of multivariate random variables, where a vector of variables is measured at each location [14]. Mathematically, this means that at location \( s \in \mathbb{R}^d, d \geq 1 \), each variable is considered as one component of the \( p \)-dimensional vector \( Z(s) \), i.e., \( Z(s) = \{Z_1(s), \ldots, Z_p(s)\}^\top \), where \( \top \) is the transpose operator, \( p \) is the number of variables, and \( Z_i(s) \in \mathbb{R} \) indicates the value of the \( i \)th variable at location \( s \), \( i = 1, \ldots, p \). When \( Z(s) \) is Gaussian, it is completely characterized by its mean vector and multivariate covariance function, which is more commonly known as cross-covariance function in geostatistical statistics and multivariate kernel in computer science.

Cokriging is the prediction of multiple variables using an optimal predictor [12], [13]. To perform cokriging on
a multivariate Gaussian random field, one only needs to specify a mean vector and a cross-covariance function. In this work, we assume that the multivariate Gaussian random field has mean zero and focus our attention on the cross-covariance function, which introduces considerable computational challenges. This zero-mean assumption is reasonable since regression can be used to remove the mean. Furthermore, we restrict our attention to a specific cross-covariance function that is stationary and isotropic. These assumptions on the cross-covariance function are building blocks to more sophisticated ones including nonstationary and anisotropic cross-covariance functions, which can be readily accommodated by our proposed framework. They do not present significant limitations on the large-scale multivariate geospatial analysis aimed at this work.

In practice, a class of cross-covariance functions is first selected and its unknown parameters are estimated from the data [12], [15], [16]. Estimation relies on the maximum likelihood estimation (MLE) approach [17], [18]. Suppose \( \theta \in \mathbb{R}^q \) collects all the \( q \) true unknown parameters of the cross-covariance function. The MLE of \( \theta \), denoted by \( \hat{\theta} \in \mathbb{R}^q \), is the \( q \)-dimensional vector which maximizes the log-likelihood function

\[
\ell(\theta) = \frac{-np}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma(\theta)| - \frac{1}{2} Z^\top \Sigma(\theta)^{-1} Z,
\]

with respect to all \( q \) parameters in \( \theta \). Here \( Z \in \mathbb{R}^{pn} \) collects all the \( p \)-dimensional vectors \( Z(s_i) \), at \( n \) locations, \( s_1, s_2, \ldots, s_n \), i.e., \( Z = \{Z(s_1)^\top, Z(s_2)^\top, \ldots, Z(s_n)^\top\}^\top \). \( \Sigma(\theta) \) is the \( pn \times pn \) cross-covariance matrix for \( Z \) and \( |\Sigma(\theta)| \) denotes the determinant of \( \Sigma(\theta) \). The entries of \( \Sigma(\theta) \) are calculated from the cross-covariance function that is known up to \( \hat{\theta} \). The procedure in constructing this cross-covariance matrix is discussed in more details in Section IV.

The MLE involves the computation of the log-likelihood function in Equation (1) for each iteration in the optimization. In large-scale multivariate problems with irregularly spaced locations, the log-likelihood requires \( \mathcal{O}(p^2n^2) \) memory and \( \mathcal{O}(p^4n^3) \) operations per iteration, due to the Cholesky decomposition of the cross-covariance matrix \( \Sigma(\theta) \). The prohibitive computational cost and corresponding storage of computing the log-likelihood function can be reduced by relying on low-rank approximation techniques that exploit the low-rank representations of the cross-covariance matrix, for instance, the Tile Low-Rank (TLR) approximation [19], [20], [21]. TLR is the preferred approximation approach in case of parallel execution and it involves dividing the matrix into a set of tiles, then applying low-rank approximation separately to each tile.

Scalable large-scale geospatial statistical modeling have been attempted mostly in the univariate (\( p = 1 \)) setting [21], [22], [23]. When \( p > 1 \), the computation becomes much more challenging. We introduce a framework to deal with large-scale multivariate geospatial statistics problems that provides both the dense and the TLR-based approximate versions of MLE operations on very large problem sizes. The framework depends on the asynchronous task-based dense linear algebra library CHAMELEON [24] and the Hierarchical Computations linear algebra library HiCMA [25] for support in dense and low-rank matrix computations, respectively. Both libraries rely on the dynamic runtime system STARPU [26] to exploit the computing power on shared and distributed-memory systems based on multi-core, many-core, and hybrid hardware architectures.

Predictions can be made at unsampled locations by kriging. The quality of these predictions can be assessed using the mean square prediction error (MSPE) [21] and the mean square relative prediction error (MSRPE) [27]. However, it has been shown that these commonly used criteria cannot adequately assess the prediction efficiency when different approximation methods are involved and are up for comparisons [28]. As an alternative, under univariate modeling, the authors in [28] proposed two criteria, namely, the mean loss of efficiency (MLOE) and the mean misspecification of the mean square error (MMOM), to more appropriately assess the loss of prediction efficiency when an approximated version of the model was used instead of the exact one. In this work, we first present a parallel implementation based on our software stack of the univariate MLOE/MMOM criteria, and then we propose a modified algorithm which extends these criteria to assess the multivariate approximations modeling on large-scale multivariate spatial datasets.

The remainder of the paper is as follows. Section II summarizes the contribution of this work. Section III contains a brief discussion on multivariate geospatial statistics, an introduction to the log-likelihood estimation problem, and a review to some approximation techniques which ameliorate the complexities encountered in MLE operations. Section IV establishes the research contributions of this paper. Section V provides detailed illustrations of our proposed modeling framework on synthetic and real datasets. Section VI delivers an overall summary and conclusion.

## 2 Contributions

The six-fold contributions of the paper are as follows:

- We present multivariate geospatial modeling/inference in large-scale systems on both exact and TLR-based approximation computations with reduced complexity on the log-likelihood for both estimation and spatial prediction.
- We propose a novel multivariate assessment algorithm based on existing univariate criteria to evaluate our TLR-based parameter estimation accuracy.
- We implement a parallel version of the univariate and the new multivariate criteria to assess the prediction efficiency on synthetic and real datasets.
- We port the proposed implementation on shared-memory, GPUs, and distributed-memory systems using a modular software stack.
- We evaluate the performance of both the parallel exact and TLR-based MLE computations, as well as the proposed multivariate assessment criteria, using different parallel platforms such as Intel Xeon Skylake/Cascade Lake, AMD EPYC (Rome), ARM ThunderX2, NVIDIA V100 GPU, and a distributed-memory Cray XC40 system.
- We conduct a set of experiments designed to assess the accuracy of our implementation in terms of inference and prediction on both synthetic and real datasets.
3 Overview of the Problem
This section describes multivariate spatial modeling and inference based on a given cross-covariance function, with a brief background on low-rank approximation techniques that have been used in the literature to reduce the complexity of likelihood function estimation in large-scale problems.

3.1 Cross-Covariance Function
Quantifying spatial dependence of multiple variables is one of the main foci of multivariate geospatial statistical modeling. It is a matrix-valued function of dimension $p \times p$, parametrized by $\theta \in \mathbb{R}^d$, that describes the degree of dependence between values at two locations $s_1$ and $s_2$, and is of the form $C(\|h\|; \theta) = \{C_{ij}(\|h\|; \theta)\}_{i,j=1}^{p}$ under the isotropy assumption, where $h = s_1 - s_2 \in \mathbb{R}^d$, $\| \cdot \|$ denotes the Euclidean norm, and $C_{ij}(\|h\|; \theta) = \text{cov}(Z_i(s), Z_j(s + h))$. When $i = j$, $C_{ii}(\|h\|; \theta)$ is called the marginal covariance function and it measures the spatial dependence between the $i$-th variable at the two locations $s$ and $s + h$. When $i \neq j$, $C_{ij}(\|h\|; \theta)$ is called the cross-covariance function and it measures the spatial dependence between the $i$-th variable at spatial location $s$ and the $j$-th variable at spatial location $s + h$, for $i, j = 1, \ldots, p$. The choice of cross-covariance function $C_{ij}(\|h\|; \theta)$ is data and application driven. However, $C_{ij}(\|h\|; \theta)$ needs to ensure that the $\Sigma(\theta)$ it builds is a positive definite matrix for any $n \in \mathbb{N}$ and any finite set of points $s_1, \ldots, s_n$.

3.2 Matérn Cross-Covariance Function
The geospatial statistical Gaussian modeling landscape is replete with cross-covariance function models. A comprehensive review of the available cross-covariance functions can be found in [12]. The most popular and flexible is the Matérn cross-covariance function which was introduced in [29] and has the form

$$C_{ij}(\|h\|; \theta) = \frac{\sigma^2_{ij}}{2^{\nu_{ij} - 1} \Gamma(\nu_{ij})} \frac{\|h\|^{\nu_{ij}}}{a_{ij}^{\nu_{ij}}} K_{\nu_{ij}}(\|h\|/a_{ij}),$$

for $i, j = 1, \ldots, p$, where $K_{\nu}(\cdot)$ is the modified Bessel function of the second kind of order $\nu$ and $\Gamma(\cdot)$ is the gamma function. Here $\theta$ includes, for $i = 1, \ldots, p$, the marginal variance ($\sigma^2_{ii} > 0$), smoothness ($\alpha_{ii} > 0$), and spatial range ($\alpha_{ii} > 0$) parameters, and for $i, j = 1, \ldots, p$, $i \neq j$, the collocated covariance ($\sigma^2_{ij}$), cross smoothness ($\nu_{ij} > 0$), and cross spatial range ($a_{ij} > 0$) parameters. There are different classes of the multivariate Matérn depending on the added restrictions on the parameter space; see [29], [30].

3.3 The Parsimonious Multivariate Matérn
The parsimonious multivariate Matérn, proposed in [29] refers to the multivariate Matérn model with

$$a_{ij} = a, \quad \nu_{ij} = \frac{1}{2}(\nu_{ii} + \nu_{jj}), \quad \sigma^2_{ij} = \rho_{ij}\sigma_{ii}\sigma_{jj}$$

where

$$\rho_{ij} = \beta_{ij} \frac{\Gamma(\nu_{ii} + \frac{d}{2})^{1/2} \Gamma(\nu_{jj} + \frac{d}{2})^{1/2}}{\Gamma(\nu_{ij} + d/2)^{1/2} \Gamma\left(\frac{1}{2}(\nu_{ii} + \nu_{jj})\right)^{1/2} \Gamma\left(\frac{1}{2}(\nu_{ii} + \nu_{jj}) + \frac{d}{2}\right)^{1/2}},$$

for any $\sigma^2_{ii}, a, \nu_{ii} > 0$, $i = 1, \ldots, p$, $d \geq 1$. Here $(\beta_{ij})_{i,j=1}^{p}$ with $\beta_{ii} = 1$ for $i = 1, \ldots, p$, is a symmetric and positive definite correlation matrix. Under this class, the components of $Z(s)$ are assumed to have the same spatial range parameter, $a$. The colocated correlation coefficient parameter, $\rho_{ij}$, describes the dependence or correlation between the $i$-th and $j$-th components situated at the same location, through the latent parameter $\beta_{ij}$. When $\beta_{ij} = 0$, $Z_i(s)$ and $Z_j(s)$ are independent. Otherwise, when $\beta_{ij} > 0$ ($\beta_{ij} < 0$), the two are positively (negatively) dependent. Sample realizations from the parsimonious bivariate Matérn model above are shown in Fig. 3 where $\theta = (\sigma^2_{i1}, \sigma^2_{22}, a, \nu_{11}, \nu_{22}, \beta)^\top = (1, 1, 0.2, 0.5, 1, 0.5)^\top$.

3.4 Multivariate Prediction
The cross-covariance matrix $\Sigma(\theta)$ is crucial in obtaining predictions of unknown variables at a prediction location and measuring the uncertainty of these predictions. A prediction location may have all or some variables that are missing. The first case happens when there are locations with sensors that collect measurements for atmospheric variables like temperature, precipitation, and wind speed, for example, and one might be interested in predicting the values of these variables at locations with no sensors. The second case occurs when measurements of one variable are difficult or expensive to obtain while measurements of another variable, correlated with the first one, are easy to collect. In this scenario, there will be more locations with data collection instruments for the cheaper variable, while observations will be sparse for the expensive one. The first case is more prevalent in environmental applications wherein sensors measuring different variables simultaneously were deployed at predetermined sites. Hence, in this work, we assume that all prediction locations are missing the measurements for all $p$ variables. Multivariate geospatial prediction proceeds as follows. Suppose $s_0 \in \mathbb{R}^d$ is a prediction location with an unknown vector of $p$ variables $Z(s_0)$. Under the squared-error loss criterion, the best linear unbiased predictor of $Z(s_0)$, given $Z = \{Z(s_1)^\top, \ldots, Z(s_n)^\top\}$, also called the cokriging predictor, is

$$\hat{Z}(s_0) = c_0^\top \Sigma(\theta)^{-1} Z.$$ 

Here $c_0$ is the $pn \times p$ matrix formed by taking the cross-covariance between $Z(s_0)$ and $Z(s_r)$, at all sampled locations $s_r$, $r = 1, \ldots, n$, i.e.,

$$c_0 = \{C(s_0 - s_1; \theta), \ldots, C(s_0 - s_n; \theta)\}^\top.$$ 

3.5 Low-Rank Approximation
Geospatial statistical Gaussian modeling relies heavily on the operations done on $\Sigma(\theta)$. In the early stages of modeling, $\Sigma(\theta)$ has to be formed by evaluating the cross-covariance function at $n$ locations, for all $p$ variables. Parameters then have to be estimated, with the cross-covariance function evaluated every time new sets of parameters are assumed. Further, the Gaussian log-likelihood in Equation (1) requires the inverse and the determinant of $\Sigma(\theta)$. Prediction at unsampled locations also involves the inverse of $\Sigma(\theta)$.

Several techniques to bypass these computing obstacles have been proposed, all of which exploit data sparsity.
Low-rank approximations have been widely used for data modeling. Several low-rank representations of the original Gaussian processes had been proposed during recent years, including predictive process [31], where a select set of knots is used to approximate the original process and a low-rank model is obtained. Later, the modeling approaches were extended to multi-resolution approximation [32] to capture spatial structures from different scales. A low-rank approximation can also be applied to Vecchia’s representation for the composite likelihood [33], resulting in reduced computation burden in obtaining the composite likelihood. Moreover, the low-rank approximation can also be applied to cross-covariance matrices with limited loss of information. Another possibility to facilitate computations for exascale processing parallelism using three backend runtime systems, and MPI) and runs on different hardware architectures (e.g., CPU/GPU, shared/distributed-memory). STARPU may decide at runtime to execute the task on different hardwares, driven by performance models.

### 3.6 Univariate MLOE/MMOM

A common metric to assess the quality of the predictions is the mean squared prediction error (MSPE). The MSPE is computed as $\text{MSPE} = \frac{1}{n_{\text{pred}}} \sum_{l=1}^{n_{\text{pred}}} \|Z(\theta_{0,l}) - Z(\hat{\theta})\|^2$, where $s_{0,1}, s_{0,2}, \ldots, s_{0,n_{\text{pred}}}$ are the $n_{\text{pred}}$ prediction locations. When the predictions are obtained through an approximated cross-covariance matrix, e.g., the TLR version of the exact covariance matrix, an appropriate metric should be used. The authors in [25] suggested the use of the mean loss of efficiency (MLOE) and the mean misspecification of the mean square error (MMOM). However, the formulations of these metrics are only available for univariate predictions ($p = 1$). The formulas and the algorithm for the univariate MLOE/MMOM can be found in [28]. In Section 4.4 we extend these metrics and the algorithm to multivariate.

### 4 Proposed Framework

In this section, we explain in detail our software dependencies, proposed multivariate modeling and inference implementation, and multivariate MLOE/MMOM criteria which are used to assess the accuracy of the multivariate modeling.

#### 4.1 Parallel Software Architecture

Our proposed framework internally relies on a list of software dependencies including CHAMELEON [24], STARPU [26], HiCMA [25], STARS-H [35], and NLOPT [36], as demonstrated in Fig. 1. CHAMELEON is a tile-based high-performance numerical library based on task-parallel programming models which offer a more structured way of expressing parallelism using three backend runtime systems, namely, QUARK [37], PARSEC [38], and STARPU. In this study, we use the STARPU dynamic runtime system because of its ability to support a wide range of parallel heterogeneous hardware architectures from different vendors like Intel, AMD, NVIDIA, and ARM. STARPU executes defined generic task graphs, generated by a built-in sequential task flow (STF) programming model. The sequential task-based program can then be represented as a Directed Acyclic Graph (DAG), where nodes correspond to tasks and edges to data dependencies, as shown in Fig. 1 with a $4 \times 4$ Cholesky factorization DAG example. The task execution order depends on a set of data dependencies (e.g., read, write, and read-write) defined over the application data. STARPU scheduler pushes the set of tasks to the available processing unit based on these dependencies which may lead to asynchronous execution. STARPU supports different programming languages (e.g., Pthreads, CUDA, OpenCL, and MPI) and runs on different hardware architectures (e.g., CPU/GPU, shared/distributed-memory). STARPU may decide at runtime to execute the task on different hardwares, driven by performance models.

#### 4.2 Exact Multivariate Modeling

In multivariate modeling, there are two ways to build $\Sigma(\theta)$ [12]. The first approach (Representation I) is to build an $n \times n$ matrix with block elements of $p \times p$ matrices $C(s_l - s_r; \theta)$, $l, r = 1, \ldots, n$. The second approach (Representation II) is to build a $p \times p$ matrix with block elements of $n \times n$ matrices $\{C_{ij}(s_l - s_r; \theta)\}_{i, j = 1}^{n}$, $i, j = 1, \ldots, p$. To illustrate, suppose $p = 2$ and $n = 3$. Fig. 2 shows the form of the $6 \times 6$ cross-covariance matrix $\Sigma(\theta)$ under the two representations.

The two representations in Fig. 2 yield a symmetric positive definite matrix. Cholesky factorization, the backbone
of MLE, is performed on this symmetric positive definite matrix to obtain its inverse and log determinant required for maximizing the log-likelihood function in Equation (1).

A simulation study on the parsimonious bivariate Matérn was conducted to assess the efficiency in parameter estimation and accuracy of predictions under the two representations via comparison of the medians and standard deviations of the estimated parameters and the MSPE of the predictions. The results indicate that the two representations are numerically equivalent in exact computation and either one can substitute for the other.

To hasten parameter estimation of the exact parsimonious Matérn cross-covariance function, we maximize the profile log-likelihood in lieu of the full log-likelihood in Equation (1). The profile likelihood is a variant of the full profile likelihood for maximizing the log-likelihood function in Equation (1). The example was drawn from a synthetic set of parameters \( \theta \), which represents a moderate spatial dependence between the two variables, \( Z_1 \) and \( Z_2 \). Other sets of parameters representing varying strengths of spatial dependence were also examined and the results do not differ significantly from what is shown in Fig. 4.

Fig. 3: An example of TLR approximation tile: \( T_{2,1} \) with dimension \( nb \times nb \) is approximated by two matrices \( U_{2,1} \) and \( V_{2,1} \) with dimension \( nb \times k_{2,1} \).

The effectiveness of the TLR mechanism depends on the ranks of the matrix tiles, which, in turn, depend on the accuracy requirements of the given application. To reduce the ranks of the cross-covariance matrix tiles, which suffers from the existence of numerous synchronization points that slow down the overall performance.

Fig. 4: Rank distributions of a 7200×7200 bivariate covariance matrix using \( nb = 720 \) with parsimonious bivariate Matérn parameters \( \theta = (1, 1, 0.09, 0.5, 1, 0.5)^\top \) under (a) TLR5, (b) TLR7, and (c) TLR9.

4.3 TLR-Based Multivariate Modeling

Through the last decade, tile algorithms were created to adapt to parallel architectures that require data sharing [40]. The tiling mechanism improved block-based algorithm which suffers from the existence of numerous synchronization points that slow down the overall performance.

Tile Low-Rank (TLR) approximation algorithms have been implemented based on the tiling technique. Instead of applying the low-rank approximation to the whole matrix, each tile is compressed as a separate unit. Here, we rely on the TLR implementation of [41], where the authors have implemented the TLR approximation by performing the singular value decomposition (SVD) algorithm for each off-diagonal tile by preserving the most significant values and vectors in the corresponding tile, i.e., the tile rank. The diagonal tiles are kept dense since they cannot be approximated. Ranks of the off-diagonal tiles are determined based on the accuracy requirement of the application. Fig. 5 shows an example of compressing an off-diagonal tile \( T_{2,1} \) to two matrices \( U_{2,1}, V_{2,1} \).
Fig. 5: Memory footprint of exact and TLR-based MLE with varying size of \( n \) and two measurement vectors \( Z_1 \) and \( Z_2 \).

dense representation for \( 10^{-5}, 10^{-7}, \) and \( 10^{-9} \), respectively. The memory saving increases with larger problem sizes, as seen in the figure.

The performance model of TLR is driven by the most time-consuming kernel, i.e., the TLR matrix-matrix multiplication (TLR-MM). The arithmetic complexity of a single TLR-MM is \( 36 \times nb \times k^2 \) [43], with \( nb \) the tile size and \( k \) the tile rank that depends on the number of significant singular values after compression. The total number of operations is then \( O(n^2k) \) attained when \( nb = O(\sqrt{n}) \). This tile size is a trade-off between the arithmetic intensity of the kernel and the degree of parallelism of the algorithm. The performance model of TLR is driven by a quadratic regime, which contrasts with the cubic regime for exact computations. The detailed complexity analysis of TLR Cholesky factorization can be found in [44]. We are also looking into more advanced matrix compression strategies [45], [46], [47] that exhibit better arithmetic complexity but may be challenging to implement on massively parallel systems due to their hierarchical structures.

### 4.4 Proposed Multivariate MLOE/MMOM Criteria

Assessing the estimation accuracy of the modeling approach is challenging and requires a well-developed algorithm. Our novel multivariate prediction assessment implementation depends on the MLOE/MMOM criteria in [28]. There are two possible multivariate versions of the MLOE/MMOM. The first version is a naive extension done by computing the respective MLOE/MMOM of each variable and averaging these values. This approach requires univariate covariance models and the univariate version of the cokriging equation (5). Another version, which we propose, is to utilize cross-covariance functions and the cokriging equation (5). Denote the cokriging error vector \( e(s_0) = (e_1, \ldots, e_p)^\top \), where \( e_i = Z_i(s_0) - Z_i(s_0) \) is the predictor for variable \( i, i = 1, \ldots, p \), at a prediction location \( s_0 \), obtained using the cokriging equation (5), with the true cross-covariance function parameters, \( \theta \). The mean square error of this predictor is

\[
E_t = \text{tr}\{ C(0; \theta) - c_0^\top (\Sigma(0; \theta))^{-1} c_0^0 \},
\]

where the subscript \( t \) in \( c_0^t \) specifies that the parameters used in Equation (5) are the true parameters \( \theta \) and \( \text{tr} \) indicates the trace of the matrix. Suppose now that the set of estimated parameters derived from using a certain approximation of the covariance matrix, \( \hat{\theta} \), were used to build the cokriging equation (5). The error introduced by the approximation is \( e_a(s_0) = (e_{a,1}, \ldots, e_{a,p})^\top \), where \( e_{a,i} = Z_i^a(s_0) - Z_i(s_0) \) and \( Z_i^a(s_0) \) is the predictor for variable \( i \) at a prediction location \( s_0 \), obtained using the cokriging equation (5) with \( \hat{\theta} \). The mean square error of this predictor is

\[
E_a = \text{tr}\{ C(0; \theta) - c_0^a^\top (\Sigma(\hat{\theta}))^{-1} c_0^a \} + c_0^a^\top (\Sigma(\hat{\theta}))^{-1} \Sigma(\theta) \Sigma(\hat{\theta})^{-1} c_0^a \}
\]

(8)

where \( c_0^a \) is Equation (6) evaluated using \( \hat{\theta} \) and \( \Sigma(\hat{\theta}) \) is the parametric \( pn \times pn \) cross-covariance matrix also evaluated using \( \hat{\theta} \). The subscript \( a, \theta \) specifies that given the true parameters \( \theta \), the estimated parameters \( \hat{\theta} \) from the approximated model are used instead.

The multivariate MLOE/MMOM, denoted as MLOE\(^{CK}\) and MLOE\(^{MMOM}\), respectively, are as follows:

\[
\text{MLOE}^{CK} = \frac{1}{n_{\text{pred}}} \sum_{l=1}^{n_{\text{pred}}} \text{LOE}^{CK}(s_0, l)
\]

(9)

\[
\text{MLOE}^{MMOM} = \frac{1}{n_{\text{pred}}} \sum_{l=1}^{n_{\text{pred}}} \text{MOM}^{MMOM}(s_0, l),
\]

(10)

where \( \text{LOE}^{CK}(s_0) = E_{t,a} - 1 \) and \( \text{MOM}^{MMOM}(s_0) = \frac{E_a}{E_t} - 1 \). The superscript \( CK \) stipulates that the multivariate extensions were derived from the cokriging equation (5) and \( E_a \) is Equation (7) evaluated using \( \hat{\theta} \) and \( c_0^a \).

The algorithm implementing this approach for \( p = 2 \) is outlined in Algorithm 1. This new algorithm is similar to [28] except now the matrix-valued Matérn cross-covariance function is utilized instead of the scalar-valued univariate Matérn covariance function, e.g., BiMatérn\((s_1, s_2; \theta)\) returns a \( 2 \times 2 \) matrix.

**Algorithm 1:** Algorithm for Parallel Univariate/Bivariate MLOE/MMOM.

**Input**: \( n_{\text{pred}} \): number of prediction locations; \( n \): number of sampled locations; \( s_{0,1}, \ldots, s_{0,n_{\text{pred}}} \): prediction locations; \( s_{1,1}, \ldots, s_{1,n} \): sampled locations; dist\((a, b)\): function that computes the distance between \( a \) and \( b \); \( \theta \): true parameters for the bivariate Matérn; \( \hat{\theta} \): estimated parameters for the approximated bivariate Matérn; BiMatérn\((a, b; \theta)\): bivariate Matérn covariance function evaluated at locations \( a \) and \( b \); Equation \( 1 \) CovMat\((\theta)\): \( 2n \times 2n \) cross-covariance matrix.

**Result**: MLOE := mean of LOE\(^{CK}\), MMOM := mean of MOM\(^{MMOM}\).

1. \( \Sigma = \text{CovMat}(\theta); \)
2. \( \Sigma^a = \text{CovMat}(\hat{\theta}); \)
3. \( LL^\top = \Sigma; \) \{cholesky factorization\}
4. \( L(L^\top)^\top = \Sigma; \) \{cholesky factorization\}
5. \( \text{for } l = 1 \text{ to } n_{\text{pred}} \text{ do} \)
6. \( \text{if } r = 1 \text{ to } n \text{ do} \)
7. \( c_{0}^a((2r-1) : (2r), 1 : 2) = \text{BiMatérn}(s_{0,1}, s_{0,l}; \hat{\theta});\) \{Equation (2)\}
8. \( c_{0}^a((2r-1) : (2r), 1 : 2) = \text{BiMatérn}(s_{0,a}, s_{0,l}; \hat{\theta});\) \{Equation (4)\}
9. \( \text{end for} \)
10. \( \text{tmp1} = \text{tr}\{ \text{BiMatérn}(s_{0,1}, s_{0,l}; \theta) - 2c_{0}^a(L(L^\top)^\top L)^\top c_{0}^a + c_{0}^a(L(L^\top)^\top L)^\top (L(L^\top)^\top L)^\top c_{0}^a \}; \) \{Equation (3)\}
11. \( \text{tmp2} = \text{tr}\{ \text{BiMatérn}(s_{0,a}, s_{0,l}; \theta) - c_{0}^a(L(L^\top)^\top L)^\top c_{0}^a \}; \) \{Equation (4)\}
12. \( \text{tmp3} = \text{tr}\{ \text{BiMatérn}(s_{0,a}, s_{0,l}; \theta) - c_{0}^a(L(L^\top)^\top L)^\top c_{0}^a \}; \) \{Equation (4)\}
13. \( \text{LOE}^{CK}(l) = \text{tmp1} \times \text{tmp2} - 1; \) \{Equation (7)\}
14. \( \text{MOM}^{MMOM}(l) = \text{tmp3} / \text{tmp1} - 1; \) \{Equation (10)\}
15. \( \text{end for} \)
Assuming $n_{\text{pred}} \ll n$, the memory footprint and the arithmetic complexity of Algorithm 1 for any values of $p$ depends solely on the Cholesky factorizations of $\Sigma$ and $\Sigma^a$ (line 3 and line 4). Each Cholesky factorization requires $p^2 n^2$ memory footprint and $(1/3)p^3n^3$ number of operations. Indeed, the code section containing the nested loops (from line 5 to line 15) carries only Level-1 and Level-2 BLAS operations involving several dot products and triangular solves. Since these matrix operations account for the lower order terms [48], the overall memory footprint is then $2p^2n^2$ and the arithmetic complexity is $(2/3)p^3n^3$.

5 Performance Results

In this section, we assess the performance and the accuracy of the TLR approximation to $\Sigma(\theta)$ on large-scale simulations and real datasets. The performance assessment involves a wide range of parallel hardware systems while the accuracy assessment entails simulating synthetic datasets from the parsimonious Matérn function, estimating the six parameters of this model, and predicting values at screened locations. Estimation and prediction accuracy of the TLR approximation are also evaluated on real datasets. The designed experiments show that the TLR approximation outperforms the exact bivariate computation while maintaining the accuracy required by geospatial statistics applications.

5.1 Testbed and Methodology

All experiments described in this paper were performed using a variety of shared-memory systems: Intel Xeon Skylake, Intel Xeon Cascade Lake, AMD EPYC (Rome), ARM ThunderX2 Cavium. For the distributed memory experiments, we rely on a 6,174 nodes Cray XC40 system. The AD artifact includes more details about the hardware and the list of our software dependencies on different systems including compiling options.

To obtain timing results, we run each simulation three times on every single hardware with the same configuration and report the average. We find runtime variations between 0.1% and 0.5% on shared-memory systems and between 1% and 3% on distributed-memory systems. The latter is slightly higher since the runs are subject to network fluctuations depending on the current load of the system.

5.1.1 Synthetic Datasets

We perform large-scale simulations from the parsimonious Matérn cross-covariance function. Fig. 6 shows a bivariate random field simulated from Equation (2) at $n = 24,964$ locations using our synthetic data generator with the following configuration:

- $\beta = 0.5$, i.e., $Z_1$ and $Z_2$ are positively correlated. This parameter controls how correlated $Z_1$ and $Z_2$ at any location. The effect of this parameter is visually detectable since wherever there are red (blue) spots in $Z_1$, red (blue) spots in $Z_2$ tend also to be seen.
- $\nu_1 = 0.5$ and $\nu_2 = 1$, i.e., $Z_2$ is smoother than $Z_1$. The smoothness parameters show through observing that the values of $Z_2$ changes more slowly than the values of $Z_1$ from one pixel or location to another.
- $a = 0.2$. This parameter affects $Z_1$ and $Z_2$ in different ways. For $Z_1$, this value of the scale parameter suggests that the marginal covariance of $Z_1$ drops to 0.05 when the locations are 0.6 units apart. For $Z_2$, it takes 0.8 units separation for its marginal covariance to drop to 0.05. Visually, this parameter dictates the sizes of the red and blue spots. The larger the $a$ becomes, the bigger the sizes of the spots are.

Fig. 6: Spatial images of the simulated bivariate realizations from the parsimonious Matérn cross-covariance model on a 158 × 158 regular grid on a unit square. Here $\sigma_{11}^2 = \sigma_{22}^2 = 1$, $\alpha = 0.2$, $\nu_{11} = 0.5$, $\nu_{22} = 1$, and $\beta = 0.5$.

To validate the accuracy of the TLR approximation, we simulate 50 different bivariate Gaussian random fields of the same configuration as the example in Fig. 6 and perform three kinds of experiments:

- Experiment 1: We show the merits of bivariate spatial modeling in the exact computations by varying the degree of colocated dependence between $Z_1$ and $Z_2$, controlled by $\rho$ through $\beta$, and examining whether there is a gain in prediction when using the parsimonious bivariate Matérn for different values of $\beta$, while fixing the other parameters as follows. $\sigma_{11}^2 = \sigma_{22}^2 = 1$, $\nu_{11} = 0.5$, $\nu_{22} = 1$, and $a = 0.09$.
- Experiment 2: We examine the quality of parameter estimates for the exact and TLR under different accuracies (TLR5, TLR7, and TLR9), using the available data in $n = 22,464$ locations (chosen randomly) and reserving the remaining $n_{\text{pred}} = 2,500$ as prediction locations. Furthermore, we contrast these with the estimation results to another approximation technique aforementioned, the Diagonal Super Tile (DST). Fixing $\nu_{11} = 0.5$ and $\nu_{22} = 1$, we vary the value of the remaining parameter responsible for spatial dependence over long distances, i.e., the range parameter, $a$. Different values of $a$ were chosen to represent weak ($a = 0.03$), moderate ($a = 0.09$), and strong ($a = 0.20$) spatial dependencies.
- Experiment 3: Predictions are made at the 2,500 prediction locations and the errors produced by the exact and the approximation models are assessed using the newly proposed multivariate MLOE/MMOM.

Section 5.4 presents the results of these experiments.

5.1.2 Real Datasets

We fit the parsimonious bivariate Matérn covariance function on a wind dataset consisting of two variables: zonal wind component, $U$ (variable 1), and meridional wind
component, $V$ (variable 2), both in $\text{m/s}$. The wind dataset was gathered from a Weather Forecasting and Research (WRF) model simulation as in [49] on the $[43^\circ E, 65^\circ E] \times [5^\circ S, 24^\circ N]$ region of the earth. Both variables come with a horizontal spatial resolution of 5 km.

We restricted the wind dataset to the Arabian Sea. This leaves us with $n = 116,100$ locations and a covariance matrix of dimension $232 \times 232$. The two locations which are located farthest from each other have a great circle distance of 2,681 km. The choice of this particular subregion was motivated by the need to ensure that the measurements exhibit spatial isotropy, i.e., the cross-covariance depends only on the distance between any two locations and not on the locations themselves. Often, this isotropy assumption holds when the locations are situated in areas with similar characteristics. As the locations are all on the ocean, this behavior can be expected. In order to obtain a zero-mean bivariate random field, we remove a spatially varying mean using the longitudes and latitudes as covariates. The resulting values after mean removal are approximately Gaussian and are plotted in Fig. 7.

Similar to the analysis we had for Fig. 6, we can make several observations for Fig. 7:

- $U$ and $V$ appear to be positively correlated as high and low measurements of both variables occur at almost the same locations.
- The changes in the measurements as you move from one location to the next is not abrupt, for both $U$ and $V$, unlike $Z_1$ in Fig. 6. The smoothness behavior of the two variables more closely mirror that of $Z_2$ in Fig. 6. Hence, the estimated values of $\nu_1$ and $\nu_2$ is closer to 1 than 0.5.
- The range parameter $a$ is not easily observable from Fig. 7 even from our sample random field in Fig. 6. However, the sizes of the red and blue spots in Fig. 7 very much resemble the sizes of the red and blue spots in Fig. 6. Thus, $a$ is likely to be close to 0.2.

5.2 TLR-Based Bivariate MLE Performance

In this section, we evaluate the TLR-based multivariate MLE performance compared to the exact MLE. All the experiments show one iteration of the MLE optimization since all the iterations have the same complexity for both exact and TLR-based computation. Fig. 8 shows the TLR performance on different shared-hardware architectures. The execution time is shown in the Y-axis while the number of spatial locations is shown in the X-axis. We use TLR5 as the benchmark of the speedup gained by the TLR computation.

With an Intel 56-core Skylake system, TLR5 can achieve on average 4X speedup compared to exact MLE, while on an Intel 40-core Cascade Lake system, the average speedup can reach 4.3X. With a 128-core AMD EPYC (Rome) system and a 64-core ThunderX2 ARM system, the average speedup reaches to 6X and 5.5X, respectively. All figures reveal more gains from the TLR-based approximation with larger problem sizes. Moreover, with a larger number of cores, the average speedup factor achieved increases.

On the Cray XC40 distributed-memory system, TLR achieves lower speedup compared to the exact but still outperforms it with different problem size and number of nodes. Tuning the tile size ($nb$) is challenging on distributed systems and it seems that our baseline runtime systems, i.e., STARPU, impacts performance with a large number of nodes. Fig. 9 shows the performance of different TLR accuracy levels gains from the TLR-based approximation with larger problem size up to 325K on 64 nodes. The average speedup gained is about 2X. With 128 nodes, the average speedup gained is about 1.8X as shown by Fig. 9.

Fig. 10 shows the strong scalability results using single node 40-core Intel Cascade Lake system with different number of cores and Cray XC40 machine using different number of nodes (up to 128 nodes). In Fig. 10a the Cascade Lake system shows decent parallel speedup as we increase the numbers of threads with $n = 63,001$. The parallel efficiency on average is around $72\%$ (i.e., $T_{min}/(N* T_N) \times 100\%$, where one thread execution time is $T_1$, and $N$ threads execution time is $T_N$), compared to single thread executions and across different computation variants, i.e., exact, TLR5, TLR7, and TLR9. In Fig. 10b, the exact computation achieves around 66.7X speedup, while the TLR approximation with different accuracy levels obtains around 51.7X speedup on average with different number of nodes using $n = 168,100$. The parallel efficiency of the TLR approximation varies between 60% and 43%. This is lower than the efficiency of the exact computation but expected due to the memory-bound versus compute-bound regime of executions opposing TLR and exact computations, respectively.

5.3 Univariate/Bivariate MLOE-MMOM Criteria Performance

In this set of experiments, we aim at assessing the performance of the proposed multivariate MLOE-MMOM criteria algorithm. We choose the bivariate as an example of a multivariate case with synthetic datasets generated by our framework. The experiments were performed on different shared-memory hardware architectures. We provide the time breakdown of the assessment operation of Algorithm 1 and split them into three parts: matrices generation time ($GEN\_TIME$) in lines 1-2, factorization time ($FACT\_TIME$) in lines 3-4, and computation time ($COMP\_TIME$) in lines 5-15. Figs. 11 and 12 show the time for each operation on both parallel univariate and bivariate MLOE-MMOM implementations.

As shown in both figures, the $COMP\_TIME$ is the most time-consuming part of Algorithm 1 as it requires an iterative execution loop equal to the number of the given missing locations, i.e., 100 in this set of experiments. However, with a larger matrix size, the $FACT\_TIME$ takes...
more time for the whole operation. One striking observation from Fig. 11d is that using a system with V100 GPU speeds up the $\text{FACT\_TIME}$ compared to $\text{COMP\_TIME}$. This is because the computation part involves several matrix-vector operations (i.e., Level-2 BLAS) that cannot exploit the computational power of the enclosed GPU.

### 5.4 TLR-Based Bivariate MLE Accuracy Assessment

Here, we assess the accuracy of proposed TLR-based bivariate MLE compared to the exact computation using both synthetic and real datasets.

#### 5.4.1 Synthetic Datasets

Fig. 13 summarizes the results of Experiment 1. The figure shows that the higher the value of the parameter responsible for the colocated dependence ($\beta$), the lower the prediction error becomes. In bivariate datasets, the inclusion of a second variable effectively increases the number of samples available for any one of the two variables when the colocated dependence between them is high (positive or negative). The more correlated $Z_1$ and $Z_2$ are, through $\beta$, the more information we get about $Z_1$ from $Z_2$, and vice versa. This echoes the theoretical results in [13], where it was shown that the colocated correlation parameter dictates the improvement introduced by cokriging (multivariate prediction) over kriging (univariate prediction). Similar conclusions were derived using other values of $a$, the parameter controlling the long range spatial dependence. This suggests that bivariate or multivariate ($p > 2$) modeling should be pursued regardless because while the gain in using a bivariate model is not so pronounced when the colocated dependence is not so high in the positive or negative direction, there is significant error reduction when the variables turn out to be highly correlated.

TABLE 1: A summary of the estimated parameters of the models and their corresponding prediction errors (MSPE). Note that the spatial range parameter $a$ is shared by the two variables in the bivariate model. The best model is that which reports the lowest MSPE.

| Model   | $\sigma^2$ | $\hat{a}$ | $\hat{\nu}$ | MSPE     |
|---------|------------|------------|--------------|----------|
| IDW     | -          | -          | -            | 0.072334 |
| U       | 0.484      | 0.134      | 2.533        | 0.000189 |
| Exact Bivariate | 0.718 | 0.161 | 2.283 | 0.000189 |
| V       | -          | -          | -            | 0.070375 |
| IDW     | -          | -          | -            | 0.070375 |
| Univariate | 1.014 | 0.198 | 2.153 | 0.000269 |
| Exact Bivariate | 0.710 | 0.161 | 2.033 | 0.000261 |

Fig. 14 plots the accuracy of our estimation procedure under the different TLR and exact implementations at different strengths of spatial dependence controlled by the range parameter, $a$. It also includes the results of the parameter estimation under the two different sizes of the DST. DST 40/60 means that 40% of the tiles from the diagonal are kept and the remaining 60% are annihilated. Similarly, DST 70/30 denotes that 70% of the tiles from the diagonal are kept and the remaining 30% are annihilated. When the spatial dependence is weak ($a = 0.03$), the boxplots across the different TLRs and the exact are identical, i.e., the medians and the standard deviations of the parameter estimates are almost equal. While the medians of the parameter estimates under DST 40/60 and DST 70/30 are close to the true...
parameter values, the estimates have more variability. As the dependence in space increase, i.e., $a = 0.09$ (moderate) and $a = 0.2$ (strong), TLR5 is insufficient and it obtains parameter estimates that are very far from the true value. While increasing the accuracy to TLR7 solves the problem for moderate spatial dependence, this level of accuracy is still inadequate for simulations with strong spatial dependence as the medians of the parameter estimates still do not coincide with the true parameter values. TLR9 remedies this problem. DST 40/60 and DST 70/30 give estimates that are far from the true values and they perform worse as the strength of spatial dependence increase. This is expected as the DST technique throws away significant amount of information in the cross-covariance matrix which is vital when the dependence in space is strong. All in all, while there are significantly more parameters to estimate in the parsimonious bivariate Matérn model, our estimation procedure can satisfactorily recover all of them. Furthermore, TLR approximation outperforms another approximation technique, i.e., DST and remains competitive with the exact model in terms of parameter estimation accuracy when using a higher accuracy level when there is stronger spatial dependence.

Using the parameter estimates in Experiment 2, we predict at the 2,500 unsampled locations and measure the multivariate MLOE/MMOM for all the TLR approximation models and the exact using Algorithm 2. Fig. 15 charts the behavior of the multivariate MLOE/MMOM of 5 randomly chosen sample bivariate random fields as we increase the accuracy levels. Commensurate to the findings in Experiment 2, bivariate random fields with higher spatial dependence necessitate TLR approximations with higher accuracy levels in order to remain competitive with the exact model.

### 5.4.2 Real Dataset

The parameter estimates for the parsimonious bivariate Matérn fitted to the wind dataset at $n = 104,490$ observation locations are presented in Table 1. We also fit the univariate Matérn covariance function and contrast the univariate to the bivariate prediction performance. Our estimation procedure obtains a positive colocated correlation parameter, $\hat{\beta} = 0.192$, a relatively high value for the two smoothness parameters, $\nu_{11} = 2.283$ and $\nu_{22} = 2.033$, and the spatial range parameter, $a = 0.161$. These results are consistent with our initial observations enumerated in Section V-A. Table 1 also shows the corresponding MSPE values for the predictions done on $n_{\text{pred}} = 11,610$ prediction locations. The gain in bivariate prediction is only apparent for the $V$ component, as the MSPEs of the univariate and bivariate predictions on the $U$ component are equal. These results reflect our findings in Experiment 1, i.e., when the colocated correlation between the two variables is near 0, the advantage of using a bivariate model is not so prominent.

Ultimately, we juxtapose the prediction performance of these geospatial statistical models with a well-known non-geospatial statistical approach called Inverse Distance
1. For kriging and cokriging models, the weights are functions not only of the distance but also of the spatial relationship among the sampled locations; see Equation (5).

2. **Conclusion**

We proposed a high-performance framework for modeling and inference of large spatial datasets, based on the multivariate geospatial statistical modeling concept. In the context of climate and weather applications, the framework can operate on the Gaussian log-likelihood function with two (or more) associated variables, for the purpose of estimating a parameter vector, in order to predict missing measurements. We showed from the experiments the benefits of multivariate over univariate modeling. Both the exact and TLR-based approximation computations of the MLE operations were implemented and evaluated on large-scale experiments. The TLR-based approximation for the MLE outperformed the fully double-precision exact MLE counterpart up to 10X and 2X on different hardware architectures. Comprehensive qualitative experiments were conducted to assess the accuracy of the TLR-based estimation and prediction. We demonstrated the effectiveness of the approximation technique in achieving high performance, while preserving a convenient accuracy level. Additionally, an algorithm to compute the newly proposed multivariate MLOE/MMOM criteria was devised. This algorithm allows for the assessment of the quality of the MLE operations involving approximated models.

**Acknowledgments**

The authors would like to thank NVIDIA Inc., Cray Inc., and Intel Corp., the Cray Center of Excellence and Intel Parallel Computing Center awarded to the Extreme Computing Research Center (ECRC) at KAUST. For computer time, this research used GPU-based systems as well as Shaheen supercomputer hosted at the Supercomputing Laboratory at King Abdullah University of Science and Technology (KAUST).
REFERENCES

[1] M. Asch, T. Moore, R. Badia, M. Beck, P. Beckman, T. Bidot, F. Bodin, F. Cappello, A. Choudhary, B. de Supinski, E. Deelman, J. Dongarra, Á. Dubey, G. Fox, H. Fu, S. Girona, W. Gropp, M. Heneux, Y. Ishikawa, K. Keane, D. Keyes, W. Kramer, J.-F. Lavignon, Y. Lu, S. Matsuoka, B. Mohr, D. Reed, S. Requena, J. Saltz, T. Schulthess, R. Stevens, M. Swany, A. Szalay, W. Tang, G. Varoquaux, J.-P. Villette, R. Wisniewski, Z. Xu, and I. Zacharov, “Big data and extreme-scale computing: Pathways to convergence—toward a shaping strategy for a future software and data ecosystem for scientific inquiry,” The International Journal of High Performance Computing Applications, vol. 32, no. 4, pp. 435–479, 2018.

[2] X.-k. Liao, K. Lu, C.-q. Yang, J.-w. Li, Y. Yuan, M.-c. Lai, L.-b. Huang, P.-j. Lu, J.-b. Fang, J. Ren, and J. Shen, “Moving from exascale to zettascale computing: challenges and techniques,” Frontiers of Information Technology & Electronic Engineering, vol. 19, no. 10, pp. 1236–1244, 2018.

[3] S. Abdulah, H. Ltaief, Y. Sun, M. G. Genton, and D. E. Keyes, “ExaGeoStat: A high performance unified software for geostatistics on manycore systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 12, pp. 2771–2784, 2018.

[4] N. Doucet, H. Ltaief, D. Gratadour, and D. Keyes, “Mixed-precision tomographic reconstruction algorithms on hardware accelerators,” in 2019 IEEE/ACM 5th Workshop on Irregular Applications - Architectures and Algorithms (IA3). IEEE, 2019, pp. 31–38.

[5] B. H. Park, H. A. Aziz, A. Morton, and R. Stewart, “High performance data driven agent-based modeling framework for simulation of commute mode choices in metropolitan area,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018, pp. 3779–3784.

[6] H.-T. Chu, “High-performance computing for measurement of cancer gene signatures,” Application of Omics, AI and Blockchain in Bioinformatics Research, vol. 21, p. 109, 2019.

[7] J. Drake and I. Foster, “Introduction to the special issue on parallel computing for measurement of cancer gene signatures,” Application of Omics, AI and Blockchain in Bioinformatics Research, vol. 21, p. 109, 2019.

[8] S. Abdulah, H. Ltaief, Y. Sun, M. G. Genton, and D. E. Keyes, “ExaGeoStat: A high performance unified software for geostatistics on manycore systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 12, pp. 2771–2784, 2018.

[9] N. Doucet, H. Ltaief, D. Gratadour, and D. Keyes, “Mixed-precision tomographic reconstruction algorithms on hardware accelerators,” in 2019 IEEE/ACM 5th Workshop on Irregular Applications - Architectures and Algorithms (IA3). IEEE, 2019, pp. 31–38.

[10] B. H. Park, H. A. Aziz, A. Morton, and R. Stewart, “High performance data driven agent-based modeling framework for simulation of commute mode choices in metropolitan area,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018, pp. 3779–3784.

[11] H.-T. Chu, “High-performance computing for measurement of cancer gene signatures,” Application of Omics, AI and Blockchain in Bioinformatics Research, vol. 21, p. 109, 2019.

[12] J. Drake and I. Foster, “Introduction to the special issue on parallel computing for measurement of cancer gene signatures,” Application of Omics, AI and Blockchain in Bioinformatics Research, vol. 21, p. 109, 2019.

[13] S. Abdulah, H. Ltaief, Y. Sun, M. G. Genton, and D. E. Keyes, “ExaGeoStat: A high performance unified software for geostatistics on manycore systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 12, pp. 2771–2784, 2018.

[14] N. Doucet, H. Ltaief, D. Gratadour, and D. Keyes, “Mixed-precision tomographic reconstruction algorithms on hardware accelerators,” in 2019 IEEE/ACM 5th Workshop on Irregular Applications - Architectures and Algorithms (IA3). IEEE, 2019, pp. 31–38.

[15] B. H. Park, H. A. Aziz, A. Morton, and R. Stewart, “High performance data driven agent-based modeling framework for simulation of commute mode choices in metropolitan area,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018, pp. 3779–3784.

[16] H.-T. Chu, “High-performance computing for measurement of cancer gene signatures,” Application of Omics, AI and Blockchain in Bioinformatics Research, vol. 21, p. 109, 2019.
Mary Lai Salvaña is a PhD student in the Spatio-Temporal Statistics & Data Science group at King Abdullah University of Science and Technology (KAUST). She received her MS and BS degrees in Applied Mathematics in 2015 and 2016 from Ateneo de Manila University, Philippines. Her research interest includes multivariate spatio-temporal statistics and high-performance computing for large spatial and spatio-temporal datasets.

Sameh Abdullatif is a research scientist at the Extreme Computing Research Center, King Abdullah University of Science and Technology (KAUST). Before working at KAUST, he did research on statistical computing for climate applications as a postdoc at the National Center for Atmospheric Research (NCAR), the Statistical and Applied Mathematical Sciences Institute (SAMSI), and Duke University. He received his Ph.D. in Statistics in 2017 from KAUST, master, and bachelor in Mathematics in 2014 and 2011 from Fudan University. His research interest includes spatio-temporal statistics, functional data analysis, Bayesian modeling, machine learning, and high-performance computing for large datasets.

Huang Huang is a research scientist in the Spatio-Temporal Statistics & Data Science group at King Abdullah University of Science and Technology (KAUST). Before working at KAUST, she did research on statistical computing for climate applications as a postdoc at the National Center for Atmospheric Research (NCAR), the Statistical and Applied Mathematical Sciences Institute (SAMSI), and Duke University. She received her Ph.D. in Statistics in 2017 from KAUST, master, and bachelor in Mathematics in 2014 and 2011 from Fudan University. Her research interest includes spatio-temporal statistics, functional data analysis, Bayesian modeling, machine learning, and high-performance computing for large datasets.

Hatem Ltaief is the Principal Research Scientist in the Extreme Computing Research Center at King Abdullah University of Science and Technology, Saudi Arabia. His research interests include parallel numerical algorithms, parallel programming models, and performance optimizations for multicore architectures and hardware accelerators. His current research collaborators include Aramco, Total, Observatoire de Paris, NVIDIA, and Intel.

Ying Sun is an Associate Professor of Statistics at King Abdullah University of Science and Technology (KAUST) in Saudi Arabia. She joined KAUST in June 2014 after one year of service as an assistant professor in the Department of Statistics at Ohio State University, USA. At KAUST, she leads a multidisciplinary research group on environmental statistics, dedicated to developing statistical models and methods for using space-time data to solve important environmental problems. Prof. Sun received her Ph.D. degree in Statistics from Texas A&M University in 2011, and was a postdoctoral researcher in the research network of Statistics in the Atmospheric and Oceanic Sciences (STATMOS), affiliated with the University of Chicago and the Statistical and Applied Mathematical Sciences Institute (SAMS). Her research interests include spatio-temporal statistics with environmental applications, computational methods for large datasets, uncertainty quantification and visualization, functional data analysis, robust statistics, and statistics of extremes.

Marc G. Genton is a Distinguished Professor of Statistics at King Abdullah University of Science and Technology (KAUST) in Saudi Arabia. He received his Ph.D. in Statistics from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 1996. He also holds a M.Sc. in Applied Mathematics Teaching and a degree of Applied Mathematics Engineering from the same institution. Prof. Genton is a Fellow of the American Statistical Association, of the Institute of Mathematical Statistics, and the American Association for the Advancement of Science, and is an elected member of the International Statistical Institute. In 2010, he received the El-Shaarawi award for excellence from the International Environmetrics Society and the Distinguished Achievement award from the Section on Statistics and the Environment of the American Statistical Association. Prof. Genton’s research interests include statistical analysis, flexible modeling, prediction, and uncertainty quantification of spatio-temporal data, with applications in environmental and climate science, renewable energies, geophysics, and marine science.

David Keyes received the BSE degree in aerospace and mechanical sciences from Princeton University in 1978 and the PhD degree in applied mathematics from Harvard University in 1984. He works at the interface between parallel computing and the numerical analysis of PDEs, with a focus on scalable implicit solvers, Newton-Krylov-Schwarz (NKS) and Additive Schwarz Preconditioned Inexact Newton (ASPIN) are methods that he helped develop and popularize. Before joining KAUST as a founding dean, in 2009, where he currently directs the Extreme Computing Research Center, he led multi-institutional projects researching and developing scalable solver software in the SciDAC and ASCI programs of the US DOE, ran University collaboration programs at LLNL’s ICASE, and taught at Columbia, Old Dominion, and Yale Universities. He is a fellow of SIAM, AMS, and the AAAS, and has been awarded the ACM Gordon Bell Prize, the IEEE Sidney Fernbach Award, and the SIAM Prize for Distinguished Service to the Profession.