Accelerating Geostatistical Modeling and Prediction With Mixed-Precision Computations: A High-Productivity Approach with PaRSEC

Accelerating Geostatistical Modeling and Prediction With Mixed-Precision Computations: A High-Productivity Approach with PaRSEC

Sameh Abdulah, Qinglei Cao, Yu Pei, George Bosilca, Jack Dongarra,
Marc G. Genton, David E. Keyes, Hatem Ltaief, and Ying Sun

Abstract—Geostatistical modeling, one of the prime motivating applications for exascale computing, is a technique for predicting desired quantities from geographically distributed data, based on statistical models and optimization of parameters. Spatial data is assumed to possess properties of stationarity or non-stationarity via a kernel fitted to a covariance matrix. A primary workhorse of stationary spatial statistics is Gaussian maximum log-likelihood estimation (MLE), whose central data structure is a dense, symmetric positive definite covariance matrix of dimension of the number of correlated observations. Two essential operations in MLE are the application of the inverse and evaluation of the determinant of the covariance matrix. These can be rendered through the Cholesky decomposition and triangular solution. In this contribution, we reduce the precision of weakly correlated locations to single- or half- precision based on distance. We thus exploit mathematical structure to migrate MLE to a three-precision approximation that takes advantage of contemporary architectures offering BLAS3-like operations in a single instruction that are extremely fast for reduced precision. We illustrate application-expected accuracy worthy of double-precision from a majority half-precision computation, in a context where uniform single precision is by itself insufficient. In tackling the complexity and imbalance caused by the mixing of three precisions, we deploy the PaRSEC runtime system. PaRSEC delivers on-demand casting of precisions while orchestrating tasks and data movement in a multi-GPU distributed-memory environment within a tile-based Cholesky factorization. Application-expected accuracy is maintained while achieving up to $1.59X$ by mixing FP64/FP32 operations on 1536 nodes of HAWK or 4096 nodes of Shaheen II, and up to $2.64X$ by mixing FP64/FP32/FP16 operations on 128 nodes of Summit, relative to FP64-only operations, This translates into up to 4.5, 4.7, and 9.1 (mixed) PFlop/s sustained performance, respectively, demonstrating a synergistic combination of exascale architecture, dynamic runtime software, and algorithmic adaptation applied to challenging environmental problems.

Index Terms—Climate/Weather Prediction, Dynamic Runtime Systems, Geospatial Statistics, High Performance Computing, Multiple Precisions, User-Productivity.

- *S. Abdulah, M. G. Genton, D. E. Keyes, H. Ltaief, and Y. Sun with Computer, Electrical, and Mathematical Sciences and Engineering Division (CEMSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia. E-mail: {sameh.abdulah, marc.genton, david.keyes, hatem.ltaief, ying.sun}@kaust.edu.sa*
Q. Cao, Y. Pei, G. Bosilca, and J. Dongarra with the Innovative Computing Laboratory, University of Tennessee, Knoxville, TN 37996, US. E-mail: {qcao3, ypei2}@vols.utk.edu, {bosilca, dongarra}@icl.utk.edu

Manuscript received mm dd, yyyy; revised mm dd, yyyy.

1 Introduction

Geostatistics is a means of modeling and predicting desired quantities from spatially distributed data based on statistical assumptions and optimization of parameters. It is complementary to first-principles modeling approaches rooted in conservation laws and typically expressed in PDEs. Alternative statistical approaches to predictions from first-principles methods, such as Monte Carlo sampling wrapped around simulations with a distribution of inputs, may be vastly more computationally expensive than sampling from a distribution based on a much smaller number of simulations. Geostatistics is relied upon for economic and policy decisions for which billions of dollars or even lives are at stake, such as engineering safety margins into developments, mitigating hazardous air quality, locating fixed renewable energy resources, and planning agricultural yields or weather-dependent tourist revenues. Climate and weather predictions are among the principal workloads occupying supercomputers around the world and planned for exascale computers, so even minor improvements for production applications pay large dividends. A wide variety of such codes have migrated or are migrating to mixed-precision environments; we describe a novel migration of one important class of such codes.

A main computational kernel of stationary spatial statistics considered herein is the evaluation of the Gaussian log-likelihood function, whose central data structure is a dense covariance matrix of the dimension of the number of (presumed) correlated observations, which is generally the product of the number of observation locations and the number of variables observed at each location. In the maximum log-likelihood estimation (MLE) technique considered herein, two essential operations on the covariance matrix are the application of its inverse and evaluation of its determinant. These operations can all be rendered through the classical Cholesky decomposition and triangular solution, occurring inside the optimization loop that fits statistical model parameters to the input data. The covariance matrix is dense, symmetric, and positive definite, and possesses a mathematical structure arising from its physical origin that motivates approximations of various kinds for highdimensional problems, especially in view of the demands on storage and computation of the Cholesky formulation. ExaGeoStat [1] is designed to provide controllable approximations to extreme-scale MLE problems by introducing novel algorithmic, architectural, and programming model features and packing the power of hybrid distributed-shared memory computing under the high-productivity statistical package R. Based on tile algorithms [2], the resulting Cholesky factorization takes advantage of the covariance matrix structure, which under a proper ordering [3] clusters the most significant information around the diagonal.

We introduce ExaGeoStat_PaRSEC, i.e., ExaGeoStat powered by PaRSEC, extending the approach in [4] to accelerate the Cholesky factorization by mixing FP64 double-precision (DP), FP32 single-precision (SP) and FP16 half-precision (HP) to take advantage of the tensor cores of modern GPUs, e.g., NVIDIA V100s. Precision adaptation inveighs against predictable load-balancing, which therefore requires reliance on a dynamic runtime system to schedule computationally rich tasks of tile-sized granularity and data exchanges. The nimble runtime system PaRSEC is leveraged to deal with the complexity of the proposed mixed-precision algorithm, tackle the introduced imbalance, and limit the memory usage on distributed-memory systems equipped with multiple GPUs. While mixed-precision algorithmic optimizations translate into performance gains, we still guarantee application-expected accuracy that drives the modeling and the ultimate prediction phases for climate and weather applications. To the best of our knowledge, this work is the first to highlight performance of large-scale, task-based, and three-precision Cholesky factorization for geostatistical modeling and prediction. Among the architectural imperatives for exascale computing discussed in [5], we: (1) reside on average higher on the memory hierarchy by selectively using reduced precision words, (2) reduce artifactual synchronizations, (3) exploit specialized SIMD/SIMT instructions, and (4) exploit heterogeneity.

Our main contributions are as follows: (1) powering the ExaGeoStat framework with the PaRSEC runtime system and demonstrating their ability to perform modeling and prediction on geospatial data using MLE with a novel mixed-precision implementation of DP, SP and HP in a Cholesky factorization; (2) optimizing the performance of mixed-precision Cholesky factorization by shepherding the task execution order and balancing the GPU workloads; (3) validating accuracy via synthetic datasets and real datasets; and (4) performing large-scale mixed-precision Cholesky factorization on AMD-based, Intel-based CPU systems and IBM-based multi-GPU system with up to $196,608$ cores, $131,072$ cores and 768 GPUs respectively.

The remainder of the paper is organized as follows. Section 2 covers related work. Section 3 gives a brief overview of the problem. Section 4 describes the ExaGeoStat framework and PaRSEC dynamic runtime system. Section 5 describes the proposed mixed-precision Cholesky approach. Section 6 highlights how PaRSEC helps to tune the performance of ExaGeoStat with the three precisions approximation of the MLE operation in a single Cholesky factorization. Section 7 analyses accuracy using synthetic and real datasets in the context of climate/weather applications and illustrates the performance results. We conclude in Section 8.

2 Related Work

This section gives a brief review of the existing works on both mixed-precision in climate/weather applications and the existing efforts on runtime systems to accelerate large-scale applications.

Large-Scale Climate/Weather Modeling. Large-scale modeling is often prohibitive in climate/weather applications. In literature, numerous approximation algorithms have been proposed to be able to analyse big geospatial data and reduce the arithmetic complexity and memory footprint in extreme problems. One way is to convert the given dense covariance to a sparse matrix by replacing values of large distance correlations with zero. In this case, sparse matrices algorithms [6] or covariance tapering strategy [1] can be used for fast computation. Dimension reduction is another way to approximate and generate the covariance matrix. For instance, the authors in [7] propose the Gaussian Predictive Processes (GPP) to achieve the reduction by projecting the original problem space into a subspace at a certain set of locations. Although such means can reduce the complexity of estimating the model parameters, they usually underestimate the variance parameter [8]. Other methods of dimension reduction include Kalman filtering [9], moving averages [10], and low-rank splines [11]. Large covariance matrix dimension has been also widely accommodated using Hierarchical matrices ($\mathcal{H}$-matrices) and low-rank approximations. In the literature, different data approximation techniques based on $\mathcal{H}$-matrices have been proposed such as, Tile Low-Rank (TLR) [12], [13], Hierarchically Off-Diagonal Low-Rank (HODLR) [14], [15], Hierarchically Semi-Separable (HSS) [16], or $\mathcal{H}_2$-matrices [17], [18].

Mixed-Precision in Climate/Weather Applications and Beyond. To the best of our knowledge, existing works on mixed-precision and climate/weather applications are related to studying the impact of applying mixed-precision computation on the modeling operation. For instance, the work in [19] provides a study on the effect of faulty hardware low precision arithmetic on the accuracy of weather and climate prediction. The authors have proved that such faults have no impact on the overall accuracy of such applications. In [20], the authors show how single- and half- precision can replace full double-precision calculations for weather and climate applications which can maintain the desirable accuracy at the end. In [21], mixed-precision Krylov sub-space solver for climate/weather applications has been proposed. The study shows numerical instabilities that impact the accuracy of prediction. For solving a linear system of equations, mixed-precision iterative refinement approaches have been studied using FP64/FP32 arithmetics for sparse and dense linear algebra [22], [23], and lately extended with FP16 [24], [25].

Runtime Systems. With the increased complexity of the underlying hardware, delivering performance while abstracting the hardware becomes critical. Beyond just MPI+X, more revolutionary solutions explore more dynamic, task-based systems as a substitute solution to both local and distributed data dependencies management. The ideas behind are similar to the concepts put forward in workflow, parallelizing an algorithm over a heterogeneous set of distributed resources by dividing it into sets of interdependent tasks and organizing the data transfers to maximize the occupancy of most resources. Many efforts to provide such an abstraction via a fine-grain, task-based dataflow programming exist, adding to those that have transitioned from a grid-based workflow toward a task-based environment. Some of the recent task-based runtimes like `OmpSs` [26], `StarPU` [27], `OpenMP` [28], `Legion` [29], `HPX` [30], and `PaRSEC` [31], among others, abstract the available resources to isolate application developers from the underlying hardware complexity and simplify the process of writing massively parallel scientific applications.

In this paper, we focus on mixed-precision arithmetic to approximate and accelerate large-scale climate/weather prediction applications. In particular, we extend the mixed two-precision arithmetic approach [4] initially based on `StarPU` to `PaRSEC` instead with mixed three-precision computations. This represents much more than a simple swap between runtimes. The precision conversion becomes now a runtime decision made by `PaRSEC` as opposed to a user decision with `StarPU` in [4]. This permits to provide on-demand casting of precisions, while orchestrating tasks and data movement on distributed-memory environment systems equipped with multiple GPU hardware accelerators. `PaRSEC` is now empowered by not only task scheduling and data motion but also converting data precision at runtime to match the task operand datatypes. We integrate this novel high productive programming model based on `PaRSEC` into `ExaGeoStat` [1] and assess their synergism on large-scale environmental applications using massively parallel homogeneous and heterogeneous systems.

3 Overview of Geospatial Modeling

Tackling the complexity of large-scale geospatial modeling in the context of climate/weather applications requires efficient algorithms that are able to provide an accurate estimation of the underlying spatial model with the aid of leading-edge hardware architectures. This section provides a brief background on geospatial modeling and prediction from a statistical point of view.

Climate Modeling and Prediction using MLE. Spatial data associated with climate and weather applications consist of a set of locations regularly or irregularly distributed across a given specific geographical region where each location is linked with climate or environmental variables, such as soil moisture, temperature, humidity, or wind speed. In geostatistics, spatial data are usually modeled as a realization from a Gaussian spatial random field. Assume a realization of a Gaussian random field $\mathbf{Z} = \{Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n)\}^\top$ at a set of n spatial locations $\mathbf{s}_1, \ldots, \mathbf{s}_n$ in $\mathbb{R}^d$, $d \geq 1$. We assume a stationary and isotropic Gaussian random field with mean zero and a parametric covariance function $C(\mathbf{h};\boldsymbol{\theta}) = \operatorname{cov}\{Z(\mathbf{s}), Z(\mathbf{s}+\mathbf{h})\}$, where $\mathbf{h} \in \mathbb{R}^d$ is a spatial lag vector and $\boldsymbol{\theta} \in \mathbb{R}^q$ is an unknown parameter vector of interest. $C(\mathbf{h};\boldsymbol{\theta})$ values depend on the distance between any two locations and denoted by $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ with entries $\boldsymbol{\Sigma}_{ij} = C(\mathbf{s}_i - \mathbf{s}_j;\boldsymbol{\theta})$, $i, j = 1, \ldots, n$. The matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is symmetric and positive definite. Statistical inference about $\boldsymbol{\theta}$ is often based on the Gaussian log-likelihood function as follows:

$$\ell(\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}(\boldsymbol{\theta})| - \frac{1}{2}\mathbf{Z}^\top\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}\mathbf{Z}. \quad (1)$$

The modeling operation depends on computing $\widehat{\boldsymbol{\theta}}$, the parameter vector that maximizes Equation (1). When the number of locations n is large, the evaluation of the likelihood function becomes computationally challenging due to the Cholesky factorization, requiring $\mathcal{O}(n^3)$ flops and $\mathcal{O}(n^2)$ memory. The estimated $\widehat{\boldsymbol{\theta}}$ can be used to predict missing measurements at some other locations in the same region. Prediction can be represented as a multivariate normal joint distribution with the existing n known measurements $\mathbf{Z}_n$ and m missing measurements $\mathbf{Z}_m$ [32], [33] as follows:

$$\begin{bmatrix}\mathbf{Z}_m\\ \mathbf{Z}_n\end{bmatrix} \sim N_{m+n}\left(\begin{bmatrix}\boldsymbol{\mu}_m\\ \boldsymbol{\mu}_n\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{mm} & \boldsymbol{\Sigma}_{mn}\\ \boldsymbol{\Sigma}_{nm} & \boldsymbol{\Sigma}_{nn}\end{bmatrix}\right), \quad (2)$$

with $\boldsymbol{\Sigma}_{mm} \in \mathbb{R}^{m\times m}$, $\boldsymbol{\Sigma}_{mn} \in \mathbb{R}^{m\times n}$, $\boldsymbol{\Sigma}_{nm} \in \mathbb{R}^{n\times m}$, and $\boldsymbol{\Sigma}_{nn} \in \mathbb{R}^{n\times n}$. The associated conditional distribution can be represented as

$$\begin{aligned}\mathbf{Z}_m|\mathbf{Z}_n \sim N_m(&\boldsymbol{\mu}_m + \boldsymbol{\Sigma}_{mn}\boldsymbol{\Sigma}_{nn}^{-1}(\mathbf{Z}_n - \boldsymbol{\mu}_n),\\ &\boldsymbol{\Sigma}_{mm} - \boldsymbol{\Sigma}_{mn}\boldsymbol{\Sigma}_{nn}^{-1}\boldsymbol{\Sigma}_{nm})\end{aligned} \quad (3)$$

Assuming that the observed vector $\mathbf{Z}_n$ has a zero-mean function (i.e., $\boldsymbol{\mu}_m = \mathbf{0}$ and $\boldsymbol{\mu}_n = \mathbf{0}$), the unknown vector $\mathbf{Z}_m$ can be predicted [32] by solving

$$\mathbf{Z}_m = \boldsymbol{\Sigma}_{mn}\boldsymbol{\Sigma}_{nn}^{-1}\mathbf{Z}_n, \quad (4)$$

with associated prediction uncertainty given by

$$\mathbf{U}_m = diag[\boldsymbol{\Sigma}_{mm} - \boldsymbol{\Sigma}_{mn}\boldsymbol{\Sigma}_{nn}^{-1}\boldsymbol{\Sigma}_{nm}] \quad (5)$$

where $diag$ denotes the diagonal of a matrix.

Computing the last two equations is challenging since they require applying the Cholesky factor of the covariance matrix during the forward and backward substitutions on several right-hand sides.

Covariance Functions. Constructing a corresponding covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ for a set of given locations in MLE modeling or prediction operations requires defining a covariance function to describe the correlation over a given distance matrix. The Matérn family [34] has shown its ability on a wide variety of applications, for example, geostatistics and spatial statistics [35] and machine learning [36]. In this study, we are interested in the powered exponential covariance function [37] to model the geospatial data, an alternative to the general Matérn covariance function. The powered exponential covariance function is defined as:

$$C(r;\boldsymbol{\theta}) = \theta_0 \exp\left(\frac{-r^{\theta_2}}{\theta_1}\right), \quad (6)$$

where $r = \|\mathbf{s} - \mathbf{s}'\|$ is the distance between two spatial locations s and s′, and $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2)^\top$. Here $\theta_0 > 0$ is the variance, $\theta_1 > 0$ is a spatial range parameter that measures how quickly the correlation of the field decays with distance, and $\theta_2 > 0$ controls the smoothness of the random field, with larger values of θ_2 corresponding to smoother fields.

4 Powering ExaGeoStat with PaRSEC

We provide essential information on the high-performance geostatistics modeling software `ExaGeoStat` and dynamic runtime system `PaRSEC` before highlighting their synergism to solve large-scale environmental applications.

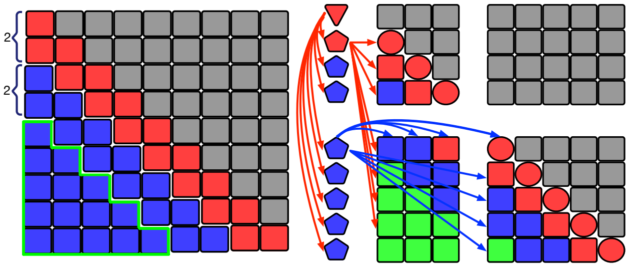
The ExaGeoStat Framework. ExaGeoStat [1] is a computational software for geostatistical and environmental applications. ExaGeoStat has three main components, namely, the synthetic data generator, the modeling tool, and the predictor. It provides a generic tool for generating a reference set of synthetic measurements and locations, which generates test cases of prescribed size for standardizing comparisons with other methods. This tool facilitates the assessment of the quality of any proposed approximation method with a wide range of datasets with different features. ExaGeoStat performs modeling based on the maximum likelihood estimation (MLE) approach (see Eq. 1). ExaGeoStat depends on various software libraries to provide a unified framework that is able to run on different parallel hardware architectures. The overall MLE optimization is performed using the NLOPT optimization library [38] which aims at maximizing the likelihood estimation function by using different sets of the statistical model parameters based on the given covariance function. Furthermore, to perform the underlying linear algebra matrix operations, ExaGeoStat relies on the state-of-the-art numerical libraries Chameleon [39] (for dense operator [1]) and HiCMA [40] (for data-sparse operator [41]). Both libraries rely on task-based programming models that enable fine-grained asynchronous computations by splitting the matrix operator into tiles. The numerical algorithm is translated into a Directed Acyclic Graph (DAG), where the nodes represent tasks and the edges define data dependencies. The dynamic runtime system deploys the tasks across different hardware resources, while ensuring the integrity of data dependencies. The runtime might orchestrate task scheduling and overlap communication with computations to reduce load imbalance, while maintaining high occupancy. Last but not least, the ExaGeoStat predictor tool aims at predicting a set of unknown measurements at new spatial locations using the parameters (i.e., $\widehat{\boldsymbol{\theta}}$ vector) estimated during the modeling phase, as explained in Section 3. In the literature, we assess the prediction quality with the mean squared prediction error (MSPE), which can be computed as: MSPE $= \frac{1}{m}\sum_{l=1}^{m} \|\widehat{\mathbf{Z}}(\mathbf{s}_{0,l}) - \mathbf{Z}(\mathbf{s}_{0,l})\|^2$, where $\mathbf{s}_{0,1}, \mathbf{s}_{0,2}, \ldots, \mathbf{s}_{0,m}$ are the m prediction locations.

PaRSEC Dynamic Runtime System. PaRSEC [42], an event-driven task-based runtime for distributed heterogeneous architectures based on data-flow, is capable of dynamically unfolding a description of a DAG of tasks onto a set of resources. PaRSEC understands data dependencies and efficiently shepherds data between memory spaces (between nodes but also between different memories on different devices) and schedules tasks across heterogeneous resources. PaRSEC facilitates the design of Domain Specific Languages (DSLs) [43] that allow domain experts to focus on their scientific application rather than on the underlying complex hardware architecture. These DSLs rely on a data-flow model to create dependencies between tasks and target the expression of maximal parallelism with high productivity in mind. The DSL used in this paper, Parameterized Task Graph (PTG) [44], uses a concise, parameterized, task-graph description known as Job Data Flow (JDF) to represent the dependencies between tasks. The main algorithmic idea is that the unfolding of the parameterized description may eventually lead to a complete description of data dependencies between tasks from the DAG. Similar to other runtimes, the task execution order depends on a set of data dependencies (e.g., read, write, and read-write) defined over the application data. The distributed runtime scheduler assigns sets of tasks to the available processing unit based on these dependencies which may lead to runtime opportunities for asynchronous executions. To enhance the productivity of the application developers, PaRSEC implicitly infers all communications from the expression of the tasks, supporting one-to-many and many-to-many types of communications. PaRSEC supports different programming languages (e.g., Pthreads, CUDA, OpenCL, and MPI) and runs on different hardware architectures (e.g., CPU/GPU, shared/distributed-memory). From a performance standpoint, algorithms described in PTG have been shown capable of delivering a significant percentage of the hardware peak performance on many hybrid distributed-memory machines for several scientific fields [45]–[49].

In this paper, we leverage PaRSEC runtime system within ExaGeoStat to perform operations beyond what a traditional runtime system does. These operations are inherent to the application but can be offloaded to runtimes, in addition to their current duties of data movement and task scheduling. In particular, we empower PaRSEC with mixed-precision support to enable approximation within ExaGeoStat for climate/weather prediction applications. It becomes PaRSEC's responsibility to convert on-the-fly the precision arithmetic according to the datatypes of the task operands, as explained in the next section.

5 ExaGeoStat Multi-Precision Cholesky Factorization for MLE

We design a mixed-precision approach for the Cholesky factorization targeting the MLE climate modeling and prediction. We apply tile-centric precision arithmetic by exploiting the data sparsity structure of the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. The correlations between nearby geospatial locations are strong and usually reside around the matrix diagonal, thanks to Morton ordering [3]. As we move away from the main diagonal, the correlations between remote geospatial locations weaken, and we capture this in the computation by relying on a band strategy to appropriately select the precision of the tiles C_{ij} based on their row and column coordinates (i, j) in the global matrix, with $i \geq j$ considering the lower triangular part of the symmetric matrix. This approach is generic and accommodates for as many precisions as necessary, but for the sake of simplicity we will use a three-precision approach in the rest of this paper. The tiles are tagged accordingly with DP, SP, and even HP precision arithmetic for $i \sim j$, $i > j$, and $i \gg j$, respectively. More precisely, we introduce band_size_dp and band_size_sp (the number of bands/sub-diagonals) to control the tile precision located in the DP and SP band regions. The remaining tiles are located in the HP band region. We rely on the standard Two-Dimensional Block Cyclic Data Distribution (2DBCDD) to describe how the matrix tiles are shared among a grid of processors in a distributed-memory environment.



(a) Data storage. (b) Data flow.

Fig. 1: Mixed-precision Cholesky: (a) data storage and (b) data flow, and both with `band_size_dp` = 2 and `band_size_sp` = 2 of a matrix with 9 × 9 tiles. Colors for tiles/arrows represent different precisions: `DP` in red; `SP` in blue; `HP` in green. In (b), data-flow for the 1st panel factorization with different shapes/kernels: triangle / `POTRF`, square / `GEMM`, pentagon / `TRSM`, and circle / `SYRK`.

Algorithm 1: Mixed-Precision Cholesky.

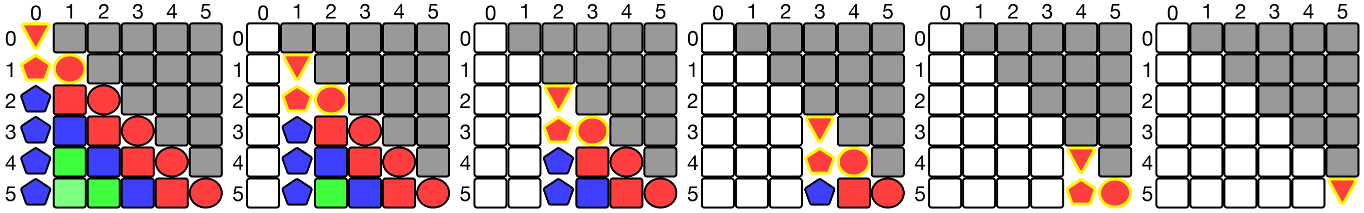
```
for $k = 0$ to $NT - 1$ /* Panel Factorization (PF) */
    DPOTRF ($C_{kk}$)
    for $m = k + 1$ to $NT - 1$
        if $m - k < band\_size\_dp$
            DTRSM ($C_{kk}, C_{mk}$)
        else
            STRSM ($C^{*S}_{kk}, C_{mk}$)
    for $m = k + 1$ to $NT - 1$
        DSYRK ($C^{*D}_{mk}, C_{mm}$)
    for $m = k + 2$ to $NT - 1$ /* Trailing Submatrix Update */
        for $n = k + 1$ to $m - 1$
            if $m - n < band\_size\_dp$
                DGEMM ($C^{*D}_{mk}, C^{*D}_{nk}, C_{mn}$)
            else if $m - n < band\_size\_dp + band\_size\_sp$
                SGEMM ($C_{mk}, C^{*S}_{nk}, C_{mn}$)
            else
                HGEMM ($C^{*H}_{mk}, C^{*H}_{nk}, C^{*H}_{mn}$)
```

Fig. 1a shows the tile-centric precision format for data storage in the proposed three-precision approach. Since `HP` is currently only supported for the `GEMM` operation (i.e., `HGEMM`), we generate the data in the parts corresponding to `HP`, in other terms below the `band_size_sp` (e.g., parts with green contour in Fig. 1a), in `SP`. This is still an advantage in terms of memory footprint compared to the traditional mixed-precision iterative refinement (IR) methods [24], [25]. Due to the tile storage, our approach is not required to maintain multiple copies of the original matrix with different precisions like IR methods do. We only have a single copy of the matrix containing a collection of tiles with various precisions. The data-flow of the mixed-precision Cholesky is the same as the regular single-precision Cholesky except that now it also encapsulates the datatype information for each operand of the computational tasks. Fig. 1b depicts the representative data-flow during the first Panel Factorization (PF) that engenders communications (red and blue arrows). There are two possible modes of operations as far as the handling of the precision conversions is concerned. The sender-based approach first converts the data tile locally to the required precisions for all its dependents before sending it. The receiver-based approach receives the remote data tile in its original precision before locally converting it to the required precision. Although the sender-based approach sends the data tile in the right precision required at the destination, it may end up sending several copies of the same data tile with different precisions to the same processor due to the 2DBCDD. On the other hand, the receiver-based approach may receive the data tile at a different precision from what is needed for the local task and needs a type conversion. However, there is only a single copy of the remote data tile with its original precision, leading to a reduction in network traffic. The receiver-based approach is the one we adopt throughout the paper.

Algorithm 1 details the new mixed-precision Cholesky factorization for lower triangular matrices composed by $NT \times NT$ tiles using `DP`, `SP` and `HP`. The resulting pseudocode structure is quite similar to the regular Cholesky factorization using one precision with the usual computational phases, i.e., the PF and the update of the trailing submatrix. The naming conventions for the numerical kernels follow the concatenation of "precision" and "kernel", where "precision" can be D (`DP`), S (`SP`) or H (`HP`) and "kernel" represents `POTRF`, `TRSM`, `SYRK` or `GEMM`. Moreover, the operands of the tasks with superscripts (i.e., *D, *S, or *H) indicate that once received, they may (*or may not* in case of the source and target precisions of the data tile are the same) need to be eventually converted from their current precision to the required precision of the kernels. Fig. 2 demonstrates Algorithm 1 by unrolling the entire algorithm of the mixed-precision Cholesky factorization with 6 × 6 tiles, `band_size_dp` = 2, and `band_size_sp` = 1. At the beginning of the factorization, numerical kernels with all three precisions, i.e., `DP`, `SP` and `HP`, operate at the same time. The tasks operating on the tiles with yellow boundaries are launched sequentially since they belong to the critical path of the DAG for that PF. These tasks need to be overlapped with sufficient task parallelism coming from the updates of the trailing submatrix (see Algorithm 1) in order to reduce idle time. As the factorization proceeds, tasks in `HP` disappear, and only tasks in `DP`/ `SP` continue to operate, starting from the 3rd PF. As we reach the end of the factorization in the 5th PF, we observe only `DP` tasks. This mixture of three precisions for the Cholesky factorization necessitates runtime decisions to provide on-demand casting of precision. The support for multiple precisions inherently brings load imbalance to an algorithm that may be otherwise regular. These load imbalance issues require novel runtime features and optimizations to maximize performance while ensuring high user-productivity.

6 PaRSEC Runtime Optimizations

We embed the support of multiple precisions into `PaRSEC` by incorporating the datatype information of the task operands into the data-flow. To our knowledge, this is the first time a runtime system provides a precision-agnostic mechanism to seamlessly handle workloads with variable precisions. This comes at the cost of introducing load imbalance in terms of computations and communications. But this performance bottleneck falls back into the original duty of dynamic runtime systems.



(a) 1st PF. (b) 2nd PF. (c) 3rd PF. (d) 4th PF. (e) 5th PF. (f) 6th PF.

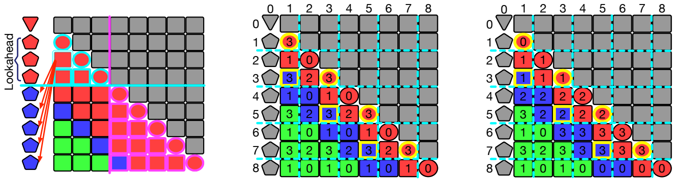
Fig. 2: Mixed-precision Cholesky factorization with 6×6 tiles, band_size_dp $= 2$, and band_size_sp $= 1$. White tiles represent the completed task. Other colors represent different precisions for each tile: DP in red, SP in blue, and HP in green. Different shapes indicate different kernels: triangle POTRF, square GEMM, pentagon TRSM, and circle SYRK.

Load Imbalance. Although the total number of operations is the same for each precision variant, performing HP computations is usually twice faster than SP, which is in turn usually twice faster than DP. With the recent advances in hardware compute capabilities (e.g., NVIDIA Tensor Cores), these performance speedups increase disproportionally for lower precision computations, especially for the GEMM kernels that represent the most critical tasks for the Cholesky factorization. Moreover, communications get also impacted by load imbalance. The mixed-precision Cholesky factorization may necessitate data movement involving tiles with various precisions, as highlighted in Fig. 1b with the red/blue arrows. To mitigate the load imbalance issue, we design and implement two optimizations to guide PaRSEC at runtime.

Lookahead Strategy. We apply a versatile lookahead strategy, which permits to hide tasks located in the critical path of every panel factorization with concurrent tasks (i.e., updates of the trailing submatrix), as explained in Section 5. This is a standard strategy used in linear algebra libraries [48], [50], [51] to hide communication and limit idle time. We further extend this strategy to mitigate the overhead of load imbalance in the context of mixed-precision workloads. The main idea consists in giving a higher scheduling priority to tasks that belong to the critical path than tasks that reside outside of the critical path. In fact, tasks that permit to directly unlock data dependencies of those executed in the critical path are also promoted with a higher scheduling priority. We define the depth of the lookahead as a tunable parameter that dynamically changes based on the structure of the mixed-precision matrix.

We implement this strategy within PaRSEC by utilizing the concept of control dependency between tasks. These additional control dependencies guide the task execution order and infer the proper priorities by adding empty dependencies (without extra communication). In particular, we apply control dependencies in the panel factorization k in Algorithm 1 between the top DGEMM ($m = k + 2$ and $n = k + 1$, the utmost important task to release the DTRSM in the critical path of the next panel factorization) and xTRSMs with $m - k > \mathit{lookahead}$ in the same panel factorization. In this way, tasks with the lower precision that are far away from the critical path will be delayed, prioritizing the critical path, expediting the discovery of the next panel factorization and eventually accelerating the whole Cholesky factorization. Fig. 3a presents a lookahead set to three, which prioritizes upcoming tasks of the critical path within the next three panels (i.e., the cyan boundary tiles in Fig. 3a) over the non-critical tasks (i.e., the magenta boundary tiles in Fig. 3a released by the red arrows data dependencies) that would otherwise delay progress in computations. Meanwhile, tasks operating on these cyan boundary tiles could be executed simultaneously, not starving the hardware resources.

Nested Block Cyclic Data Distributions. Porting the ExaGeoStat_PaRSEC as well as mixed-precision Cholesky proposed here is implemented with complete GPU support, i.e., distributed multi-GPUs, making it more prominent than most of those about mixed-precision in the related works [4], [21]–[23], [25], [52]. PaRSEC automatically handles asynchronous data transfers between hosts and devices to overlap data movement with computations, and also provides data locality scheduling policies to reduce communications and improve load balancing. However, when extending to GPU hardware accelerators in the context of the mixed-precision Cholesky factorization, load imbalance becomes so severe that lookahead and existing GPU-related optimizations may not be sufficient to mitigate the overheads. This load imbalance is indeed more exacerbated on GPU-based platforms than on homogeneous CPU systems. This is because GPUs, e.g., NVIDIA V100, provide customized hardware for performing much faster GEMM in HP than SP/DP. Currently, the proposed mixed-precision Cholesky factorization relies on the standard 2DBCDD to distribute the whole tiled matrix not only among MPI processes but also among all the GPUs dedicated to each parent MPI process. The non-critical tasks in the mixed-precision Cholesky factorization (mostly HGEMM tasks) are expedited and do



(a) Lookahead. (b) One-level DD. (c) Two-level DD.

Fig. 3: Runtime optimizations of a matrix with 9×9 tiles. Colors for tiles/arrows represent different precisions: DP in red, SP in blue, and HP in green. Different shapes represent different kernels: triangle POTRF, square GEMM, pentagon TRSM, and circle SYRK. (a) band_size_dp $= 4$ and band_size_sp $= 1$; (b, c) band_size_dp $= 2$, band_size_sp $= 2$, with process grid $P \times Q = 2 \times 2$ in cyan, the number of GPUs per MPI parent process $g = 4$, and GPU ID (0, 1, 2, 3) annotates each tile.

not slowdown the execution anymore, thanks to the high GPU computational power and the lookahead optimization. The performance bottleneck appears then in the tasks of the critical path that are not evenly distributed among GPUs within the parent MPI process. Fig. 3b showcases this load imbalance with a matrix of 9×9 tiles, $\texttt{band\_size\_dp} = 2$, $\texttt{band\_size\_sp} = 2$, and using a 2DBCDD with an MPI process grid $P \times Q = 2 \times 2$. We set the number of GPUs per process $g = 4$ and annotate each tile with GPU ID (0, 1, 2, 3) following also the traditional 2DBCDD. The figure reveals how only a single GPU out of four (i.e., GPU ID 3) executes the tasks (i.e., yellow boundary tiles) allocated to their MPI parent process. Therefore, a two-level 2DBCDD (MPI and GPU) backfires, and considering the performance discrepancy between multiple precision tasks observed when running on GPUs, it requires a new nested level of data distribution to maintain high occupancy on the devices. Fig. 3c demonstrates a new nested two-level data distribution using 2DBCDD for the MPI processes and 1DBCDD among the GPUs belonging to each MPI parent process. This nested 2DBCDD-1DBCDD now provides proper load balancing for tiles located in the critical path, operating in `DP` and `SP` on GPUs. For instance, most of GPUs of the parent MPI process ID #3 (located at the right bottom of a 2×2 process grid) are now busy operating in `DP` and `SP`, as highlighted with the yellow boundary tiles. The nested 2DBCDD-1DBCD contributes toward load balancing, while increasing the GPU hardware occupancy with tasks executed in the critical path.

7 Performance Results and Analysis

The correctness and performance of our mixed-precision approach are measured by synthetic and real datasets with different sizes and characteristics, on three HPC clusters with various kinds of architectures to evaluate the proposed approach's effectiveness:

- **`Shaheen II`** at KAUST: an Intel-based Cray XC40 system with $6,174$ compute nodes, each of which has two 16-core Intel Haswell CPUs at 2.30 GHz and 128 GB of memory.
- **`HAWK`** at HLRS: an AMD-based system with $5,632$ compute nodes, each of which has two 64-core AMD EPYC 7742 CPUs at 2.25 GHz and 256 GB of main memory.
- **`Summit`** at ORNL: an IBM-based system with $4,356$ compute nodes, each of which has two 22-core Power9 CPUs at 3.07 GHz and 256 GB of main memory, and each CPU is deployed with three NVIDIA Tesla V100 GPUs.

We use the term "`'a'D:'b'S:'c'H`" to represent the percentage of different precision formats per band regions, where $a = \texttt{band\_size\_dp}/NT * 100$ (NT is the number of tiles in a dimension), $b = \texttt{band\_size\_sp}/NT * 100$, and $a + b + c = 100$. For BLAS and LAPACK, we link against the vendor optimized libraries for each HPC cluster, i.e., Intel Math Kernel Library (MKL) on `Shaheen II`, AMD Optimizing CPU Libraries (AOCL) on `HAWK`, and IBM Engineering and Scientific Subroutine Library (ESSL) along with Compute Unified Device Architecture (CUDA) on `Summit`. The matrix is distributed by two-dimensional block cyclic data distribution (2DBCDD) with a process grid $P \times Q$ (as square as possible) where $P \leq Q$.

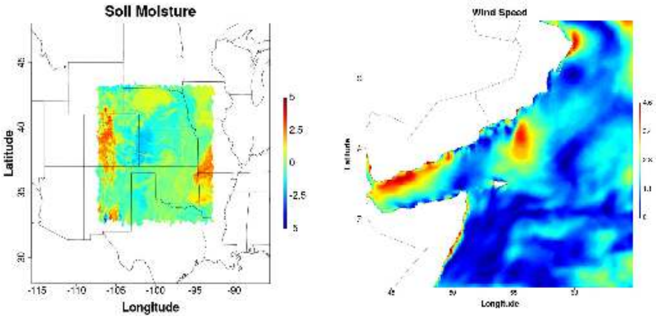


Fig. 4: **Left:** Soil moisture residuals at the topsoil of the Mississippi River basin. **Right**: Wind speed (m/s) in the Arabian Sea.

7.1 Synthetic Datasets

Synthetic datasets are a common way to validate the effectiveness of statistical modeling and prediction before applying them to real datasets. Herein, we use Monte Carlo simulations to show the impact of changing the precision of the covariance matrix using the proposed three-precision approach. Herein, we generate 40K synthetic datasets with different characteristics to mimic real cases. The generation process is performed using `ExaGeoStat_PaRSEC` software at irregular locations in a two-dimensional space with an unstructured covariance matrix, as suggested in [53]. To ensure that no two spatial locations are too adjacent, the data locations are generated using $n^{1/2}(r - 0.5 + X_{rl}, l - 0.5 + Y_{rl})$ for $r, l \in \{1, \ldots, n^{1/2}\}$, where n represents the number of locations, and X_{rl} and Y_{rl} are generated using uniform distribution on (−0.4, 0.4). Our Monte Carlo simulations strategy depends on generating 100 datasets with specific characteristics (i.e., correlation and smoothness) using a set of truth model parameters. All datasets are then modeled using mixed-precision variants to estimate the underlying model parameters for each dataset. The quality of each computation variant will depend on how close is the median of estimated parameters from the truth parameters.

7.2 Real Datasets

In this study, we consider two real datasets from two different regions of the world as follows.

The Soil Moisture Dataset. The U.S. soil moisture dataset is a high-resolution daily soil moisture data at the topsoil layer of the Mississippi River Basin (MRB) observed on January 1st, 2004. This dataset has been widely used to assess the quality of the spatial data modeling in literature [41], [54]–[56]. In [54], the original soil dataset has been updated by fitting a zero-mean Gaussian process model with a Matérn covariance function to the residuals to reduce the possibility of non-stationary data. The spatial resolution of the original dataset is of 0.0083 degrees, and the distance of one-degree difference in this region is approximately 87.5 km. The grid consists of $1830 \times 1329 = 2{,}432{,}070$ locations with $2{,}153{,}888$ measurements, as shown in Fig. 4. We have only considered a random subset of the dataset with size 1M in this paper although the whole dataset can be processed, as shown in previous work [41].

The Wind Speed Dataset. The wind speed dataset from the Middle-East region is a 2D dataset consisting of two variables, zonal wind component, U , and meridional wind component, V. A single univariate wind speed value (ws) can be computed from both components using, $ws = \sqrt{U^2 + V^2}$. Herein, we use a horizontal spatial resolution of 5 km gathered from a Weather Forecasting and Research (WRF) model simulation on the $[43°E, 65°E] \times [5°S, 24°N]$ region of the earth [57]. The target dataset has been restricted to the Arabian Sea, as shown in Fig. 4, with a total number of 116,100 locations. The choice of this particular subregion is motivated by the need to ensure that the measurements exhibit spatial isotropy, i.e., the cross-covariance depends only on the distance between 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 in the $116K$ dataset, this behavior can be expected. One more modification has been applied to the wind speed dataset to obtain a zero-mean random field: we remove a spatially varying mean using the longitudes and latitudes as covariates (we assume means are zero in our experiments).

7.3 Qualitative Analysis Using Synthetic Datasets

We use the Monte Carlo simulation to estimate the parameters of a powered exponential covariance model, with a set of truth parameters. We fix the variance parameter (θ_0) to 1.5 and we use two levels of smoothness (θ_2), 0.6 (rough field), and 1.5 (smooth field). We use the rough field with the three correlation lengths and give one example of smooth and strong correlated data. For the range parameter (θ_1), we compute it using Effective Ranges (ER) with weak, medium, and strong correlations. ER refers to the distance at which the marginal correlation drops to 0.05. We report our results as a set of boxplots to differentiate between different variants of mixed-precision computations when assessing estimation quality, number of iterations to converge, prediction accuracy, and prediction uncertainty.

Parameter Estimation.

In spatial statistics, the accuracy of the model parameters is critical to better understand and analyze the underlying spatial data. Fig. 5 presents the sensitivity of the parameter vector in presence of mixed-precision MLE computations (based on Cholesky factorization) for various correlation strengths and field characteristics. The figure presents the MLE boxplots of the estimated parameters for the synthetic datasets generated from a set of truth $\boldsymbol{\theta_t}$ vector. There are four columns, each labelled with the truth $\boldsymbol{\theta_t}$ vector that corresponds, from left to right, to rough field with weak correlations, to rough field with medium correlations, to rough field with strong correlations, and to smooth field with strong correlations. Each row provides the estimation accuracy of the variance θ_0, range θ_1, and smoothness θ_2 parameters based on the powered exponential matrix kernel, as defined by the initial truth $\boldsymbol{\theta_t}$ vector (i.e., red dotted lines). The first three columns in the given boxplots show that when correlation increases, the parameters vector becomes harder to estimate for configurations with lower precisions. Thus, one may experience accuracy loss with highly correlated data when using configurations with lower precisions. Moreover, when comparing the 3rd/4th columns with rough / smooth fields and strong correlations, smooth fields seem to require higher precision accuracy to properly estimates the model parameters, even with less correlated data (not shown in Fig. 5). Fig. 6 reports the impact of mixed-precision MLE computations on the total number of iterations performed during the learning phase. The single iterations of mixed-precision MLE are usually faster than the pure DP MLE. We observe that the mixed-precision MLE converges faster than DP MLE as the correlation strengths become stronger or in presence of smooth fields. This indicates that mixed-precision MLE has attained a local maximum that may or may not be close to the global maximum retrieved by the pure DP MLE. For instance, the mixed-precision MLE configurations with strong correlations and smooth field (4th column) do around four times less iterations than pure DP MLE but fail to precisely estimate θ_0 and θ_1, as shown in Fig. 5. However, some mixed-precision MLE configurations manage to successfully estimate θ_2.

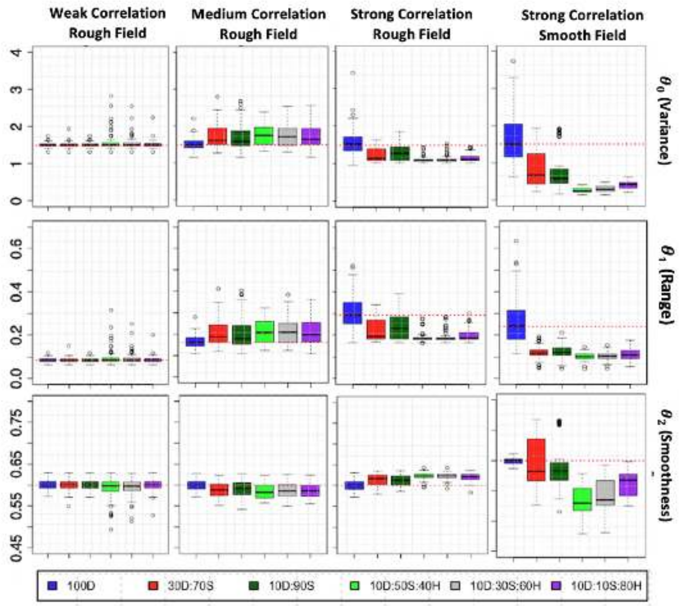


Fig. 5: Parameter estimation boxplots on 2D synthetic datasets with 40K locations using different mixed-precision MLE variants. "'a'D:'b'S:'c'H" represents the percentage of different precision formats, i.e., Double, Single, and Half, per band region.

Prediction Accuracy. Prediction accuracy in spatial statistics can be defined by two metrics, i.e., the Mean Square Prediction Error (MSPE) and the prediction uncer-

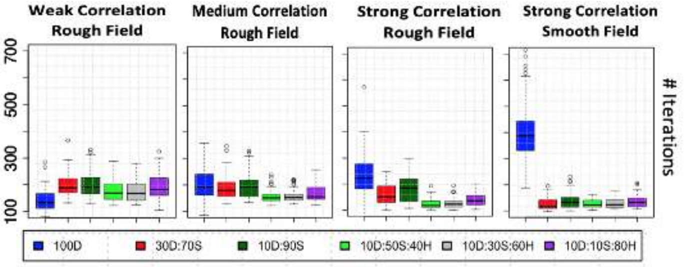
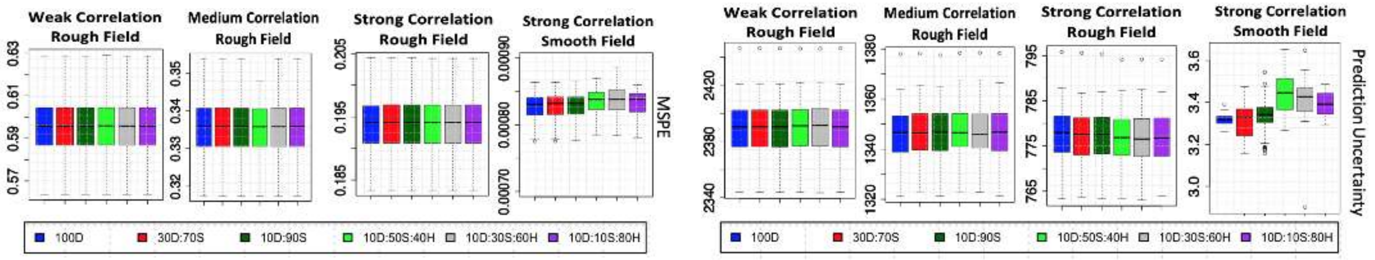


Fig. 6: Number of iterations on 40K 2D synthetic datasets using different mixed-precision MLE variants.



(a) Mean Square Prediction Errors (MSPEs).

(b) Prediction uncertainty.

Fig. 7: Prediction error (MSPE) and prediction uncertainty boxplots using 40K 2D synthetic datasets for 90% observed locations and 10% missing locations with different mixed-precision MLE variants.

TABLE 1: Qualitative assessment of the MLE based on the mixed-precision approach using 2D soil moisture dataset.

Variants	Variance (θ_0)	Range (θ_1)	Smoothness (θ_2)	Log-Likelihood (llh)	MSPE	Prediction Uncertainty	Iterations
100D	0.7223	0.0933	0.9983	-59740.65974	0.044926	4.734439e+03	180
10D:90S	0.7314	0.0953	0.9969	-59741.37532	0.044933	4.736149e+03	207
10D:30S:60H	0.7239	0.0936	0.9982	-59740.65200	0.044927	4.734435e+03	244
5D:5S:90H	0.7106	0.0927	0.9967	-59741.35348	0.044935	4.736572e+03	204
1D:99H	0.9330	0.1286	0.9863	-59867.53239	0.044980	4.750953e+03	159

TABLE 2: Qualitative assessment of the MLE based on the mixed-precision approach using 2D wind speed dataset.

Variants	Variance (θ_0)	Range (θ_1)	Smoothness (θ_2)	Log-Likelihood (llh)	MSPE	Prediction Uncertainty	Iterations
100D	0.8407	0.0751	1.9905	241480.9994	1.752914E-02	2.2855E+00	666
10D:90S	0.9924	0.1794	1.9757	239908.1004	1.766194E-02	2.9170E+00	91
10D:30S:60H	0.9761	0.1804	1.9576	232783.9932	1.765651E-02	5.2836E+00	94

tainty. We use 100 samples each with 40K locations to validate the prediction accuracy using synthetic datasets. Fig. 7 shows two boxplots assessing both MSPE and the prediction uncertainty. The MSPE boxplots do not show a significant difference with mixed-precision MLE variants, except for the smooth case (i.e., 4th column) in Fig. 7a. In general, it seems that the MSPE accuracy is slightly impacted by the mixed-precision approach. Fig. 7b shows the prediction uncertainty with different mixed-precision variants. With strong correlation and smooth field spatial data, the prediction uncertainty values of MP variants are higher than the DP variant's uncertainty values. However, if the data characteristic has exclusively one of those cases (i.e., strong correlation and smooth field), the prediction uncertainty difference compared to the high precision variant remains insignificant. Another observation from the figure that If comparing different mixed-precision variants to each other, the uncertainty values do not necessarily increase the uncertainty values with less precision. With the MP approximation, the process starts to be non-linear, and non-expected uncertainty values can pop up.

7.4 Qualitative Analysis Using Real Datasets

We estimate the underlying model parameters for the two aforementioned real datasets. For the 1M soil moisture dataset, Table 1 reports all the results corresponding to different mixed-precision MLE variants. The estimation of the model parameters (i.e., variance, range, and smoothness) for different configurations are close to the pure DP MLE, except for the 1D:99H variant. We tried several band sizes for each precision and kept only the ones showing some difference in parameters estimation, MSPE, or prediction uncertainty. Moreover, we observe from the estimated parameters that this dataset has medium correlated data with an average smooth field. This corroborates the analysis made with synthetic datasets that concludes on the effectiveness of the mixed-precision MLE for such data characteristics even with most of the computations performed in HP. The table also shows the sensitivity of the maximum log-likelihood values that correspond to the estimated parameters for each computation variant. The log-likelihood values also reflect the accuracy of the parameter estimation for each variant. Thus, all the mixed-precision MLE variants reach a similar log-likelihood value estimation after convergence, except for the 1D:99H configuration. The prediction accuracy (i.e., MSPE and prediction uncertainty) using the estimated parameters suggests that the mixed-precision MLE preserves it. In fact, such dataset characteristic seems to be resilient to accuracy loss even with the extreme 1D:99H variant.

For the wind speed dataset, Table 2 reports the parameters estimation and the prediction accuracy. This dataset comes from a highly smooth field (θ_2). Thus, the estimation of the model parameters is impacted starting from the first mixed-precision 10D:90S variant and further deteriorates with lower precision configurations. Indeed, the results show differences in parameter estimations, likelihood estimation, and prediction accuracy. For instance, the prediction uncertainty is even doubled 10D:30S:60H although MSPE is still acceptable. This qualitative assessment demonstrates how important it is to consider all these statistical metrics for obtaining an effective insight. These reported results match the trend seen for synthetic datasets boxplots in Fig. 5, where highly smooth data suffers when mixed-precision MLE is used. The two tables also show the total number of iterations to converge in each case. The reported results confirm the findings from the synthetic datasets in Fig. 6, where the number of iterations with the pure DP MLE are larger than the lower precision MLE variants in the case of

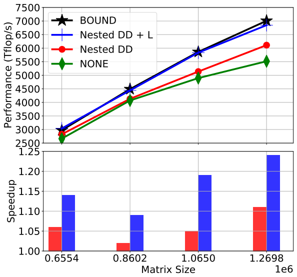


Fig. 8: Incremental effect of optimizations on Summit.

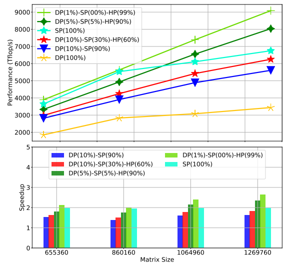
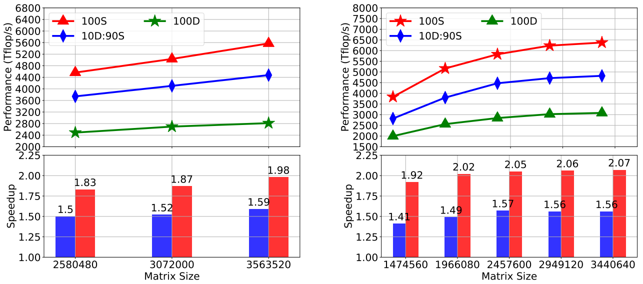


Fig. 9: Performance of mixed precisions on Summit.



(a) 1536 nodes on HAWK. (b) 4096 nodes Shaheen II.

Fig. 11: Performance of mixed DP/ SP.

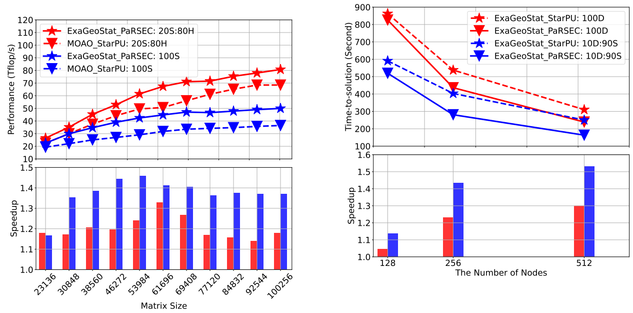
strong correlation and rough data (Table 1) and even larger for strong correlation and smooth data (Table 2).

7.5 Performance Impact of Optimizations

Two optimizations are proposed to guide the PaRSEC runtime system and efficiently tackle the load imbalance incurred by using mixed-precision Cholesky factorization. Fig. 8 shows the incremental impact of the lookahead (L) and nested data distribution (DD) optimizations on 128 nodes Summit using the mixed-precision Cholesky factorization variant 10D:10S:80H, which provides decent qualitative assessment for various data characteristics. In the figure, NONE means no optimization, and we also provide an upper bound (BOUND) for the performance, which executes the entire mixed-precision Cholesky, while disabling all HGEMMs. The mixed-precision Cholesky factorization achieves up to 10% performance improvement with the nested DD and up to 24% when both nested DD and lookahead are applied, reaching the upper bound. The resulting performance of 6.9 PFlop/s is about $1.6X$ compared to the DP Linpack performance on 128 Summit nodes.

7.6 Performance Comparisons

We compare the proposed mixed-precision Cholesky against two state-of-the-art mixed-precision applications on shared- and distributed-memory, i.e., a computational astronomy (i.e., MOAO_StarPU [58]) and a geostatistics applications (i.e., ExaGeoStat_StarPU [4]), with 20S:80H and 10D:90S mixed precision configurations, respectively. We only report on these two configurations since they maintain sufficient accuracy for both applications. Both applications are powered by StarPU runtime system, which does not provide inherent support for mixed-precision computations like PaRSEC. Therefore, the user is in charge of manually converting the tiles at the receiver side, which engenders higher volume of communication than PaRSEC. MOAO_StarPU mixes SP and HP, and targets a shared-memory system with four V100 GPUs; ExaGeoStat_StarPU deals with DP and SP computations on distributed-memory systems. Fig. 10 shows the detailed performance comparisons. When running both applications with the same precision, PaRSEC outperforms StarPU thanks to a native support for collective communications, while StarPU uses point-to-point communications. For 20S:80H, ExaGeoStat_PaRSEC outperforms MOAO_StarPU with up to $1.46X$ speedup, while achieving 80.0 TFlop/s on four V100 GPUs (Fig. 10a). For 10D:90S, ExaGeoStat_PaRSEC outperforms ExaGeoStat_StarPU on a distributed-memory system, and the advantage is more significant as the number of nodes increases with up to $1.53X$ speedup (Fig. 10b), thanks to a reduction in communication volume.



(a) MOAO_StarPU/PaRSEC comparison; higher is better.

(b) ExaGeoStat_StarPU/PaRSEC comparison; lower is better.

Fig. 10: Performance comparison against state-of-the-art (i.e., PaRSEC speedup compares to two different StarPU-based applications, MOAO_StarPU [58] and ExaGeoStat_StarPU [4]) using: (a), shared-memory: performance on four V100 GPUs; (b), distributed-memory: strong scalability with matrix size $640K \times 640K$ on Shaheen II.

7.7 Performance Evaluation at Scale

In this section, we evaluate the proposed mixed-precision Cholesky factorization at a large scale on the three before mentioned HPC clusters. HAWK and Shaheen II do not support HP, so Fig. 11 showcases only the mixed DP and SP performance for 100D, 10D:90S and 100S, along with the speedup of 100S and 10D:90S to 100D, on 1536 HAWK nodes and 4096 Shaheen II nodes. On Shaheen II, We report about $1.56X$ speedup from 10D:90S to 100D and $2.05X$ speedup from 100S to 100D when matrix size is larger than 2.4M. For the performance of 100D, it could achieve about 3.2 PFlop/s which is about 88% of the DP Linpack performance. Similarly on HAWK, we achieve performance of about 2.8 PFlop/s for 100D, while 4.5 PFlop/s for 10D:90S, and 5.6 PFlop/s for 100S with up to $1.59X$ speedup from 10D:90S to 100D and $1.98X$ speedup from 100S to 100D. On Summit, Fig. 9 shows the performance results with different combinations of DP, SP and HP, and their speedup relative to 100D on 128 nodes. The SP and DP curves show performance efficiency degradation after a certain matrix size due to memory swapping between host and device main memory. With the mixed-precision Cholesky factorization, we save memory footprint and we can achieve a significant efficiency and scalability as we increase the matrix size. In particular, we obtain up to 9.1 PFlop/s for `1D:99H`, i.e. $2.06X$ of the DP Linpack performance, that translates into up to $2.64X$ speedup against the `DP` Cholesky factorization.

All in all, these results show the efficiency and scalability `ExaGeoStat_PaRSEC` for mixed-precision Cholesky factorization while maintaining acceptable accuracy for geostatistical modeling and prediction.

8 Conclusion and Future Work

We demonstrate Maximum Likelihood Estimation (MLE) with a novel mixed three-precision Cholesky factorization powered by a dynamic runtime system on four major HPC systems. The resulting `ExaGeoStat_PaRSEC` framework exploits the mathematical structure of the covariance matrix by on-demand casting of precisions in computations and communications. This synergistic approach permits to achieve up to 9.1 (mixed) PFlop/s sustained performance by maximizing hardware occupancy using lookahead and nested data distributions. Application-expected accuracy is achieved thanks to a band region mechanism to set the precision arithmetics, tunable to preserve high productivity for users. In future work, we intend to leverage Tile Low-Rank approximations [48], [49] with mixed precisions to further reduce memory footprint and shorten time to solution.

References

[1] 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 T. on Parallel and Distributed Systems*, vol. 29, no. 12, pp. 2771–2784, 2018.

[2] E. Agullo, J. Demmel, J. Dongarra, B. Hadri, J. Kurzak, J. Langou, H. Ltaief, P. Luszczek, and S. Tomov, “Numerical Linear Algebra on Emerging Architectures: The PLASMA and MAGMA Projects,” *J. Phys.: Conf. Ser.*, vol. 180, no. 1, 2009.

[3] G. Morton, *A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing*. International Business Machines Company, New York, 1966.

[4] S. Abdulah, H. Ltaief, Y. Sun, M. G. Genton, and D. E. Keyes, “Geostatistical Modeling and Prediction Using Mixed Precision Tile Cholesky Factorization,” in *2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, 2019, pp. 152–162.

[5] D. E. Keyes, “The Arab World Prepares the Exascale Workforce,” *Communications of the ACM*, vol. 64, no. 4, pp. 82–87, 2021.

[6] C. G. Kaufman, M. J. Schervish, and D. W. Nychka, “Covariance Tapering for Likelihood-based Estimation in Large Spatial Datasets,” *J. of the American Statistical Association*, vol. 103, no. 484, pp. 1545–1555, 2008.

[7] S. Banerjee, A. E. Gelfand, A. O. Finley, and H. Sang, “Gaussian Predictive Process Models for Large Spatial Datasets,” *J. of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 4, pp. 825–848, 2008.

[8] Y. Sun, B. Li, and M. G. Genton, “Geostatistics for large datasets,” in *Advances and challenges in space-time modelling of natural events*. Springer, 2012, pp. 55–77.

[9] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, “Kalman filtering with intermittent observations,” *IEEE T. on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[10] J. M. Ver Hoef, N. Cressie, and R. P. Barry, “Flexible Spatial Models for Kriging and Cokriging using Moving Averages and The Fast Fourier Transform (FFT),” *J. of Computational and Graphical Statistics*, vol. 13, no. 2, pp. 265–282, 2004.

[11] Y.-J. Kim and C. Gu, “Smoothing Spline Gaussian Regression: More Scalable Computation via Efficient Approximation,” *J. of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 66, no. 2, pp. 337–356, 2004.

[12] T. Mary, “Block Low-Rank Multifrontal Solvers: Complexity, Performance, and Scalability,” Ph.D. dissertation, Paul Sabatier University, Toulouse, France, November 2017.

[13] D. E. Keyes, H. Ltaief, and G. Turkiyyah, “Hierarchical algorithms on hierarchical architectures,” *Philosophical Transactions of the Royal Society A*, vol. 378, no. 2166, p. 20190055, 2020.

[14] A. Aminfar, S. Ambikasaran, and E. Darve, “A Fast Block Low-rank Dense Solver with Applications to Finite-Element Matrices,” *J. of Computational Physics*, vol. 304, pp. 170–188, 2016.

[15] C. Geoga, M. Anitescu, and M. Stein, “Scalable Gaussian Process Computations using Hierarchical Matrices,” *J. of Computational and Graphical Statistics*, vol. 29, no. 2, pp. 227–237, 2020.

[16] P. Ghysels, X. S. Li, F.-H. Rouet, S. Williams, and A. Napov, “An Efficient Multicore Implementation of a Novel HSS-structured Multifrontal Solver using Randomized Sampling,” *SIAM J. on Scientific Computing*, vol. 38, no. 5, pp. S358–S384, 2016.

[17] S. Börm and S. Christophersen, “Approximation of Integral Operators by Green Quadrature and Nested Cross Approximation,” *Numerische Mathematik*, vol. 133, no. 3, pp. 409–442, 2016.

[18] W. Boukaram, G. Turkiyyah, and D. Keyes, “Hierarchical Matrix Operations on GPUs: Matrix-vector Multiplication and Compression,” *ACM T. on Mathematical Software (TOMS)*, vol. 45, no. 1, pp. 1–28, 2019.

[19] P. Düben, H. McNamara, and T. Palmer, “The Use of Imprecise Processing to Improve Accuracy in Weather & Climate Prediction,” *J. of Computational Physics*, vol. 271, pp. 2–18, 2014.

[20] T. Thornes, P. Düben, and T. Palmer, “On The Use of Scale-dependent Precision in Earth System Modelling,” *Quarterly J. of the Royal Meteorological Society*, vol. 143, no. 703, pp. 897–908, 2017.

[21] C. M. Maynard and D. N. Walters, “Mixed-precision Arithmetic in the ENDGame Dynamical Core of the Unified Model, A Numerical Weather Prediction and Climate Model Code,” *Computer Physics Communications*, vol. 244, pp. 69–75, 2019.

[22] A. Buttari, J. Dongarra, J. Langou, J. Langou, P. Luszczek, and J. Kurzak, “Mixed Precision Iterative Refinement Techniques for the Solution of Dense Linear Systems,” *Int. J. of High Performance Computing Applications*, vol. 21, no. 4, pp. 457–466, 2007.

[23] I. Yamazaki, M. F. Hoemmen, E. G. Boman, and J. Dongarra, “Communication-avoiding & Pipelined Krylov Solvers in Trilinos,” Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2019.

[24] E. Carson and N. J. Higham, “Accelerating The Solution of Linear Systems by Iterative Refinement in Three Precisions,” *SIAM J. on Scientific Computing*, vol. 40, no. 2, pp. A817–A847, 2018.

[25] A. Haidar, S. Tomov, J. Dongarra, and N. Higham, “Harnessing GPU Tensor Cores for Fast FP16 Arithmetic to Speed up Mixed-precision Iterative Refinement Solvers,” in *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2018, pp. 603–613.

[26] A. Duran, R. Ferrer, E. Ayguade, R. Badia, and J. Labarta, “A Proposal to Extend the OpenMP Tasking Model with Dependent Tasks,” *Parallel Programming*, vol. 37, no. 3, pp. 292–305, 2009.

[27] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier, “StarPU: A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures,” *Concurrency and Computation: Practice and Experience*, vol. 23, no. 2, pp. 187–198, 2011.

[28] OpenMP, “OpenMP 5.1 Complete Specifications,” 2020.

[29] M. Bauer, S. Treichler, E. Slaughter, and A. Aiken, “Legion: Expressing Locality and Independence with Logical Regions,” in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*. IEEE, 2012, pp. 1–11.

[30] T. Heller, H. Kaiser, and K. Iglberger, “Application of the ParalleX execution model to stencil-based problems,” *Computer Science - Research and Development*, vol. 28, no. 2-3, pp. 253–261, 2013.

[31] G. Bosilca, A. Bouteiller, A. Danalis, T. Herault, P. Lemarinier, and J. Dongarra, “DAGuE: A Generic Distributed DAG Engine for High Performance Computing,” *Parallel Computing*, vol. 38, no. 1-2, pp. 37–51, 2012.

[32] M. G. Genton, “Separable approximations of space-time covariance matrices,” *Environmetrics: The Official J. of the International Environmetrics Society*, vol. 18, no. 7, pp. 681–695, 2007.

[33] N. Cressie and C. K. Wikle, *Statistics for Spatio-temporal Data*. John Wiley & Sons, 2015.

[34] B. Matérn, “Spatial Variation, Volume 36 of Lecture Notes in Statistics,” 1986.

[35] J.-P. Chiles and P. Delfiner, *Geostatistics: Modeling Spatial Uncertainty*. John Wiley & Sons, 2009, vol. 497.

[36] S. Börm and J. Garcke, “Approximating Gaussian Processes with H^2-Matrices,” in *European Conference on Machine Learning*. Springer, 2007, pp. 42–53.

[37] J. Q. Shi and T. Choi, *Gaussian Process Regression Analysis for Functional Data*. CRC Press, 2011.

[38] S. Johnson, “The NLopt Nonlinear-OPTimization Package,” 2014.

[39] “The Chameleon project,” January 2021.

[40] “The HiCMA project,” January 2021.

[41] S. Abdulah, H. Ltaief, Y. Sun, M. G. Genton, and D. E. Keyes, “Parallel Approximation of the Maximum Likelihood Estimation for the Prediction of Large-scale Geostatistics Simulations,” in *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, 2018, pp. 98–108.

[42] G. Bosilca, A. Bouteiller, A. Danalis, M. Faverge, T. Hérault, and J. J. Dongarra, “PaRSEC: Exploiting Heterogeneity to Enhance Scalability,” *Computing in Science & Engineering*, vol. 15, no. 6, pp. 36–45, 2013.

[43] G. Bosilca, A. Bouteiller, A. Danalis, M. Faverge, T. Herault, and J. Dongarra, “PaRSEC: Exploiting Heterogeneity to Enhance Scalability,” *Computing in Science Engineering*, vol. 15, no. 6, pp. 36–45, Nov 2013.

[44] A. Danalis, G. Bosilca, A. Bouteiller, T. Herault, and J. Dongarra, “PTG: An Abstraction for Unhindered Parallelism,” in *2014 Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing*. IEEE, 2014, pp. 21–30.

[45] A. Danalis, H. Jagode, G. Bosilca, and J. Dongarra, “PaRSEC in Practice: Optimizing a Legacy Chemistry Application through Distributed Task-Based Execution,” in *2015 IEEE International Conference on Cluster Computing*, Sept 2015, pp. 304–313.

[46] H. Jagode, A. Danalis, G. Bosilca, and J. Dongarra, *Accelerating NWChem Coupled Cluster Through Dataflow-Based Execution*. Springer International Publishing, 2016, pp. 366–376.

[47] Q. Cao, Y. Pei, T. Herault, K. Akbudak, A. Mikhalev, G. Bosilca, H. Ltaief, D. Keyes, and J. Dongarra, “Performance Analysis of Tile Low-Rank Cholesky Factorization Using PaRSEC Instrumentation Tools,” in *IEEE/ACM International Workshop on Programming and Performance Visualization Tools (ProTools)*. IEEE, 2019, pp. 25–32.

[48] Q. Cao, Y. Pei, K. Akbudak, A. Mikhalev, G. Bosilca, H. Ltaief, D. Keyes, and J. Dongarra, “Extreme-Scale Task-Based Cholesky Factorization Toward Climate and Weather Prediction Applications,” in *Proceedings of the Platform for Advanced Scientific Computing Conference*, 2020, pp. 1–11.

[49] Q. Cao, Y. Pei, K. Akbudak, G. Bosilca, H. Ltaief, D. E. Keyes, and J. Dongarra, “Leveraging PaRSEC Runtime Support to Tackle Challenging 3D Data-Sparse Matrix Problems,” in *International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2021.

[50] J. J. Dongarra, “Performance of Various Computers using Standard Linear Equations Software,” *ACM SIGARCH Computer Architecture News*, vol. 20, no. 3, pp. 22–44, 1992.

[51] E. Agullo, J. Demmel, J. Dongarra, B. Hadri, J. Kurzak, J. Langou, H. Ltaief, P. Luszczek, and S. Tomov, “Numerical Linear Algebra on Emerging Architectures: The PLASMA and MAGMA projects,” in *J. of Physics: Conference Series*, vol. 180, no. 1. IOP Pub., 2009.

[52] A. Haidar, S. Tomov, J. Dongarra, and N. Higham, “Harnessing GPU Tensor Cores for Fast FP16 Arithmetic to Speed up Mixed-precision Iterative Refinement Solvers,” in *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2018, pp. 603–613.

[53] Y. Sun and M. L. Stein, “Statistically and computationally efficient estimating equations for large spatial datasets,” *J. of Computational and Graphical Statistics*, vol. 25, no. 1, pp. 187–208, 2016.

[54] H. Huang and Y. Sun, “Hierarchical Low Rank Approximation of Likelihoods for Large Spatial Datasets,” *J. of Computational and Graphical Statistics*, vol. 27, no. 1, pp. 110–118, 2018.

[55] Y. Hong, S. Abdulah, M. G. Genton, and Y. Sun, “Efficiency Assessment of Approximated Spatial Predictions for Large Datasets,” *Spatial Statistics*, 2021.

[56] N. W. Chaney, P. Metcalfe, and E. F. Wood, “HydroBlocks: A Field-scale Resolving Land Surface Model for Application Over Continental Extents,” *Hydrological Processes*, vol. 30, no. 20, pp. 3543–3559, 2016.

[57] C. M. A. Yip, “Statistical Characteristics and Mapping of Near-surface and Elevated Wind Resources in the Middle East,” Ph.D. dissertation, King Abdullah University of Science and Technology (KAUST), 2018.

[58] N. Doucet, H. Ltaief, D. Gratadour, and D. Keyes, “Mixed-Precision Tomographic Reconstructor Computations on Hardware Accelerators,” in *2019 IEEE/ACM 9th Workshop on Irregular Applications: Architectures and Algorithms (IA3)*, 2019, pp. 31–38.

Sameh Abdulah is a research scientist with the Extreme Computing Research Center, King Abdullah University of Science and Technology, Saudi Arabia. He received the MS and PhD degrees from Ohio State University, Columbus, in 2014 and 2016, respectively. His work is centered around high performance computing (HPC) applications, big data, large spatial datasets, parallel statistical applications, algorithm-based fault tolerance, and machine learning and data mining algorithms.

Qinglei Cao is a PhD student in the Innovative Computing Laboratory, University of Tennessee. He received his BS in information and computational science from Hunan University and MS in computer application technology from Ocean University of China. Also, he worked at National University of Defense Technology as a software engineer. His research interests evolve distributed/parallel computing, task-based runtime system and linear algebra.

Yu Pei is a computer science PhD student in the Innovative Computing Laboratory, University of Tennessee, Knoxville. His research interests include programming interfaces of distributed task-based runtime systems, efficient implementation of numerical linear algebra algorithms and their applications. He received his MS degree in statistics from UC Davis in 2015.

George Bosilca is a research director and adjunct assistant professor in the Innovative Computing Laboratory, University of Tennessee, Knoxville. His research interests evolve around the concepts of distributed algorithms, parallel programming paradigms, and software resilience, from both a theoretical and practical perspective.

Jack Dongarra holds an appointment with the University of Tennessee, Oak Ridge National Laboratory, and the University of Manchester. He specializes in numerical algorithms in linear algebra, parallel computing, use of advanced-computer architectures, programming methodology, and tools for parallel computers. He is a fellow of the AAAS, ACM, IEEE, and SIAM and a foreign member of the Russian Academy of Science and a member of the US National Academy of Engineering.

Marc Genton received the PhD degree in statistics from the Swiss Federal Institute of Technology (EPFL), Lausanne. He is a distinguished professor of statistics at KAUST. He is a fellow of the ASA, of the IMS, and the AAAS, and is an elected member of the ISI. His 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 directs the Extreme Computing Research Center at KAUST. He works at the interface between parallel computing and the numerical analysis of PDEs with a focus on scalable implicit solvers, such as the Newton-Krylov-Schwarz (NKS) and the Additive Schwarz Preconditioned Inexact Newton (ASPIN) methods, which he co-developed. He received a BSE in aerospace and mechanical sciences from Princeton and a PhD in applied mathematics from Harvard. He is a fellow of SIAM, AMS, and AAAS.

Hatem Ltaief is a principal research scientist with the Extreme Computing Research Center, 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.

Ying Sun received the PhD degree in statistics from Texas A&M University in 2011. She is an associate professor of statistics with the King Abdullah University of Science and Technology (KAUST) in Saudi Arabia. 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.