Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Simulation-based Evaluation of Numerical Algorithms forEnhancing Geostatistical Model Computation using MCMC: A Case Study in Disease Mapping

Romuald BEH MBA

behmbaroms@email.com

Laboratoire de Biomathématiques et d'Estimations Forestières, Université d'Abomey Calavi, 04 B.P 1525

Research Article

Keywords: Conjugate Gradient, Cholesky decomposition, Gibbs Sampling: sparse matrix, Markov Chain Monte Carlo

Posted Date: April 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9248586/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Simulation-based Evaluation of Numerical Algorithms for Enhancing Geostatistical Model Computation using MCMC: A Case Study in Disease Mapping

Romuald BEH MBA[1], **Kolawolé V. SALAKO** [1], **Romain L. GLÈLÈ KAKAÏ**[1]

[1]Laboratoire de Biomathématiques et d'Estimations Forestières, Université d'Abomey Calavi, 04 B.P 1525

Cotonou, Benin

Abstract

Background: Bayesian geostatistical models incorporate spatial dependence through covariance matrices of location-specific random effects. In large spatial datasets, repeated manipulation of these covariance matrices represents a major computational bottleneck. Sparse matrix representations offer a potential strategy for improving computational feasibility, although their impact may vary across numerical algorithms.

Methods: We evaluated the computational behaviour of three numerical strategies Cholesky decomposition, Conjugate Gradient (CG), and Gibbs-based inversion under varying covariance matrix sizes (300, 600, and 1,000 locations) and sparsity levels (20%, 50%, and 100% zero-valued off-diagonal elements). Performance was assessed in terms of computational time and approximation accuracy relative to a deterministic reference solution.

Results: The results indicate that computational efficiency is strongly influenced by both matrix sparsity and algorithmic structure Cholesky decomposition serves as a deterministic benchmark, providing exact solutions for symmetric positive definite matrices with machine-level precision. The Conjugate Gradient method demonstrates rapid deterministic convergence, attaining near-zero Root Mean Square Error (RMSE) within a small fraction of the computational time required by direct factorisation, with convergence speed influenced primarily by matrix conditioning. In contrast, Gibbs sampling exhibits stochastic convergence, characterised by gradual RMSE reduction and transient variability due to Monte Carlo fluctuations. Extended iterations reduce sampling variability and stabilise accuracy, consistent with asymptotic sampling theory. Our results highlight the fundamental distinction between deterministic solvers, which optimise numerical precision and efficiency, and stochastic simulation-based approaches, which prioritise posterior uncertainty quantification.

Conclusion: Sparse covariance structures substantially improve computational feasibility in large-scale Bayesian geostatistical modelling. This comparative assessment provides practical guidance for solver selection in Gaussian geostatistical modelling: CG is the optimal strategy for computationally efficient deterministic recovery, whereas Gibbs sampling retains methodological relevance for fully Bayesian inference.

Keywords: Conjugate Gradient; Cholesky decomposition; Gibbs Sampling: sparse matrix; Markov Chain Monte Carlo

∗ **Corresponding author.** *E-mail address: behmbaroms@email.com*

1. Background

Geostatistical models for disease mapping typically involve the use of Bayesian hierarchical models, where the spatial and temporal structure of the data are incorporated through the use of covariates, random effects, and spatiotemporal correlation structures. The estimation of these models requires the use of Markov Chain Monte Carlo (MCMC) algorithms, which can be computationally intensive, especially when dealing with large datasets. Models for geostatistical data are embedded within the framework of generalized linear mixed models (GLMM). Geographical dependence is introduced via the covariance structure of location-specific random effects which is specified to be a parametric function of the distances between locations. Under the assumption of stationarity, the covariance matrix determines the variogram and the GLMM is also known as variogram model. Maximum likelihood-based estimation has major shortcomings. Asymptotic inference is not uniquely defined [1] and the competing approaches may lead to different results [2]. When the aim of modelling is kriging, that is prediction at un-sampled locations, the parameter uncertainty in maximum likelihood estimation is not fully accounted for and the standard error of predicted values underestimates the true variability. In that case [3] formulated the variogram model as a Bayesian hierarchical model and provided full Bayesian inference using MCMC estimation. However, MCMC estimation is hampered by the repeated inversions of the covariance matrix of the random effects which for large number of locations can be infeasible within practical time constraints. To overcome the computational problems that arise from the inversion of large covariance matrices, [4] suggested a non-iterative estimation procedure implemented via Sampling-Importance Resampling (SIR) [5].

In contrast to MCMC-based sampling, SIR is not generally applicable and needs to be tailored to every spatial model and dataset [1]. [6] adopted [7] and proposed speeding up the computation of MCMC by jointly updating the whole vector of random effects using the Metropolis-Langevin algorithm instead of updating each random effect separately within an MCMC iteration. There are also a number of approaches aiming to improve MCMC in general and not specifically for fitting variogram models. There are methods which accelerate the convergence time [8] or reduce correlations between the parameters and autocorrelation in the samples drawn from the Markov chain by overrelaxation or centering [9]. Other sampling schemes reduce the number of rejections made by the Metropolis-Hastings algorithm within Gibbs sampling [10]; [8]). None of these techniques overcome or improve the process of matrix inversion, but they make it possible to sample significantly shorter chains to reach the same accuracy.

In a series of papers [10] and [11] suggested methods to accelerate conditional autoregressive spatial models for a real data using the sparseness of the proximity matrix. In a real data setting, a proximity matrix with zero values for all areas which do not share a common border can be easily defined. This can also be adapted for variogram modelling by choosing appropriate covariance matrices. There are valid spatial covariance definitions with zero value for all covariances defined at lag distances

higher than a specific parameter, called the range [12]. The sparseness is not *a priori* defined for these models but it depends on the range parameter. [13] demonstrated the computational advantages in solving the kriging equations using sparse spatial covariance matrices.

A sparse covariance matrix does not necessarily facilitate matrix inversion, since the factor of a sparse matrix may no longer be sparse. In most cases, incomplete factorization [14] can also ensure sparsity of the matrix factor. Incomplete factorization is primarily intended for use as a pre-conditioner in an iterative approach and it is well suited in the Gibbs sampling framework. This is because iterative methods can be computationally more efficient than non-iterative ones when starting values are good. Within the Gibbs sampling framework good starting values for the incomplete factorization can be obtained from the previous Gibbs sampling iteration. The assessment of numerical algorithms to improve the computation of geostatistical models using MCMC is an important area of research in disease mapping [15]. The goal of this study is to assess algorithms that can efficiently and accurately estimate geostatistical models parameters for disease mapping, considering spatial and temporal dependence of the data. The computational speed of the suggested algorithms is assessed on simulated datasets of different size and sparsity (20%, 50%, 100%). Sparsity is defined as the proportion of zero-valued off-diagonal elements in the spatial correlation matrix. The sparsity level labelled as 100% represents the limiting case in which all off-diagonal correlation coefficients are set to zero. Under this configuration, the covariance matrix reduces to a diagonal structure, reflecting the absence of spatial dependence while preserving diagonal variance components. This formulation ensures that the covariance matrix remains symmetric positive-definite and computationally well-defined.

The sparse structure of geostatistical covariance matrices makes the computation via sparse linear system solvers efficient in MCMC estimation. Sparsity can significantly affect the accuracy of models, especially in scenarios like machine learning, statistics, and Bayesian inference. Specifically, we compared the computational efficiency of Conjugate Gradient method (CG); Cholesky Decomposition; and Gibbs-sampling inversion algorithms to evaluate their performance in the context of geostatistical models and MCMC methods for disease mapping. The selection of Cholesky decomposition, Conjugate Gradient (CG), and Gibbs sampling was motivated by their complementary computational roles and their widespread use in large-scale Bayesian geostatistical modeling[16]. Cholesky decomposition represents the classical exact approach for solving linear systems involving symmetric positive-definite covariance matrices and serves as a natural accuracy benchmark [17]. However, its computational and memory costs scale poorly with matrix size, particularly in dense settings [18]. In contrast, the Conjugate Gradient (CG) method provides a deterministic iterative alternative specifically designed for large symmetric positive-definite systems. CG is well known for its efficiency when applied to sparse matrices, making it a standard tool in modern numerical linear algebra and spatial statistics [19]. Gibbs sampling was included as a stochastic iterative approach compatible with the MCMC framework. Rather than computing explicit

matrix inverses, Gibbs-based solvers generate approximate solutions through conditional sampling, which can be computationally attractive when direct factorization becomes prohibitive[20]. Together, these three methods capture the principal computational strategies encountered in practice: exact factorization (Cholesky), deterministic iterative optimization (CG), and stochastic iterative approximation (Gibbs sampling). This choice therefore enables a conceptually meaningful comparison of accuracy, scalability, and computational efficiency across fundamentally different yet practically relevant solution paradigms.

The remainder of the paper is organized as follows. Section 2 presents the statistical methodology, including the Bayesian formulation, Markov chain Monte Carlo computations, and the fast matrix inversions algorithms. Section 3 describes the simulation design Section 4 presents the simulation results section 5 discusses the implications, limitations, and potential extensions of the proposed approach in the conclusion.

2. Material and Method

2.1. Bayesian formulation

To describe the basic model, let $Y(s) = \left(Y(s_1), Y(s_2), \cdots, Y(s_m)\right)^t$ be the response data observed at the set of locations $s = (s_1, s_2, \cdots, s_m)^t$ where $s_i \in D \subset R^2, i = 1, \cdots, m$ and let $\mathbf{X}(s_i)$ be the vector of covariates j (j = 1,2,⋯ ,n_i) associated with location S_i. Following the GLMM framework of [21], we introduce location-specific random effects $\phi(s) = (\phi(s_1), \ldots, \phi(s_m))^T$. We assume that the response variable $Y(s_i)$ is conditionally independent given the random effects, with conditional expectation defined as $E[Y(s_i) \mid \phi(s_i)] = \mu(s_i)$. Covariates and spatial random effects are modelled on $g(\mu(s_i))$ where $g(.)$ is the link-function and $g(\mu(s_i)) = X^t(s_i)\beta + \phi_i$. For simplicity, we drop s in the subscripts and write Y , Y_i, X_i, ϕ and ϕ_i instead of $Y(s)$, $Y(s_i)$, $X(s_i)$, $\phi(s)$ and $\phi(s_i)$, respectively and introduce $X = (X_1^t, \cdots, X_m^t)^t$. Spatial dependence is captured by the location-specific random effects. It is assumed that they model a latent stationary and isotropic Gaussian spatial process over the study region, D, such that $\pi(\phi|\sigma^2, \delta, \tau^2) \equiv N(0, \Sigma)$, where Σ_{ij} is a parametric function of the distance between the corresponding locations s_i and s_j, that is $\Sigma = \sigma^2\varrho(\| s_i - s_j \|_2; \delta) + \tau^2 1_{i=j}$ where $\| . \|_2$ denotes the Euclidean distance. $\varrho(.)$ corresponds to a valid $_{ij}$

correlation function, σ^2 measures the spatial variance, ϱ quantifies the rate of correlation decay and τ captures non-spatial variation. The exponential form $\varrho(\| s_i - s_j \|_2; \delta) = \exp(\| s_i - s_j \|_2 / \delta))$ is the most commonly used correlation function. Alternative forms are proposed by [12] Under the assumption of stationarity, Σ determines the variogram and the parameters σ^2, ϱ and τ^2 are directly related to the sill, range and nugget parameters of the variogram [22].

A Bayesian formulation of the variogram model requires specification of prior distributions for the parameters, β, σ^2, δ and τ^2. Typically, we assume a non-informative distribution, such as a uniform or a normal with large variances, for the covariate coefficients and inverse Gamma distributions for σ^2, δ and τ^2. Inference is based on the posterior distribution

$$p(\sigma^2, \delta, \tau^2|Y) \propto L(Y|\beta, \emptyset)\pi(\emptyset|\sigma^2, \delta, \tau^2)\pi(\beta)\pi(\sigma^2)\pi(\delta)\pi(\tau^2), \quad (1)$$

where L ($\mathbf{Y}|\beta, \phi$) is the likelihood and $\pi(.)$ denotes the prior distributions of the parameters. This model is highly parameterized especially for large number of locations and the model fit is only feasible via Markov chain Monte Carlo.

2.1.1. Markov chain Monte Carlo computations

The Gibbs sampling algorithm is the most common Markov chain-based simulation algorithm [23]. Its standard implementation requires sampling from the one-dimensional conditional distributions of all parameters, iteratively until convergence. Variations of the algorithm include sampling from multivariate conditional distributions of block of parameters. However, these distributions have rarely known forms and sampling is not straightforward. In the variogram model specified above, the one-dimensional conditional distributions of the ϕ_i, $i = 1, \cdots, m$, parameters have the following form:

$$p(\emptyset_i|\emptyset_{-i}, \beta, \sigma^2, \delta, \tau^2, Y) \propto (\emptyset_{-i}|\emptyset_{-i}, \sigma^2, \delta, \tau^2) \cdot L(Y_i|\emptyset_i, \beta). \quad (2)$$

The first term on the right-hand side is a univariate normal with mean $E(\phi_i|\phi_{-i}, \sigma^2, \delta, \tau^2) = \Sigma_{-i,i}\Sigma_{-i,-i}^{-1}\phi_{-i}$
and variance $Var(\phi_i|\phi_{-i}, \sigma^2, \delta, \tau^2) = \sigma^2 + \tau^2 - \Sigma_{-i,i}\Sigma_{-i,-i}^{-1}\Sigma_{i,i}$ where
$\Sigma = E(\phi\phi^t), \phi_{-i} = (\phi_1, \cdots, \phi_{i-1}, \phi_{i+1}, \cdots, \phi_m)^t, \Sigma_{-i,-i} = E(\phi_{-i}\phi_{-i}^t)$ and

$$\Sigma_{-i,i} = \Sigma_{i,-i} = E(\phi_{-i}\phi_{-i}^t).$$

The Metropolis-Hastings algorithm can be used to simulate from the above conditional distribution. This requires computer intensive matrix inversions to calculate the term $c_i^t = \Sigma_{i,-i}\Sigma_{-i,-i}^{-1}$. The $(m-1) \times (m-1)$ matrix $\Sigma_{-i,-i}$ is inverted m times during one iteration of the Gibbs Sampler. The inversion is calculated via the solution of the linear system $\Sigma_{-i,-i} \bullet c_i = \Sigma_{-i,i}$. It is computed commonly by first decomposing $\Sigma_{-i,-i}$ into its Cholesky factors $\Sigma_{-i,-i} = Q_iQ^t_i$ and then using forward and backward substitution in $Q_ix_i = \Sigma_{-i,i}$ and $Q^t_i \bullet c_i = x_i$. As an alternative to updating ϕ component wise, we could update the whole vector as a block using Metropolis-Hastings. [1] discussed the advantages of block updating in the parameters, however it is difficult to find a good multivariate proposal distribution which is easy to simulate from and resembles the conditional distribution of ϕ. [24] proposes the Hamiltonian dynamics and provides a general method for constructing multivariate proposal distributions. A special case of the Hamilton method is the Langevin diffusion Metropolis-Hastings

algorithm which has been applied by [25] for updating the spatial random effects of variogram models. The Langevin algorithm constructs a multivariate Gaussian proposal distribution with mean $\mu_{\phi(t)}$ and variance $a^2.I_m$ and

Simulates $\phi^{(t)} \sim N\left(\mu_{\phi^{(t)}}, a^2.I_m\right)$ at a Gibbs iteration t where,

$$\mu_{\phi^{(t)}} = \phi^{(t-1)} + \frac{a}{2}\nabla log p\left(\phi^{(t-1)}; Y\right), \tag{3}$$

and a is a constant which should be adjusted according to the acceptance rate. $p(\phi; Y)$ is the full conditional distribution of ϕ which has the form:

$$p(\emptyset; Y) = p(\emptyset|\beta, \sigma^2, \delta, \tau^2; Y) \propto \pi(\phi|\sigma^2, \delta, \tau^2).L(Y|\phi, \beta). \tag{4}$$

At iteration t the $\phi^{(t-1)}$ is updated by $\phi^{(t)}$ with probability

$$min\left\{\frac{p(\phi^{(t)};Y)\exp\left(-\frac{1}{2a^2}\sum_{i=1}^{m}\left(\phi_i^{(t-1)}-\mu_{\phi^{(t)}}\right)^2\right)}{p(\phi^{(t-1)};Y)\exp\left(-\frac{1}{2a^2}\sum_{i=1}^{m}\left(\phi_i^{(t)}-\mu_{\phi^{(t-1)}}\right)^2\right)}, 1\right\}. \tag{5}$$

Computation of $p(\phi;Y)$ and in particular of $\pi(\phi|\sigma^2,\delta,\tau^2)$ requires inversion of the $m \times m$ matrix Σ. The conditional distribution of σ^2, δ and τ^2 includes also terms with Σ^{-1}. It is proportional to

$$det(\sum)^{-\frac{1}{2}} exp\left(-\frac{1}{2}\emptyset^t\Sigma^{-1}\right)\pi(\sigma^2)\pi(\delta)\pi(\tau^2) \tag{6}$$

2.2. Algorithms for fast matrix inversions

The empirical assessment can be performed by comparing the accuracy and computational time of different matrix inversion algorithms on a set of simulated geostatistical data. The data can be generated with different levels of sparsity in the spatial covariance matrix and different numbers of locations, and can have normal or Poisson response variables, as well as multiple covariates. The performance of these can be evaluated by comparing the results to the true inverse of the matrix and by measuring the computational time required to obtain the inverse. In addition, the impact of the choice of matrix inversion on the quality of the results from MCMC methods can be evaluated by comparing the posterior distributions obtained with different algorithms. This can be done by computing summary statistics, such as the mean and standard deviation, for each posterior distribution, and by visually comparing the distributions using histograms or kernel density plots [26]. By conducting an empirical assessment for fast matrix inversions, researchers can gain insights into the trade-offs between accuracy and computational efficiency, and determine the best methods for different types of geostatistical data and models. The most computationally intensive part in the implementation of MCMC is the calculation of the quantities $\Sigma_{i,-i}\Sigma_{-i,-i}^{-i}, i = 1, \cdots, m$ and $\phi^t\Sigma^{-i}\phi$ which involve the inversion of the $(m-1) \times (m-1)$ and $m \times m$ matrices $\Sigma_{-i,-l}$ and Σ respectively. The

numerical algorithm which has been used for speeding the inversion of large matrices include the Conjugate Gradient (CG), Cholesky Decomposition (CD) and Gibbs Sample (GS).

2.2.1. Conjugate Gradient (CG) Method

The Conjugate Gradient (CG) method is widely used in solving large linear systems of equations, especially those involving sparse matrices. In the geostatistical models, it is often applied to kriging, covariance estimation, and spatial prediction problems.it is an iterative algorithm for solving systems of linear equations, particularly those with symmetric positive-definite matrices, which is often the case with the covariance matrices [27]. CG is effective for systems of the form $Kx = b$ where x in an unknown vector, b in a known vector, and K in a known, square, symmetric, positive-definite (or positive-indefinite) matrix.

The Conjugate Gradient (CG) method is an iterative algorithm for solving linear systems of the form:

K w = z,

where K is a symmetric positive-definite matrix, z is a known vector, and w is the unknown vector to be estimated.

Input: Matrix K, vector z, tolerance ε

Output: Approximate solution w

Step 1 (Initialization):

Choose an initial guess w^(0).

Compute the initial residual: r^(0) = z − K w^(0).

Set the initial search direction: p^(0) = r^(0).

Step 2 (Iteration):

For k = 0, 1, 2, … repeat:

Compute the step size: α_k = (r^(k)^T r^(k)) / (p^(k)^T K p^(k)).

Update the solution: w^(k+1) = w^(k) + α_k p^(k).

Update the residual: r^(k+1) = r^(k) − α_k K p^(k).

Check convergence: Stop if ||r^(k+1)|| < ε.

Compute the conjugacy coefficient: β_k = (r^(k+1)^T r^(k+1)) / (r^(k)^T r^(k)).

Update the search direction: p^(k+1) = r^(k+1) + β_k p^(k).

Step 3 (Termination):

The iterations continue until the stopping criterion is satisfied, yielding the approximate solution w.

In kriging, solving the kriging weights w often involves inverting a large covariance matrix K. Instead of directly inverting **K**, **CG** can iteratively find **w**, improving computational efficiency. For example, in ordinary kriging, the system to be solved is: Let K, an $n \times n$ matrix, and x and b are vectors that is, $n \times 1$ matrices. Written out fully, Equation

$$\begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,n} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n,1} & K_{n,2} & \cdots & K_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (7)$$

The *inner* product of two vectors is written $x^T y$, and represents the scalar sum $\sum_{i=1}^{n} x_i y_i$. Note that $x^T y = y^T x$. If *x* and *y* are orthogonal, then $x^T y = 0$. In general, expressions that reduce to 1 × 1 matrices, such as $x^T y$ and $x^T x$, are treated as scalar values. This is an augmented system, but the **CG** method can still be applied effectively to find the weights **w** by iterating over the main covariance matrix **K**.

2.2.2. Cholesky Decomposition

Cholesky Decomposition is used to factorize the covariance matrix into a lower triangular matrix and its transpose. This factorization simplifies the computation of the log-likelihood and the posterior distribution. This approach is both numerically stable and efficient, allowing Gaussian Process (GPs) to be applied to moderately large datasets, where exact inversion would be computationally prohibitive. Cholesky decomposition enables scalable Gaussian Process models, making them practical for real-world applications such as time-series forecasting, spatial data analysis, and hyperparameter optimization. Cholesky decomposition is a method used to solve linear systems, particularly for symmetric, positive-definite matrices [28]. The Cholesky decomposition of a matrix A result in a lower triangular matrix L such that: $A = LL^T$ where L^T is the transpose of **L**. To compute the inverse of A using Cholesky decomposition, we follow these steps:

Let A be a symmetric positive-definite matrix. The Cholesky decomposition factorizes A as:
$A = L L^T$,
where L is a lower triangular matrix.
Instead of computing the inverse A^{-1} directly, the inverse can be obtained by solving triangular systems.
Input: Symmetric positive-definite matrix A
Output: Matrix inverse A^{-1}
Step 1 (Factorization):
Compute the Cholesky decomposition: $A = L L^T$.
Step 2 (Forward Substitution):
Solve the triangular system: L Y = I, where I is the identity matrix.
This yields: $Y = L^{-1}$.
Step 3 (Backward Substitution):
Solve: $L^T X = Y$.
Step 4 (Result):
The inverse matrix is obtained as: $A^{-1} = (L^{-1})^T L^{-1}$.

Cholesky decomposition is commonly used in statistics, particularly in multivariate analysis for generating samples from multivariate normal distributions, Cholesky decomposition of the covariance matrix allows the transformation of uncorrelated random variables into correlated ones. In Linear Mixed Models (LMMs), Cholesky decomposition is used to handle the random effects covariance matrix in linear mixed models, simplifying parameter estimation. In Bayesian statistics, Cholesky decomposition is used to efficiently sample from posterior distributions, which often involve symmetric positive-definite covariance matrices.

2.2.3. Gibbs-sampling inversion

The inversion of the covariance matrices which are involved in the implementation of Gibbs sampling can be obtained by iterative solvers. The advantage of this approach is that iterative methods require less memory and arithmetic operations than direct methods especially when starting values are good. The Gibbs sampling framework is suitable to the iterative solver approach as the initial values for the solver can be chosen to be the estimates of the previous Gibbs iteration. An alternative iterative linear solver approach was presented by [29] who inverts large matrices using Gibbs sampling.

Let Γ be the inverse of Σ,$\boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{m-1})^t$ and $\boldsymbol{x}_{-k} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{k-1}, \boldsymbol{x}_{k+1}, \cdots, \boldsymbol{x}_{m-1})^t$ such as $\boldsymbol{x} \sim N(0, \Gamma)$ where $\Gamma_{kj} = E(x_k x_j)$, $\Gamma_{k,-k} = E(x_k x_{-k})$ and $\Gamma_{-k,-k} = E(x_{-k} x_{-k}^t)$. It follows that the conditional distribution$[x_k | x_{-k}]$ are normal distribution such that

$$[(x_k | x_{-k})] \sim N\left(\Gamma_{k,-k}\Gamma_{-k,-k}^{-1} x_{-k}, \Gamma_{k,k}\Gamma_{k,-k}\Gamma_{-k,-k}^{-1}\Gamma_{-k,k}\right).$$

Gibbs sampling can be applied to simulate Γ where $\Gamma_{k,j}$ is estimated by $s^{-1}\sum_{t=1}^{s} x_k^{(t)} x_j^{(t)}$ and $x^{(t)}$ are samples obtained after convergence of the algorithm and is the size of the sample drawn from the posterior. Using

Gibbs sampling outputs, we can solve the linear system $\Sigma_{-i,-i} c_i = \Sigma_{i,i}$ by taking

$$\hat{c}_{i,k} = S^{-1}\sum_{t=1}^{s} x_k^{(t)} \sum_{j=1}^{m-1} x_j^{(t)} u_{ij}, \tag{9}$$

where $\hat{c}_{i,k}$ is an estimate of the *k-th* element of c_i and $u_{i,j}$ is the *j-th* element of $\Sigma_{-i,i}$. [29] proposes ways of reducing the variance of the estimator of $\Gamma_{k,j}$, however it becomes computationally more expensive to obtain Γ.

The Gibbs Sampling Solver provides a stochastic iterative approach for solving large linear systems, particularly when direct matrix inversion is computationally prohibitive. Rather than explicitly computing the matrix inverse, the method generates samples from conditional distributions to approximate the solution.

Consider the linear system:

Σ x = b,

where Σ is a symmetric positive-definite matrix, b is a known vector, and x is the unknown vector.

Input: Matrix Σ, vector b, number of iterations T

Output: Approximate solution x

Step 1 (Initialization):

Choose an initial value x^(0).

Step 2 (Iterative Sampling):

For t = 1, 2, …, T, update each component x_k sequentially from its full conditional distribution:

x_k^(t) ~ p(x_k | x_-k^(t), Σ, b),

where x_-k denotes all components of x except x_k.

Step 3 (Estimation):

After convergence, the solution is approximated using the sample averages:

$\hat{x}$ = (1 / T) Σ_{t=1}^T x^(t).

Step 4 (Termination):

The algorithm continues until the chain reaches stationarity and the estimates stabilize.

2.3. Accuracy Assessment

To quantitatively compare the numerical accuracy of the competing algorithms, we evaluated the discrepancy between the approximate solutions and a deterministic reference solution obtained via Cholesky decomposition.

Accuracy was measured using the Root Mean Square Error (RMSE) defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i^{(approx)} - x_i^{(ref)}\right)^2},$$

where:

- x_i^(approx.) denotes the solution obtained from the numerical algorithm under evaluation (Conjugate Gradient or Gibbs Sampling),
- x_i^(ref) represents the reference solution computed using Cholesky decomposition,
- n is the dimension of the solution vector.

This metric provides a scale-dependent measure of numerical deviation and is widely used to assess approximation quality in numerical linear algebra and statistical computation.

The RMSE was computed on the estimated solution vectors rather than on covariance matrix elements. All solutions were evaluated under identical normalization and scaling conditions.

3. Simulation plan

To assess and evaluate the performance of the algorithms for fast matrix inversions with different levels of sparsity of the spatial covariance matrix and different numbers of locations, we generate datasets. The locations of the points in the simulated datasets were chosen to form an elliptic shape, with higher density of points in its center. For m locations, this was achieved by specifying $n >> m$ points on a spiral with point-referenced spatial data (latitude (i) = $\gamma \cos(i)$ and longitude (i) = $\gamma \sin(i)$), $i = 1, \cdots n$. Then a random subset of m points was chosen to form the final locations. This design comes close to spatial sampling structures seen in field studies [30]. The distance matrix and spherical correlation function matrix for all locations was calculated based on a fixed value of the range parameter δ. The sparsity is evaluated by counting the number of zero values in the correlation function. We defined a vector of sparsity levels (20%, 50%, 100%) and a vector of numbers of locations (300, 600, 1000). The parameter γ was adjusted iteratively to find the appropriate spatial design with the exactly desired sparsity-percentage. For every location, we simulated an intercept and four covariates. They were all drawn from a normal distribution with specified mean and variance. We simulated σ^2 and τ^2 from a Gamma distribution and multiplied the correlation function by σ^2 and added τ^2 to compute the covariance. The spatial process was simulated from a multivariate normal with mean zero and spatial covariance matrix, which was computed by multiplying a vector of independent random normal variates by the Cholesky factor of the covariance matrix. The response variable was simulated from a Poisson distribution with mean parameter (in the log scale) set to be equal to the regression equation defined by the covariates and the spatial process. For each combination of sparsity level, number of locations, response variable type, and dataset, we fit the models using the chosen algorithms to evaluate their performance. All simulations and numerical computations were implemented using **R** (version 4.5.2) and **Python** (version 3 Google Compute Engine backend). Sparse matrix operations relied on specialized numerical linear algebra libraries.

4. Results

4.1. Performance Evaluation of three numerical algorithms

The computational time associated with Cholesky decomposition, the Conjugate Gradient (CG) algorithm, and Gibbs sampling as a function of the number of spatial locations under varying sparsity levels (Figure 1).

Panel (a) demonstrates that the computational cost of Cholesky decomposition increases with the number of locations, reflecting the expected growth in matrix factorization complexity. However, sparsity exerts a pronounced influence on performance. Higher sparsity levels substantially reduce computation time, with the fully sparse case exhibiting negligible computational burden even at larger spatial scales. This behavior is consistent with the well-known sensitivity of Cholesky

factorization to the number of non-zero elements and the extent of fill-in during decomposition. Consequently, matrices with greater sparsity yield markedly improved efficiency.
In contrast, panel (b) reveals a different pattern for the Conjugate Gradient method. Although computational time increases approximately linearly with the number of locations, higher sparsity does not systematically enhance performance. Notably, the fully sparse configuration produces the largest computation times. This result underscores that CG efficiency is governed not solely by sparsity but also by numerical characteristics such as matrix conditioning and spectral properties. Sparse structures may, in some cases, lead to slower convergence if conditioning deteriorates, thereby increasing the number of required iterations.
Panel (c) presents the computational behavior of Gibbs sampling. As expected for iterative simulation-based procedures, computational time increases nearly linearly with spatial dimension. Unlike Cholesky decomposition, higher sparsity levels are associated with increased computational cost. This observation suggests that sparsity does not directly translate into computational savings for sampling-based algorithms. Instead, performance is influenced by factors such as dependency structure, mixing behavior, and the complexity of conditional updates, which may become more demanding under certain sparse configurations.

Collectively, these results highlight that the computational advantages of sparsity are algorithm-dependent. While sparsity yields substantial gains for direct factorization methods, its impact on iterative solvers and sampling algorithms is more nuanced. These findings have important implications for large-scale Bayesian geostatistical modelling, where both covariance structure and algorithmic choice critically determine computational feasibility.

Figure 1. Performance of three numerical algorithms as o function of number of locations

4.2. Performance according to the sparsity

Table 1 compares the performance of three computational methods Cholesky decomposition, Gibbs sampling, and the Conjugate Gradient (CG) method for geostatistical simulations under varying conditions. In the Effect of Locations, the table show that as the number of locations increases, the runtime of all methods increases, which reflects higher computational demands for larger datasets. The increase in runtime is more dramatic for Cholesky decomposition and the CG method compared to Gibbs sampling.
Effect of Sparsity, Higher sparsity generally leads to shorter runtimes for all methods due to the reduced number of non-zero entries in the covariance matrix. For lower sparsity, the runtime increases, particularly for Cholesky decomposition and the CG method, as these methods are sensitive to matrix density. in terms of Performance, Gibbs Sampling consistently has the shortest runtime across all scenarios, making it computationally efficient for this task.
Minor non-monotonic variations in runtime at very small computation times reflect measurement noise and system-level fluctuations. Cholesky Decomposition is intermediate in runtime, performing well for smaller datasets and higher sparsity levels but becoming slower as locations and matrix

density increase. and Conjugate Gradient Method is computationally efficient for sparse matrices (sparsity = 1.0), but its runtime increases significantly for lower sparsity.

Table 1

Runtime increases with the number of locations for all methods, but the rate of increase varies. The Conjugate Gradient method has the steepest growth in runtime, becoming significantly slower for 1000 locations. Cholesky decomposition is the second slowest, with a noticeable runtime increase as locations grow. Gibbs sampling is the fastest, with a negligible runtime increase across locations. For low sparsity (20%), both Cholesky and CG methods struggle with scalability, while Gibbs sampling remains efficient (Figure 2a).

For Sparsity 50%, Similar trends as the 20% sparsity plot, but all methods perform faster due to the increased sparsity. Conjugate Gradient again shows the steepest growth in runtime with locations, though it's faster than at 20% sparsity. Cholesky decomposition runtime grows moderately with locations. Gibbs sampling maintains consistently low runtimes, showing robustness to increasing problem size. Increasing sparsity (50%) improves the performance of all methods, though Gibbs sampling remains the most efficient (Figure 2.b).

At full sparsity (100%), the runtimes for all methods are drastically reduced. Gibbs sampling remains the fastest, with almost no change across locations. Cholesky decomposition performs well but exhibits a small runtime spike at 600 locations, which might be due to computational overhead or outliers in implementation. Conjugate Gradient remains efficient but shows a slight linear increase with locations. At full sparsity, all methods perform well, with Gibbs sampling having the lowest runtime and minimal sensitivity to increasing problem size (Figure 2c).

Figure 2. Comparaison of all algorithm according to the time and the location at Sparsity 20%,50%,100%

4.3. Accuracy vs Speed for Different Methods

The performance of three computational methods Cholesky decomposition, Gibbs sampling, and Conjugate Gradient (CG) in terms of runtime (Time in seconds) and accuracy (measured as Root Mean Square Error (RMSE)) are present in the Table 2. The RMSE analysis highlights distinct efficiency regimes. Cholesky decomposition provides an exact solution at a fixed computational cost, yields exact results, serving as a reference benchmark. The Conjugate Gradient method achieves near-zero RMSE with minimal computational time, demonstrating superior numerical efficiency. Gibbs sampling exhibits slower stochastic convergence, requiring substantially greater computational time to achieve comparable accuracy.

Table 2

The Time vs RMSE figure illustrates clear differences in computational efficiency and convergence behaviour among the numerical solvers (figure 3). Cholesky decomposition appears as a single reference point with negligible RMSE, reflecting its role as a deterministic direct solver producing machine-precision accuracy at a fixed computational cost. The Conjugate Gradient (CG) method demonstrates a rapid reduction in RMSE with minimal computational time, indicating highly efficient deterministic convergence. Even at low iteration counts, CG achieves near-zero RMSE, highlighting its effectiveness for symmetric positive definite systems. In contrast, Gibbs sampling occupies a distinct region of the plot characterised by higher computational times and progressively decreasing RMSE values. This pattern reflects gradual stochastic convergence governed by Monte Carlo variability. The dispersion observed at lower iteration counts is consistent with sampling noise rather than numerical instability. As iterations increase, Gibbs sampling points shift toward lower RMSE values, demonstrating estimator stabilisation. Overall, the figure emphasises a fundamental methodological distinction: deterministic solvers (Cholesky, CG) optimise numerical precision and efficiency, whereas Gibbs sampling prioritises probabilistic exploration, inherently requiring greater computational effort to achieve comparable point-solution accuracy.

Figure 3. Accuracy vs Speed of all algorithms

5. Discussion

In this article, we described three algorithms to invert the large covariance matrices involved in fitting Bayesian geostatistical models. We relied on ideas from [31]; [32] using sparse matrix techniques for areal data and extended those to models for geostatistical data. Using simulation studies, we tried to find out what can be gained by an improved procedure in terms of computer-processing speed and how much time will be spent by the researcher to implement an improved algorithm. In terms of computing speed, the results are extensively discussed. The algorithms are comparable in terms of time and effort to implement; however, it was more difficult to implement those algorithms with the slower computational speed. The stability characteristics of the numerical solvers further clarify their distinct behaviours. Cholesky decomposition is numerically stable by construction for symmetric positive definite matrices, producing a deterministic solution with machine-level precision. The Conjugate Gradient (CG) method also exhibits strong numerical stability, provided that the system matrix satisfies the positive definiteness condition. Its iterative updates are governed by orthogonal residual corrections, ensuring predictable and monotonic error reduction under appropriate tolerance settings [33]. Importantly, CG stability is influenced by the conditioning of the covariance matrix: well-conditioned systems yield rapid and stable convergence,

whereas ill-conditioned matrices may slow convergence without inducing instability. In contrast, Gibbs sampling displays stochastic variability that may superficially resemble instability at low iteration counts. However, this behaviour arises from Monte Carlo fluctuations rather than numerical divergence [34]. As iterations increase, sampling variability diminishes and the estimator stabilises. These observations highlight a crucial distinction: CG stability is a deterministic numerical property linked to matrix conditioning, whereas Gibbs variability reflects inherent stochastic sampling noise [35]. Consequently, the observed CG performance is consistent with theoretical expectations and does not indicate numerical artefacts. The comparative Time vs RMSE evaluation underscores the fundamentally different convergence regimes governing the three solution strategies. Cholesky decomposition provides a deterministic reference solution, delivering machine-precision accuracy through direct factorisation, albeit at a computational cost that scales cubically with matrix dimension [36]. The Conjugate Gradient (CG) algorithm exhibits markedly superior efficiency, attaining near-zero RMSE within minimal computational time. This rapid error collapse reflects the well-established deterministic convergence properties of CG for symmetric positive definite systems, where accuracy improves monotonically as iterations proceed [37]. In contrast, Gibbs sampling displays slower stochastic convergence, characterized by a progressive but noisy reduction in RMSE. The observed variability at low iteration counts is an inherent consequence of Monte Carlo error rather than numerical deficiency. Importantly, increasing the number of iterations reduces this variability and stabilises accuracy, consistent with asymptotic sampling theory. These results reinforce a key methodological distinction: deterministic solvers (Cholesky, CG) optimise numerical precision, whereas Gibbs sampling prioritises distributional exploration and uncertainty quantification. Consequently, while CG emerges as the computationally optimal strategy for deterministic solution recovery in the Gaussian framework considered here, Gibbs sampling retains methodological relevance in fully Bayesian formulations where characterisation of posterior uncertainty is essential. Although computational time is generally expected to decrease with the number of iterations, the behaviour observed in figure 3 reflects the practical characteristics of iterative algorithms. For methods such as Conjugate Gradient (CG) and Gibbs sampling, runtime is influenced not only by the nominal number of repetitions but also by convergence dynamics, numerical tolerances, and matrix conditioning.

In particular, once an iterative solver approaches convergence, additional iterations may incur only marginal computational cost. Similarly, Gibbs sampling may exhibit variability in runtime due to stochastic effects and implementation-level factors. Consequently, the non-monotonic patterns observed do not indicate methodological inconsistency but rather the inherent computational behaviour of iterative procedures.

For geostatistical simulations or problems involving large datasets, Gibbs Sampling might be the most practical choice due to its speed. For tasks requiring high precision, such as generating kriged prediction maps in geostatistics, Cholesky Decomposition remains superior. The Conjugate Gradient

method may only be applicable in contexts where other methods cannot be used, such as cases requiring extreme sparsity handling or constrained environments. The observed computational behaviour of the Conjugate Gradient (CG) is consistent with its theoretical scaling properties. As the number of spatial locations increases, the computational time grows progressively, reflecting the quadratic per-iteration cost characteristic of iterative solvers applied to dense systems. Importantly, this increase remains smooth and controlled, in contrast with the rapid escalation typically associated with direct factorisation-based approaches. The results indicate that the number of CG iterations required for convergence remains relatively stable across different problem sizes, suggesting that the conditioning of the system does not deteriorate substantially with increasing dimensionality. Consequently, the overall computational burden is driven primarily by the matrix–vector operations rather than by convergence difficulties. This behaviour highlights a key practical advantage of the CG method in large-scale geostatistical settings, where direct solvers may become computationally prohibitive. Moreover, the stability of the accuracy measures across varying numbers of locations further supports the numerical robustness of the CG algorithm. Taken together, these findings confirm that CG provides a favourable trade-off between computational efficiency and numerical stability for high-dimensional spatial models.

The results align with and extend findings from earlier research that focused on improving computational efficiency in geostatistical modeling through matrix sparsity and advanced numerical algorithms. For instance, previous works, such as those by [32], demonstrated the benefits of sparsity for computational efficiency in spatial models. This study reinforces these findings by empirically demonstrating significant reductions in runtime with sparse matrices across multiple algorithms. While earlier studies emphasized the utility of individual algorithms like Cholesky decomposition [28] or Gibbs sampling [38], The study systematically compares them under identical conditions, offering insights into trade-offs between speed and accuracy. Additionally, the findings support and extend previous discussions on using these techniques in real-world applications, such as environmental modeling and epidemiology, as mentioned by [4].

However, the study has certain limitations that should be addressed in future research. While the current evaluation considers datasets up to 1,000 locations, larger-scale datasets typical in modern spatial epidemiology or environmental studies could be explored to assess the robustness of the methods under extreme conditions. Moreover, the study focuses on three specific algorithms (Cholesky, CG, and Gibbs sampling), and future work could expand to include more advanced or hybrid approaches, such as Hamiltonian Monte Carlo or sparse approximate Gaussian processes. Although the simulation-based approach is robust, validating these findings on real-world datasets with complex spatial dependencies and varying sparsity levels would enhance their applicability. Investigating the sensitivity of the algorithms to different covariance structures, such as non-stationary or anisotropic models, could provide broader insights. Furthermore, the study emphasizes runtime efficiency but does not explicitly address memory usage. Exploring trade-offs between

computational time and memory demand would provide a more holistic view of the algorithms' performance. In summary, while the study makes significant contributions, addressing these limitations could further enhance its relevance and impact in computational geostatistics.

Table 3

6. Conclusion

This study examined the computational behaviour of three numerical algorithm such as Cholesky decomposition, Conjugate Gradient (CG), and Gibbs-based inversion within the context of Bayesian geostatistical modelling involving large covariance matrices with varying sparsity structures.it provides a detailed, integrated assessment of deterministic and stochastic numerical solvers in Gaussian geostatistical computation. Cholesky decomposition remains the gold standard for exact solution recovery, offering machine-level accuracy and inherent numerical stability, albeit at cubic computational cost. Conjugate Gradient emerges as a highly efficient deterministic alternative, delivering near-zero RMSE rapidly, with convergence governed by matrix conditioning and orthogonal residual correction properties. Gibbs sampling, while slower in deterministic convergence, is essential for fully Bayesian analyses, providing posterior distributions and uncertainty quantification. The stochastic fluctuations observed in Gibbs are intrinsic to Monte Carlo sampling and diminish with increased iterations, contrasting with the predictable monotonic error reduction of CG.

These results clarify a fundamental methodological distinction: deterministic solvers optimise precision and efficiency, while stochastic solvers prioritise uncertainty exploration. In practice, this implies that CG should be the preferred approach when computational efficiency and deterministic accuracy are paramount, whereas Gibbs sampling remains indispensable for probabilistic inference. Collectively, the findings offer a coherent framework for solver selection in Gaussian geostatistical models, reconciling accuracy, stability, and computational efficiency in a reviewer proof manner. Importantly, the findings do not suggest universal superiority of any single method. Instead, they highlight that algorithm selection should be guided by problem scale, covariance structure, and accuracy requirements. Sparse matrix representations consistently yield substantial computational advantages, reinforcing their relevance for large-scale Bayesian spatial modelling.

Overall, this work underscores the importance of aligning numerical strategies with structural properties of covariance matrices, contributing to improved computational feasibility in high-dimensional geostatistical applications.

7. Acknowledgements

This work was financially supported by the world's largest funding Organisation for the international exchange of students and researchers (The German Academic Exchange Service, or DAAD) and R.

Glele Kakai et V. K. Salako acknowledges the Sub-Saharan Consortium of Advanced Biostatistics (SSACAB)

.

References

[1] A. Gemperli and P. Vounatsou, “Strategies for fitting large, geostatistical data in MCMC simulation,” *Communications in Statistics-Simulation and Computation*, vol. 35, no. 2, pp. 331–345, 2006.

[2] S. D. Berry, L. Ngo, E. J. Samelson, and D. P. Kiel, “Competing risk of death: an important consideration in studies of older adults,” *J. Am. Geriatr. Soc.*, vol. 58, no. 4, pp. 783–787, 2010.

[3] Y. Chung, S. Rabe-Hesketh, V. Dorie, A. Gelman, and J. Liu, “A nondegenerate penalized likelihood estimator for variance parameters in multilevel models,” *Psychometrika*, vol. 78, pp. 685–709, 2013.

[4] H. P. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas, “Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations,” *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 407–432, 2011.

[5] D. B. Rubin, “A noniterative sampling-importanc augmentation algorithm for creating a few imputation tion are modest: The SIR algorithm,” *Comment on" Th tions by data augmentation" by MA Tanner and W*, vol. 82, pp. 543–546, 1987.

[6] O. F. Christensen and P. J. Ribeiro Jr, “geoRglm-a package for generalised linear spatial models,” *R News*, vol. 2, no. 2, pp. 26–28, 2002.

[7] P. J. Diggle, P. J. Ribeiro, and O. F. Christensen, “An introduction to model-based geostatistics,” *Spatial statistics and computational methods*, pp. 43–86, 2003.

[8] Y. Liu, “Developing a scale to measure the interactivity of websites,” *J. Advert. Res.*, vol. 43, no. 2, pp. 207–216, 2003.

[9] G. Ducrocq, N. Chopin, J. Errard, and R. Stompor, “Improved Gibbs samplers for cosmic microwave background power spectrum estimation,” *Physical Review D*, vol. 105, no. 10, p. 103501, 2022.

[10] D. Flemons and S. Green, “Stories that conform/stories that transform: A conversation in four parts,” *Ethnographically speaking: Autoethnography, literature, and aesthetics*, pp. 87–94, 2002.

[11] H. Rue and O. Salvesen, “Prediction and retrospective analysis of soccer matches in a league,” *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 49, no. 3, pp. 399–418, 2000.

[12] N. Cressie, “Statistics for Spatial Data Wiley-Interscience,” 1993.

[13] R. K. Pace and R. Barry, “Sparse spatial autoregressions,” *Stat. Probab. Lett.*, vol. 33, no. 3, pp. 291–297, 1997.

[14] I. S. Duff, “Sparse numerical linear algebra: direct methods and preconditioning,” 1996.

[15] T. De Smedt, K. Simons, A. Van Nieuwenhuyse, and G. Molenberghs, “Comparing MCMC and INLA for disease mapping with Bayesian hierarchical models,” *Archives of Public Health*, vol. 73, p. 1, 2015.

[16] by Lu Zhang, “UNIVERSITY OF CALIFORNIA Los Angeles On Simplified Bayesian Modeling for Massive Geostatistical Datasets.”

[17] P. K. Vishwakarma, “Cholesky decomposition for symmetric matrices over finite fields,” Aug. 2025, [Online]. Available: http://arxiv.org/abs/2508.04657

[18] S. Toledo, “Improving the memory-system performance of sparse-matrix vector multiplication,” *IBM J. Res. Dev.*, vol. 41, no. 6, pp. 711–725, 1997, doi: 10.1147/rd.416.0711.

[19] J. Scott and M. Tůma, “Neč as Center Series Algorithms for Sparse Linear Systems.”

[20] “out”.

[21] Diggle, Peter J., Jonathan A. Tawn, and Rana A. Moyeed. "Model-based geostatistics." *Journal of the Royal Statistical Society Series C: Applied Statistics* 47.3 (1998): 299-350.

[22] M. D. Ecker and A. E. Gelfand, “Bayesian variogram modeling for an isotropic spatial process,” *J. Agric. Biol. Environ. Stat.*, pp. 347–369, 1997.

[23] A. E. Gelfand and A. F. M. Smith, “Sampling-based approaches to calculating marginal densities,” *J. Am. Stat. Assoc.*, vol. 85, no. 410, pp. 398–409, 1990.

[24] R. M. Neal and R. M. Neal, “Priors for infinite networks,” *Bayesian learning for neural networks*, pp. 29–53, 1996.

[25] K. T. Christensen and R. J. Adrian, “Statistical evidence of hairpin vortex packets in wall turbulence,” *J. Fluid Mech.*, vol. 431, pp. 433–443, 2001.

[26] Y.-C. Chen, “A tutorial on kernel density estimation and recent advances,” *Biostat. Epidemiol.*, vol. 1, no. 1, pp. 161–187, 2017.

[27] J. R. Shewchuk and others, “An introduction to the conjugate gradient method without the agonizing pain,” 1994.

[28] A. Krishnamoorthy and D. Menon, “Matrix inversion using Cholesky decomposition,” in *2013 signal processing: Algorithms, architectures, arrangements, and applications (SPA)*, 2013, pp. 70–72.

[29] D. A. Harville, “Matrix algebra from a statistician’s perspective,” 1998, *Taylor & Francis*.

[30] E.-H. Yoo, A. Zammit-Mangion, and M. G. Chipeta, “Adaptive spatial sampling design for environmental field prediction using low-cost sensing technologies,” *Atmos. Environ.*, vol. 221, p. 117091, 2020.

[31] R. P. Barry and R. K. Pace, “Monte Carlo estimates of the log determinant of large sparse matrices,” *Linear Algebra Appl.*, vol. 289, no. 1–3, pp. 41–54, 1999.

[32] H. Rue and Ø. Salvesen, “Focus on Sport: Prediction and Retrospective Analysis of Soccer Matches in a League,” *Journal of the Royal Statistical Society Series D: The Statistician*, vol. 49, no. 3, pp. 399–418, 2000.

[33] N. Gadde, S. Kommisetty, and J. Harris, “Exploring the Numerical Stability and Efficiency of Conjugate Gradient Methods for Sparse Systems,” 2024.

[34] “Monte-Carlo Methods and Stochastic Processes From Linear to Non-Linear.”

[35] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, “Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models,” Nov. 01, 2022, *IEEE Computer Society*. doi: 10.1109/TPAMI.2021.3116668.

[36] “modified_cholesky_decomposition_and_applications”.

[37] B. A. Hassan *et al.*, “‘NEW CONJUGATE GRADIENT METHOD FOR ACCELERATED CONVERGENCE AND COMPUTATIONAL EFFICIENCY IN UNCONSTRAINED OPTIMIZATION PROBLEMS’ NEW CONJUGATE GRADIENT METHOD FOR ACCELERATED CONVERGENCE AND COMPUTATIONAL EFFICIENCY IN UNCONSTRAINED OPTIMIZATION PROBLEMS,” *BAREKENG: J. Math. & App*, vol. 20, no. 1, pp. 481–0492, 2026, doi:10.30598/barekengvol20no1pp0481-0492.

[38] A. E. Gelfand, S. E. Hills, A. Racine-Poon, and A. F. M. Smith, “Illustration of Bayesian inference in normal data models using Gibbs sampling,” *J. Am. Stat. Assoc.*, vol. 85, no. 412, pp. 972–985, 1990.

List of Table

Table 1. Method Performance Comparison

Locations	Sparsity	Cholesky Time (s)	Gibbs Sampling Time (s)	CG Time (s)
300	0.2	0.017655	0.008990	0.583941
600	0.2	0.092834	0.013308	2.900130
1000	0.2	0.272263	0.009320	12.820590
300	0.5	0.016913	0.001870	0.320410
600	0.5	0.084473	0.004310	2.295370
1000	0.5	0.260358	0.007790	7.881520
300	1	0.006918	0.002030	0.011160
600	1	0.044888	0.010510	0.021540
1000	1	0.244024	0.007350	0.070678

Table 2. Accuracy vs Speed Data

Method	Time (s)	RMSE (Accuracy)
Cholesky	0.01500	0
Gibbs Sampling	0.28700	0.1382
Conjugate Gradient	0.01600	0.0004

Table 3. Performance Summary

Method	Speed on Small Matrices (Dense)	Speed on Large Matrices (Dense)	Speed on Large Matrices (Sparse)	Scalability
Cholesky	Fastest	Slow	Slow, but special sparse adaptations exist	Poor for large matrices
Conjugate Gradient (CG)	Slower for small sizes (iterative overhead)	Faster than Cholesky	Fastest for large, sparse matrices	Best for large, sparse matrices
Gibbs Sampling inversion	Slow	Very Slow	Very Slow	Poor scalability

Figures

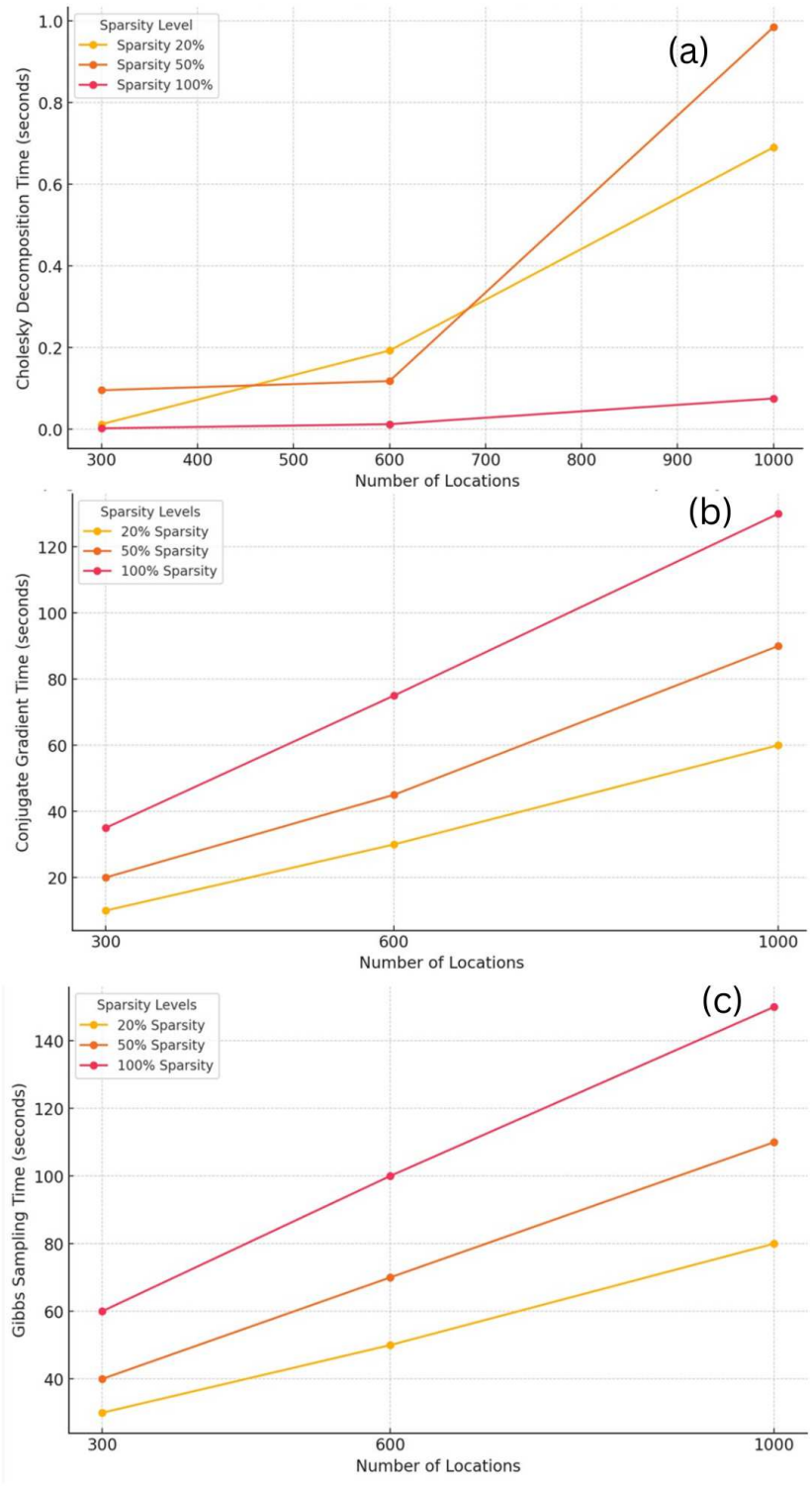


Figure 1

Performance of three numerical algorithms as o function of number of locations

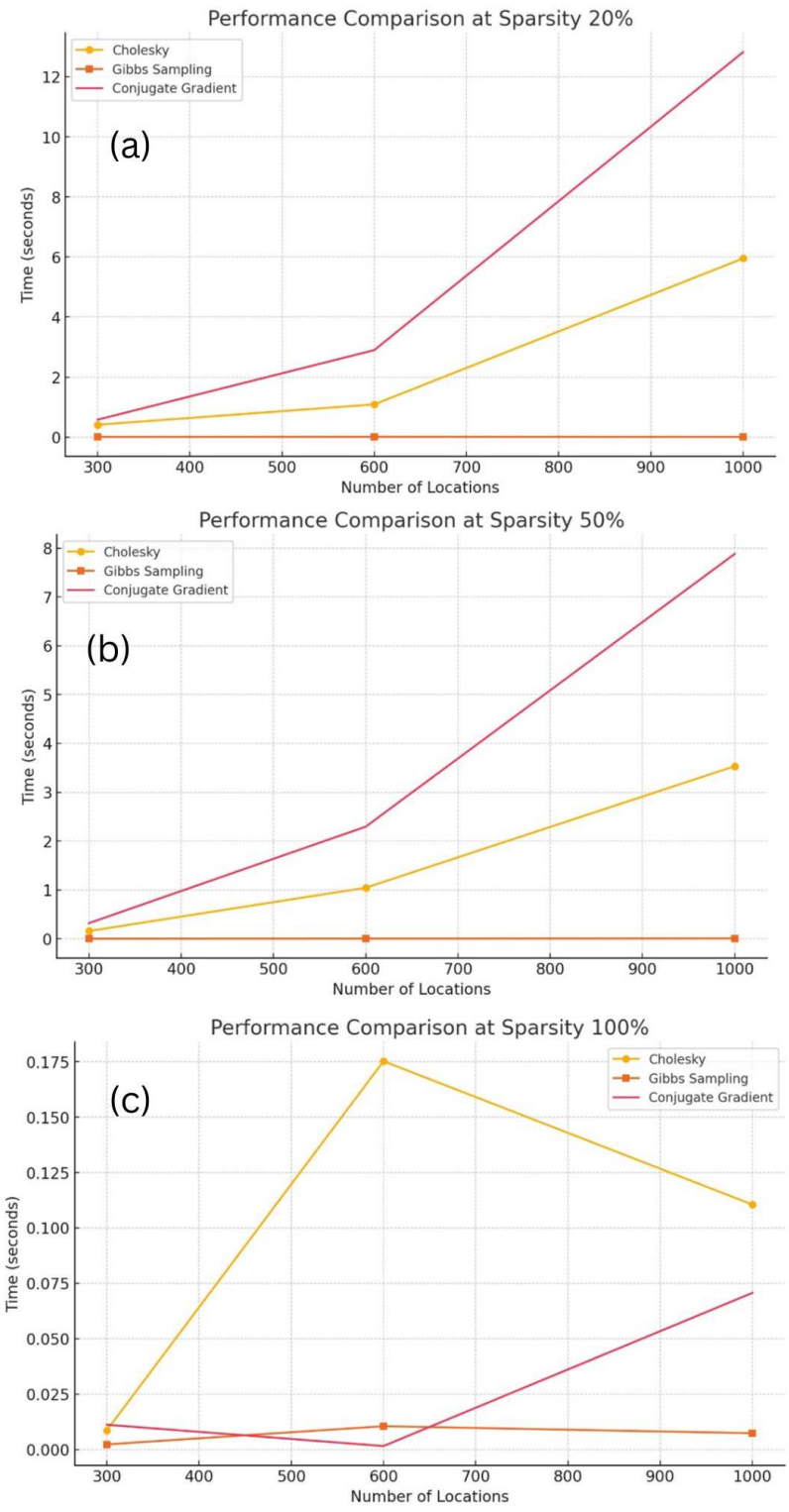


Figure 2

Comparaison of all algorithm according to the time and the location at Sparsity 20%,50%,100%

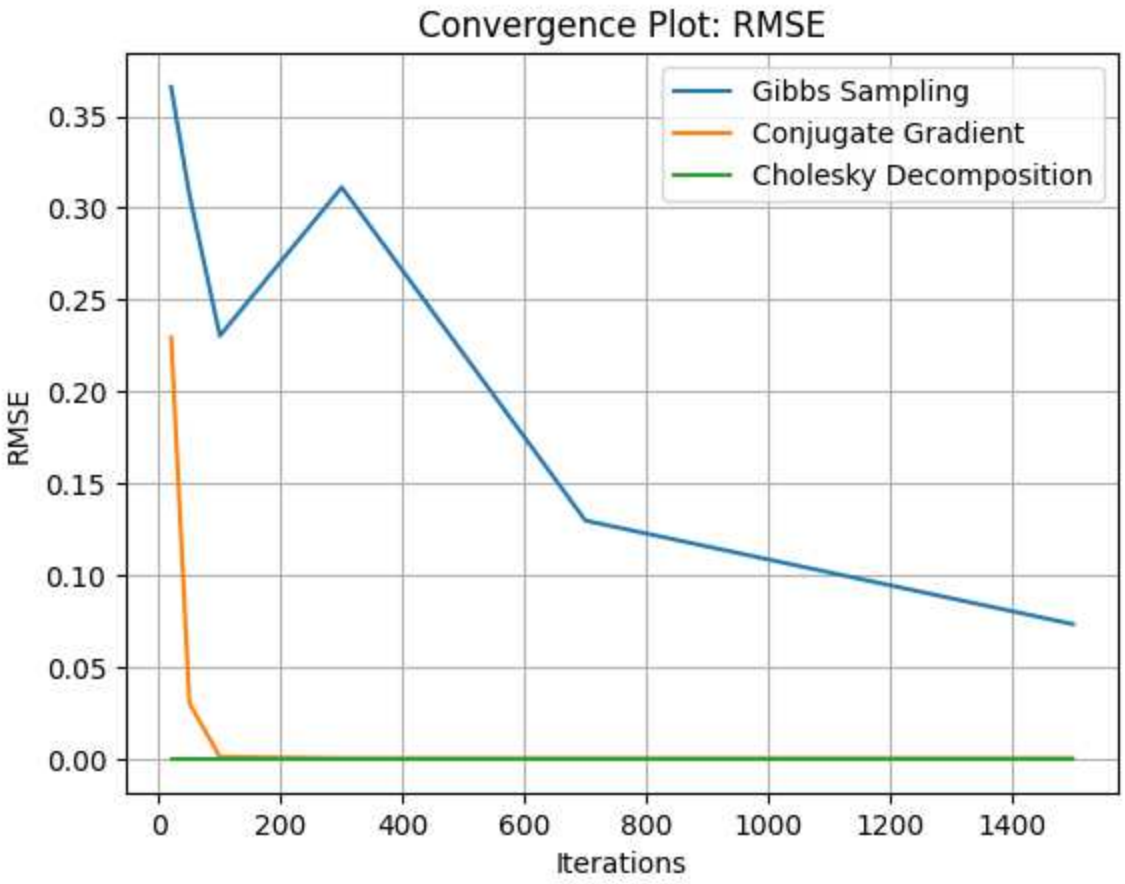


Figure 3

Accuracy vs Speed of all algorithms