Oil & Gas Science and Technology – Rev. IFP Energies nouvelles
Copyright © 2012, IFP Energies nouvelles
DOI: 10.2516/ogst/2012064

IFP Energies nouvelles International Conference
Rencontres Scientifiques d'IFP Energies nouvelles

Pore2Field - Flows and Mechanics in Natural Porous Media from Pore to Field Scale
Pore2Field - Physique des écoulements en milieux poreux naturels : de l'échelle du pore à l'échelle du réservoir

Application of Hierarchical Matrices to Linear Inverse Problems in Geostatistics

A.K. Saibaba[1], S. Ambikasaran[1], J. Yue Li[2], P.K. Kitanidis[1,2]* and E.F. Darve[1,3]

1 Institute for Computational and Mathematical Engineering, Stanford University - USA
2 Department of Civil and Environmental Engineering, Stanford University - USA
3 Department of Mechanical Engineering, Stanford University - USA
e-mail: arvindks@stanford.edu - sivaambi@stanford.edu - yuel@stanford.edu - peterk@stanford.edu - darve@stanford.edu

* Corresponding author

Résumé — Application des matrices hiérarchiques aux problèmes d'inversion linéaire en géostatistique — La caractérisation de l'incertitude en subsurface est une étape importante, tant, par exemple, dans le cadre de la recherche et de l'extraction de ressources naturelles, que dans celui du stockage de matériaux radioactifs ou de gaz tels que le gaz naturel ou le CO_2. L'imagerie en subsurface peut être posée comme un problème inverse et être résolue à l'aide d'une approche géostatistique [Kitanidis P.K. (2007) *Geophys. Monogr. Ser.* **171**, 19-30, doi :10.1029/171GM04; Kitanidis (2011) doi: 10.1002/9780470685853. ch4, pp. 71-85] qui est l'une des méthodes importantes de ce domaine. Nous commençons par décrire brièvement l'approche géostatistique dans le contexte d'un problème inverse linéaire, puis discutons les difficultés rencontrées dans le cadre d'une implémentation à grande échelle. Ensuite, en utilisant une approche basée sur les matrices hiérarchiques, nous montrons comment réduire le coût de calcul d'un produit matrice-vecteur de $O(m^2)$ à $O(m \log m)$ dans le cas de matrices de covariance denses ; m désignant ici le nombre d'inconnues. Combinée avec un solveur de Krylov, qui ne requiert pas la construction explicite de la matrice, cette méthode conduit à un algorithme beaucoup plus rapide pour résoudre le système d'équations associé à l'approche géostatistique. Nous illustrons enfin la performance de notre algorithme dans un situation spécifique, surveiller la concentration de CO_2 à l'aide de la tomographie sismique entre puits.

Abstract — Application of Hierarchical Matrices to Linear Inverse Problems in Geostatistics — *Characterizing the uncertainty in the subsurface is an important step for exploration and extraction of natural resources, the storage of nuclear material and gasses such as natural gas or CO_2. Imaging the subsurface can be posed as an inverse problem and can be solved using the geostatistical approach [Kitanidis P.K. (2007) Geophys. Monogr. Ser.* ***171****, 19-30, doi:10.1029/171GM04; Kitanidis (2011) doi: 10.1002/9780470685853. ch4, pp. 71-85] which is one of the many prevalent approaches. We briefly describe the geostatistical approach in the context of linear inverse problems and discuss some of the challenges in the large-scale implementation of this approach. Using the hierarchical matrix approach, we show how to reduce matrix vector products involving the dense covariance matrix from $O(m^2)$ to $O(m \log m)$, where m is the number of unknowns. Combined with a matrix-free Krylov subspace solver, this results in a much faster algorithm for solving the system of equations that arise from the geostatistical approach. We illustrate the performance of our algorithm on an application, for monitoring CO_2 concentrations using crosswell seismic tomography.*

INTRODUCTION

Inverse problems arise frequently in the context of earth sciences, such as hydraulic tomography [1-4], crosswell seismic traveltime tomography [5-7], electrical resistivity tomography [8, 9], contaminant source identification [10-13], etc. A common feature amongst inverse problems is that the parameters we are interested in estimating are hard to measure directly and a crucial component of inverse modeling is using sparse data to evaluate model parameters, *i.e.*, the solution to the inverse problems. Inverse problems are typically ill-posed for three reasons:

- the functional form of the relation between the model parameters and the data may be over-sensitive;
- there is a lack of data due to insufficient measurements;
- the presence of noise in the available data and limitations of the model.

These imply that there is no unique solution to such inverse problem. Instead, there is a set of solutions consistent under the given model and data. Hence, one should not look for a unique solution but a statistical ensemble of solutions. The classical approach to deal with ill-posedness is to place additional restrictions on the solution, by the use of some kind of regularization such as Tikhonov regularization. This approach, while sufficient to produce the "best estimate" does not eliminate the uncertainty. On the other hand, statistical approaches to inverse problems model all the variables as random variables, which represents the degree of information concerning their realizations, which are encoded in terms of probability distributions. The solution to the inverse problem is the posterior probability distribution.

Geostatistics is a general method for solving such inverse problems, see for example [14]. The approach is based on the idea of combining data with information about the structure of the function that needs to be estimated. In Bayesian and geostatistical approaches (for example, see discussion in [15, 16]), the structure of the function is represented through the prior probability density function, which in practical applications is often parameterized through variograms and generalized covariance functions. The method has found several applications because it is generally practical and quantifies uncertainty. The method can generate best estimates, which can be determined in a Bayesian framework as *a posteriori* mean values or most probable values, measures of uncertainty as posterior variances or credibility intervals and conditional realizations, which are sample functions from the ensemble of the posterior probability distribution.

In recent years, there have been vast improvements in the measurement technology that allow the collection of even more measurements, tremendous improvements in computational power and advances in developing state-of-the-art computational techniques to solve forward problems, in other words the governing partial differential equations, that scale well on several processors. Moreover, there is a demand for more accurate prediction of parameters that govern flow and transport in complex geological processes. There are several challenges in large-scale implementations of inverse problems. As the number of unknowns increase, the storage and computational costs involving the dense covariance matrix that represents the Gaussian random field are overwhelming, especially on unstructured grids. Recently, we have developed methods [17, 18] for large-scale implementation of the geostatistical approach that scales to unknowns of the order of $\mathcal{O}(10^6)$ with $\mathcal{O}(10^3)$ measurements that uses of the hierarchical matrix approach [19-24] in order to reduce the costs involved in computing matrix-vector products from $\mathcal{O}(m^2)$ to $\mathcal{O}(m \log m)$ [18] or $\mathcal{O}(m)$ [17], where m is the number of unknowns.

The purpose of this paper is to introduce and review the geostatistical approach to solving linear inverse problems and describe in detail the computational techniques involved in large-scale implementation of these algorithms using the hierarchical matrix approach and to illustrate its utility on some real applications. In particular, we describe how hierarchical matrices can be used to accelerate matrix operations involving dense covariance matrices. We emphasize here that this article is more of a tutorial nature, rather than development of new ideas or techniques. The remainder of the article is organized as follows. In Section 1, we review the geostatistical approach and the approaches in the literature for fast algorithms to deal with large covariance matrix. The next section, Section 2, describes the essential steps in the implementation of the hierarchical matrix approach, starting from simple observations about the structure of covariance matrices to stating the full algorithm along with complexity estimates. Subsequently in Section 3, we provide numerical evidence of the log-linear scaling of hierarchical matrices for various kernels. Then, we show how to combine the hierarchical matrix formulation with direct solvers and appropriately chosen matrix-free Krylov subspace iterative techniques to solve the system of equations that result from the geostatistical approach. Finally, we illustrate the performance of our algorithm on applications from CO_2 monitoring.

We briefly mention some other approaches to deal with large spatial datasets, in the context of kriging. One approach is to use covariance tapering [25, 26], in which the correct covariance matrix is tapered using an appropriately chosen compactly supported radial basis function which results in a sparse approximation of the covariance matrix that can be solved using sparse matrix algorithms. Another approach is to choose classes of covariance functions for which kriging can be done exactly using a multiresolution spatial process [27-29]. Other approaches include fixed rank kriging [30]

Before we proceed, we would like to clarify the notations. Lower case bold face denotes vectors, upper case bold face denotes matrices and Greek alphabets denote scalars. In general, m denotes the number of unknowns and n stands for the number of measurements. The rest of the notations should be clear from the context.

1 GEOSTATISTICAL APPROACH

1.1 General Formulation

Suppose that we want to estimate a spatially variable parameter $s(\mathbf{x})$, for example the log hydraulic conductivity, in the domain $\mathcal{D}$. The prevalent approach is to write it as a sum of a deterministic function with adjustable coefficients and a small scale variability term, or random-walk type of variability, which is typically modeled using covariance functions or variograms (for example, see Fig. 1). After appropriate discretization, the parameters to be determined, $\mathbf{s} \in \mathbb{R}^{m\times 1}$ can be written as:

$$\mathbb{E}[\mathbf{s}] = \mathbf{X}\boldsymbol{\beta}; \qquad \mathbb{E}\left[(\mathbf{s}-\mathbf{X}\boldsymbol{\beta})(\mathbf{s}-\mathbf{X}\boldsymbol{\beta})^T\right] = \mathbf{Q} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{m\times p}$ is a known matrix, $\boldsymbol{\beta} \in \mathbb{R}^{p\times 1}$ is a vector of drift coefficients and $\mathbb{E}$ is the expectation. The entries of the covariance matrix $\mathbf{Q}$, are given by $\mathbf{Q}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, where $\kappa(\cdot,\cdot)$ is a generalized covariance function. The generalized covariance function [31, 32] must be conditionally positive definite. The forward problem relates the parameters/unknowns to be determined, $\mathbf{s}$, to the set of the measurements, $\mathbf{y} \in \mathbb{R}^{n\times 1}$, by the measurement equation:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{v} \qquad \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2}$$

where, we model the noise in the measurements $\mathbf{v}$ as a normal distribution, with zero mean and covariance matrix $\mathbf{R}$.

Using Bayes' theorem, assuming a uniform prior for $\boldsymbol{\beta}$, *i.e.*, $p(\boldsymbol{\beta}) \propto 1$, we have $p(\mathbf{s}, \boldsymbol{\beta})$, the prior probability distribution of the unknowns and the probability distribution of the error can be combined to give the posterior probability distribution of the unknown parameters, $\mathbf{s}$ and $\boldsymbol{\beta}$:

$$\begin{aligned} p(\mathbf{s}, \boldsymbol{\beta}|\mathbf{y}) &= \frac{p(\mathbf{y}|\mathbf{s}, \boldsymbol{\beta})p(\mathbf{s}, \boldsymbol{\beta})}{p(\mathbf{y})} \\ &\propto p(\mathbf{y}|\mathbf{s}, \boldsymbol{\beta})p(\mathbf{s}, \boldsymbol{\beta}) \end{aligned} \tag{3}$$

Plugging in the expression for the respective terms, and using an alternative convenient notation, we find that:

$$p(\mathbf{s}, \boldsymbol{\beta}|\mathbf{y}) \propto \exp\left(-\frac{1}{2}||\mathbf{s}-\mathbf{X}\boldsymbol{\beta}||_{\mathbf{Q}^{-1}} - \frac{1}{2}||\mathbf{y}-\mathbf{H}\mathbf{s}||_{\mathbf{R}^{-1}}\right) \tag{4}$$

where $||\mathbf{x}||_{\mathbf{M}}$ is the norm induced by the positive definite matrix M and is defined as $||\mathbf{x}||^2_{\mathbf{M}} = \mathbf{x}^T\mathbf{M}\mathbf{x}$. The resulting posterior distribution (4) is also Gaussian. The posterior mean values, $\hat{\mathbf{s}}$ and $\hat{\boldsymbol{\beta}}$ is given by maximizing the *a posteriori* estimate,

$$\arg\max_{\mathbf{s},\boldsymbol{\beta}} \exp\left(-\frac{1}{2}||\mathbf{s}-\mathbf{X}\boldsymbol{\beta}||_{\mathbf{Q}^{-1}} - \frac{1}{2}||\mathbf{y}-\mathbf{H}\mathbf{s}||_{\mathbf{R}^{-1}}\right) \tag{5}$$

The above Equation (5) is equivalent to the minimization problem given below in Equation (6):

$$\arg\max_{\mathbf{s},\boldsymbol{\beta}} \left(||\mathbf{s}-\mathbf{X}\boldsymbol{\beta}||_{\mathbf{Q}^{-1}} + ||\mathbf{y}-\mathbf{H}\mathbf{s}||_{\mathbf{R}^{-1}}\right) \tag{6}$$

For $n < m$, it is more convenient to compute the solution to this optimization problem (5) by first obtaining the solution of the following linear system of equations:

$$\begin{pmatrix} \mathbf{HQH}^T + \mathbf{R} & \mathbf{HX} \\ (\mathbf{HX})^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\xi}} \\ \hat{\boldsymbol{\beta}} \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} \tag{7}$$

and then computing the resulting unknown field from the solution of the system of Equations (7) by the following transformation:

$$\hat{\mathbf{s}} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{QH}^T\hat{\boldsymbol{\xi}} \tag{8}$$

We also denote by $\boldsymbol{\Psi} \overset{\text{def}}{=} \mathbf{HQH}^T + \mathbf{R}$, $\boldsymbol{\phi} \overset{\text{def}}{=} \mathbf{HX}$ and we also define the matrix $\mathbf{A}$ as:

$$\mathbf{A} = \begin{pmatrix} \boldsymbol{\Psi} & \boldsymbol{\phi} \\ \boldsymbol{\phi}^T & \mathbf{0} \end{pmatrix} \tag{9}$$

The covariance matrix of the posterior pdf of the parameters $\mathbf{s}$ and β is simply given by the inverse of the Hessian of the objective function that we minimized, *i.e.*

$$\begin{pmatrix} \mathbf{Q}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} & \mathbf{Q}^{-1}\mathbf{X} \\ \mathbf{X}^T\mathbf{Q}^{-1} & \mathbf{X}^T\mathbf{Q}^{-1}\mathbf{X} \end{pmatrix}^{-1} \tag{10}$$

A direct implementation of this algorithm can be done in $O(m^2n + mn^2 + mnp)$. However, for realistic problem sizes, the number of unknowns, m, is quite large, $O(10^5 - 10^6)$. The covariance matrix $\mathbf{Q}$ is dense, and the construction and storage costs are high. One possibility is to form $\mathbf{QH}^T$ directly without forming $\mathbf{Q}$. Even with this possibility, the time required to form $\mathbf{QH}^T$ is quite high. In the next section, we discuss several approaches to deal with this issue.

1.2 Covariance Functions

Consider the random Gaussian field $s(\mathbf{x})$, and let $\mathbf{x} \in \mathcal{D} \subset \mathbb{R}^d$ be a bounded domain, with a covariance Kernel $\kappa(\mathbf{x}, \mathbf{y})$.

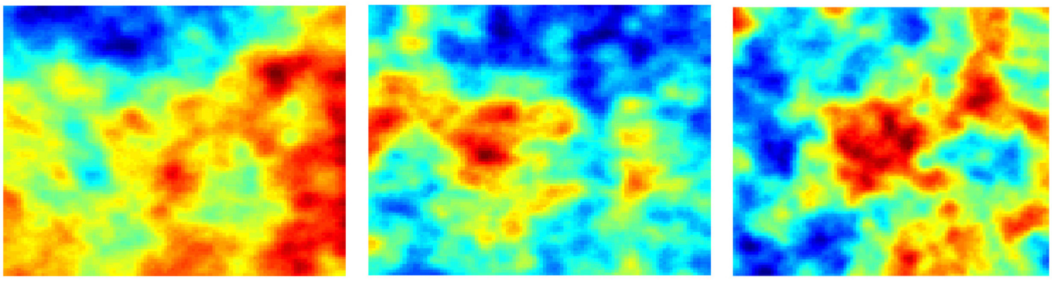

Figure 1

Three realizations of a Gaussian random field with mean zero exponential covariance function.

In terms of applications, it is useful to consider three kinds of covariance kernels [33]:

1. isotropic and translation invariant, *i.e.*, $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(|\mathbf{x}-\mathbf{y}|)$;
2. stationary and anisotropic, *i.e.*, $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} - \mathbf{y})$;
3. non-stationary.

Some possible choices for the covariance kernel $\kappa(\cdot,\cdot)$ arise from the Matérn family of covariance kernels [34], corresponding to a stationary and isotropic stochastic process. They are defined as $K_{\alpha,\nu}(\mathbf{x}, \mathbf{y}) = C_{\alpha,\nu}(r)$, $r = ||\mathbf{x} - \mathbf{y}||$ with:

$$C_{\alpha,\nu}(r) = \frac{\phi}{2^{\nu-1}\gamma(\nu)}(\alpha r)^{\nu}\mathcal{K}_{\nu}(\alpha r), \ \alpha > 0, \phi > 0, \nu > 0 \quad (11)$$

where, $\mathcal{K}_\nu$ is the modified Bessel function of second kind of order ν and γ is the gamma-function. Equation (11) takes special forms for certain parameters ν. For example, when $\nu = 1/2$, $C_{\alpha,\nu}$ corresponds to the exponential covariance function, $\nu = 1/2 + n$ where n is an integer, $C_{\alpha,\nu}$ is the product of an exponential covariance and a polynomial of order n. In the limit as $\nu \to \infty$ and for appropriate scaling of α, $C_{\alpha,\nu}$ converges to the Gaussian covariance kernel. For a more detailed discussion of permissible covariance kernels, we refer the reader to the following references [31,32,35].

After appropriate discretization of the domain, the covariance matrix takes the form $\mathbf{A}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ for a set of points $\{\mathbf{x}_i\}$. Covariance matrices, although dense, have special structure which can be exploited. They are similar to dense matrices that arise from the discretization of integral equations. Various fast summation schemes have been devised to provide matrix-vector products in $O(N \log^{\beta} N)$ for such problems, where N is the number of unknowns and $\beta \geq 0$ an integer depending on the method. They broadly fall under three different categories:

- FFT (Fast Fourier Transform) based methods,
- fast multipole methods [36-38, 60-62],
- hierarchical matrices [21, 22, 24].

1.2.1 FFT Based Methods

Covariance kernels of type (1) or (2), discretized on a uniformly spaced, regular grid, result in Toeplitz (or Block-Toeplitz) matrices. Toeplitz matrices can be embedded inside circulant matrices, for which operations such as matrix-vector products, matrix-matrix products can be efficiently computed using FFT. However, their primary deficiency is that these algorithms don't extend very easily to other types of grids that are predominant in realistic problems. In particular, Nowak and Cirpka [39] utilized this method to estimate hydraulic conductivity and dispersivities in a large-scale problem. However, they cannot easily be extended to handle non-uniform grids. In [40], they extended the FFT based algorithm to deal with irregularly spaced measurements but they did not show how to extend their algorithm for general measurement operators or the case when the underlying unknowns are not on a regular grid.

1.2.2 Fast Multipole Methods

Fast multipole methods were first proposed by Greengard and Rokhlin [37] in the context of simulating several particles with coulombic or gravitational potential, which required computing sums of the form:

$$\mathbf{q}_i = \sum_{j=1}^{N} \kappa(\mathbf{x}_i, \mathbf{y}_j)\phi_j \qquad i = 1, 2, \ldots, N \qquad (12)$$

where, $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$ are a set of N points and $\{\phi_i\}$ are a set of weights. Direct implementation of this summation can be done in $O(N^2)$. Fast multipole methods require $O(N)$ to compute $\mathbf{q}$, given a prescribed relative error and assumptions on the kernel κ. It was first proposed for the Laplacian kernel in 2-D, $\kappa(\mathbf{x}, \mathbf{y}) = -\log||\mathbf{x} - \mathbf{y}||$ but it has been subsequently extended to several other kernels, including radial basis functions. The original version of the fast multipole method relied on analytic expressions for the multipole expansion of the kernels but several kernel independent versions have been proposed [36, 38]. For more details, the reader is referred to [60-62].

1.2.3 Hierarchical Matrices

Hierarchical matrices [19, 21, 22, 24] (or $\mathcal{H}$-matrices, for short) are efficient data-sparse representations of certain densely populated matrices. The main idea that is used repeatedly in these kind of techniques is to split a given matrix into a hierarchy of rectangular blocks and approximate each of the blocks by a low-rank matrix. Hierarchical matrices have been use successfully in the approximate representation of matrices arising in the boundary element method or for the approximation of the inverse of a finite element discretization of an elliptic partial differential operator. Fast algorithms have been developed for this class of matrices, including matrix-vector products, matrix addition, multiplication and factorization in almost linear complexity.

A variant of the $\mathcal{H}$-matrix approach is the $\mathcal{H}^2$-matrix approach. The $\mathcal{H}^2$-matrix approach is an algebraic generalization of the fast multipole method [19]. The $\mathcal{H}^2$-matrix approach has also been applied to such linear inverse problems [17]. The $\mathcal{H}^2$-matrix approach enables us to further reduce the cost to $O(m)$. In this article though, we will focus our attention and explain in detail only $\mathcal{H}$-matrices. The $\mathcal{H}$-matrix approach is slightly easier to introduce to a new audience than the $\mathcal{H}^2$-matrix approach. We will, however, present some results obtained using the $\mathcal{H}^2$-matrix approach. A more detailed introduction can be found in [20].

2 HIERARCHICAL MATRICES

2.1 Low-Rank Matrices

Let $\mathbf{A}$ be any $m \times n$ matrix. The column rank of the matrix is the maximum number of linearly independent column vectors of $\mathbf{A}$, whereas the row rank is the maximum number of linearly dependent row vectors of $\mathbf{A}$. A fundamental theorem of linear algebra states that the row rank and column rank of a matrix must be equal. This number is called the "rank" of a matrix.

Let $\mathbf{A} = \mathbf{U}\boldsymbol{\sigma}\mathbf{V}^T$ be the Singular Value Decomposition (SVD) of the matrix, with $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}_n$ and $\boldsymbol{\sigma} \in \mathbb{R}^{m\times n}$ with diagonal entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}} \geq 0$ and zero non-diagonal entries. The columns of $\mathbf{U}$ form a basis for the row-space of $\mathbf{A}$, whereas the columns of $\mathbf{V}$ form a basis for the column-space of $\mathbf{A}$. For readers unfamiliar with the SVD, we recommend the following reference [41]. The SVD is a very useful idea for both theoretical analysis and numerical computations. It will also be convenient to define the following ideas. An unitarily invariant norm $||\cdot||$, is one for which $||\mathbf{UAV}|| = ||\mathbf{A}||$ for all unitary (*i.e.*, for real matrices, same as orthogonal) matrices $\mathbf{U}$ and $\mathbf{V}$. Examples of unitarily invariant norms are the spectral or 2-norm $||\mathbf{A}||_2 = \sigma_1$, where σ_1 is the largest singular value of $\mathbf{A}$ and the Frobenius norm $||\mathbf{A}||_F^2 = \sum_{i,j}|\mathbf{A}_{ij}|^2$.

We now state a theorem mentioned in [19, Section 1.1.3, page 18] that gives the best approximation to a matrix, for a prescribed rank k.

Theorem. Let $\mathbf{A}$ be a real valued $m \times n$ matrix and let $||\cdot||$ be a unitarily invariant matrix norm, then for $k \leq \min\{m,n\}$, it holds that:

$$\min_{\mathbf{M}\in\mathbb{R}^{m\times n}} \{||\mathbf{A} - \mathbf{M}|| \mid \text{rank}(\mathbf{M}) \leq k\} = ||\mathbf{A} - \mathbf{A}_k|| \qquad (13)$$

where,

$$\mathbf{A}_k = \mathbf{U}\boldsymbol{\sigma}_k\mathbf{V}^T = \sum_{i=1}^{\min\{m,n\}} \sigma_i \mathbf{u}_i\mathbf{v}_i^T \qquad (14)$$

and $\boldsymbol{\sigma}_k = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0)$. Also, $\mathbf{u}_i, \mathbf{v}_i, i = 1, \ldots, k$ are the first k columns of $\mathbf{U}$ and $\mathbf{V}$ respectively.
In other words, the best approximation of a matrix, of rank at most k is simply obtained by retaining the first k singular values and vectors of the matrix. In particular, for $\mathbf{A}_k$ as defined before.

$$||\mathbf{A} - \mathbf{A}_k||_2 = \sigma_{k+1} \qquad \text{and} \qquad ||\mathbf{A} - \mathbf{A}_k||_F^2 = \sum_{j=k+1}^{\min\{m,n\}} \sigma_j^2 \qquad (15)$$

In terms of a relative error of approximation, an alternate notion of rank is introduced [19, 42] based on a relative approximation error ε. The ε-rank of a matrix $\mathbf{A}$ in the norm $||\cdot||$ is defined as:

$$k(\varepsilon) := \min\{r \mid ||\mathbf{A} - \mathbf{A}_r|| \leq \varepsilon||\mathbf{A}||\} \qquad (16)$$

where, $\mathbf{A}_r$ is the SVD of $\mathbf{A}$ truncated to the r largest singular values.

A convenient representation of low rank matrices is the so-called outer-product form. Suppose there exist two matrices $\mathbf{M} \in \mathbb{R}^{m\times k}$ and $\mathbf{N} \in \mathbb{R}^{n\times k}$, with columns $\mathbf{m}_i$, $\mathbf{n}_i$, $i = 1, \ldots, k$ such that:

$$\mathbf{A} = \mathbf{M}\mathbf{N}^T = \sum_{i=1}^{k} \mathbf{m}_i\mathbf{n}_i^T \qquad (17)$$

One advantage of this form of representation is that the storage requirements are $k(m+n)$, as opposed to the usual mn entries of $\mathbf{A}$. This leads us to a formal definition of low-rank matrices. It should be emphasized that this definition is not the same with the usual mathematical definition of deficient rank. A matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ of rank k is called low-rank if:

$$k(m + n) \ll m \cdot n \qquad (18)$$

Moreover, low-rank matrices will always be represented in outer-product form, while the entry-wise representation will be used for all other matrices. Consider as before, $\mathbf{A} = \mathbf{M}\mathbf{N}^T$. Computing $\mathbf{y} = \mathbf{Ax}$ and $\mathbf{y} = \mathbf{A}^T\mathbf{x}$ on appropriate sized vectors $\mathbf{x}$ can be done in the following steps:

- compute $\mathbf{w} \leftarrow \mathbf{N}^T\mathbf{x}$, which costs $O(kn)$, by counting number of multiplications;
- compute $\mathbf{y} \leftarrow \mathbf{Mw}$, similarly, which costs $O(km)$.

Thus, the total cost is $O(k(m + n))$. This is to be compared with $O(mn)$ for the direct multiplication of $\mathbf{Ax}$. The same results apply when the matrix-vector multiplication involves the transpose of $\mathbf{A}$, *i.e.*, $\mathbf{y} = \mathbf{A}^T\mathbf{x}$.

2.2 Motivation and Key Ideas

We will focus our attention on a 1−D example that will help illustrate several of the features of $\mathcal{H}$−matrices.

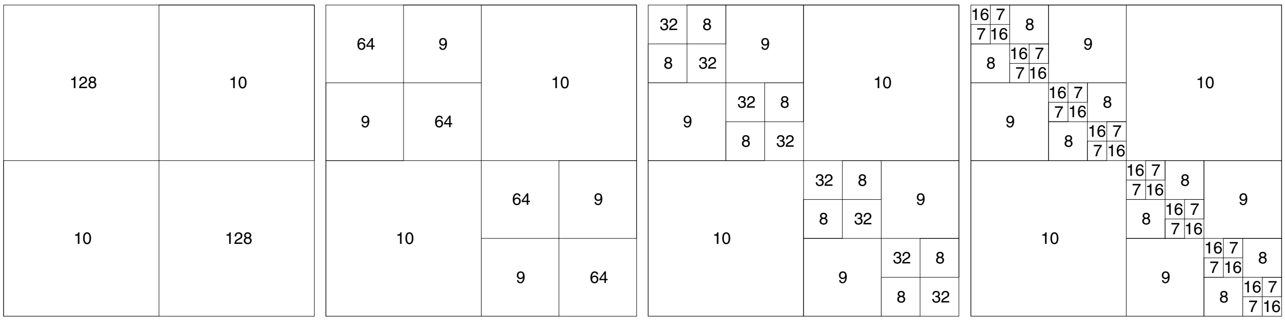

Figure 2

$k(\varepsilon)$ ranks of the sub-blocks of the matrix corresponding to $1/(|x - y| + \alpha)$ *(Eq. 19)*, with $\varepsilon = 10^{-6}$ and $m = n = 256$.

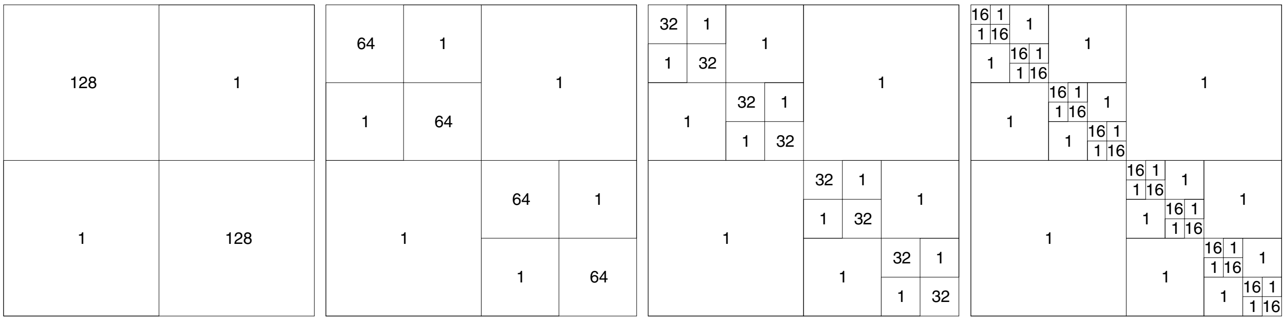

Figure 3

$k(\varepsilon)$ ranks of the sub-blocks of the matrix corresponding to $\exp(-|x - y|)$ *(Eq. 21)*, with $\varepsilon = 10^{-6}$ and $m = n = 256$.

Let us, for example, consider the kernel κ_α: $[0,1] \times [0,1] \to \mathbb{R}$ defined as:

$$\kappa_\alpha(x,y) = \frac{1}{|x-y| + \alpha}, \quad \alpha > 0 \tag{19}$$

Notice that this is an appropriate covariance function, even and positive definite (which one can verify by taking the Fourier transform). Let $\mathbf{A}$ be an $m \times n$ matrix with entries $\mathbf{A}_{ij} = \kappa_\alpha(x_i, y_j)$ for points x_i, y_j uniformly discretized in the domain $[0,1] \times [0,1]$ with $i = 1, \ldots, m$ and $j = 1, \ldots, n$ and are given by the following expressions:

$$x_i = (i-1)h_x, \ y_j = (j-1)h_y, \ h_x = \frac{1}{m-1}, \ h_y = \frac{1}{n-1} \tag{20}$$

Clearly, because this matrix is dense, storing this matrix and computing matrix-vector products are of $O(mn)$ complexity.

Figures 2 and 3 provide key insight to the structure of the matrix that we hope to exploit. Again, for the sake of illustration, we consider $m = n = 256$ (even though in actual applications where $\mathcal{H}$-matrix techniques are useful, m and n are much larger) and compute the ε-ranks of various sub-blocks of the matrix, for $\varepsilon = 10^{-6}$ and $\alpha = 10^{-6}$. The details of these computations will be discussed later, while here we focus on the results. From the first diagram in the figure, we see that the matrix has full rank, which is not surprising. In the second diagram, the block ranks of the 2×2 blocks are provided. We see that the off-diagonal blocks have low ranks whereas, the $(1,1)$ and $(2,2)$ blocks still have full rank. We subdivide these blocks further. We see that this kind of hierarchical separation of blocks and approximation of sub-blocks chosen by some criteria (in these case, simply off-diagonal sub-blocks) results in reduction in storage requirements and, as we shall see, significant reduction in computational requirements. The amount of memory (in kilo bytes) required for storing each level of approximation is as follows: 512, 296, 204 and 122. Clearly, even just using 2 levels of compression, we can reduce the storage to less than half. This represents a significant reduction in storage for problems of size $\sim 10^6$. As a second example, we consider the exponential covariance kernel, which is a part of the Matérn covariance family and is defined as:

$$\kappa_l(x,y) = \exp(-|x-y|) \tag{21}$$

As before, we plot the ε-ranks of various sub-blocks of the matrix, $\mathbf{A}$ with entries $\mathbf{A}_{ij} = \kappa(x_i, y_j)$ for $\varepsilon = 10^{-6}$. The points $x_i, i = 1, \ldots, m$ and $y_j, j = 1, \ldots, n$, for $m = n = 256$ are the same as before. However, the situation is much more dramatic.The off-diagonal blocks have ranks at most 2. Figures 2 and 3 summarize the ε-rank of the various sub-blocks for various levels of approximation.The amount of

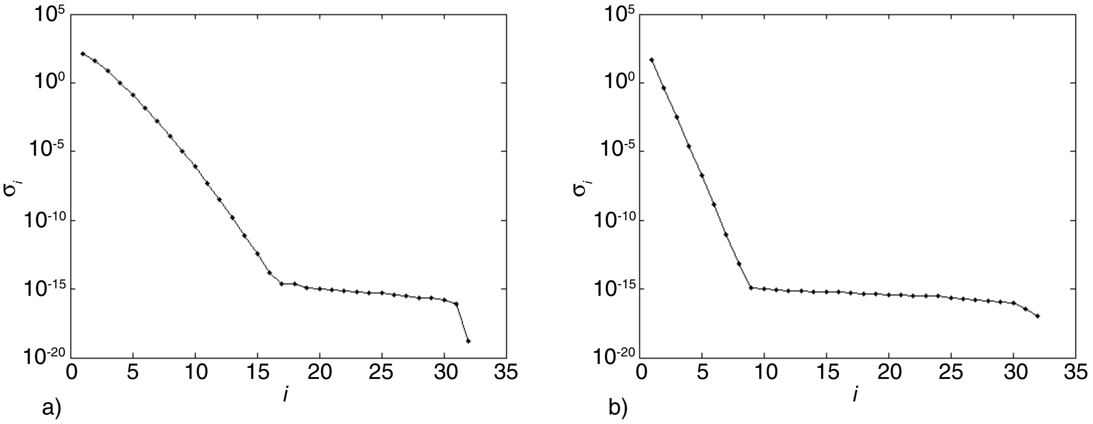

Figure 4

First 32 singular values of off-diagonal sub-blocks of matrix corresponding to non-overlapping segments with their kernel (19) a) $[0, 0.5] \times [0.5, 1]$; b) $[0, 0.25] \times [0.75, 1.0]$.

memory (in kilo bytes) required for storing each level of approximation is as follows: 512, 260, 136 and 76. At the finest level, the storage amount is less than 15% of the coarsest level of refinement. We are now in a position to summarize the key features of hierarchical matrices:

- a hierarchical separation of space;
- an acceptable tolerance ε that is specified;
- low-rank approximation of carefully chosen sub-blocks.

In the above example, the situation was rather simple because clustering of points can be easily accomplished by grouping them by intervals. In a realistic 3-D scenario, the clustering requires some slightly more complicated data structures such as a tree. The form of recursive low-rank representation of off-diagonal blocks is a special case of a class of matrices known as hierarchically semiseparable matrices (HSS) [43-45]. In the general case, submatrices corresponding to clusters that are well-separated lead to low-rank matrices. This will be discussed in detail shortly.

When we plot the singular values of off-diagonal sub-blocks of matrix corresponding to non-overlapping segments, we observe an exponential decay in the singular values, see Figure 4. The exponentially decreasing singular values, can be explained by appealing to the Taylor series representation of the kernel κ_α. Consider two sets of indices t, s, and let $X_t = \{x_i | i \in t\}$ and $X_s = \{y_j | j \in s\}$ that satisfy the following condition:

$$\min\{\text{diam}(X_t), \text{diam}(X_s)\} \leq \eta \text{ dist}(X_t, X_s),\ 0 < \eta < 1 \quad (22)$$

where we define:

$$\text{diam}(X) = \max_{x,y \in X} |x - y|, \text{dist}(X, Y) = \min_{x \in X, y \in Y} |x - y| \quad (23)$$

Now, using the Taylor series expansion for $\kappa_\alpha(x, y)$ around $\tilde{y} = \max_{X_s} y$

$$\begin{aligned} \kappa_\alpha(x, y) &= \sum_{i=0}^{\infty} \frac{(y - \tilde{y})^i}{i!} \partial_y^i \kappa_\alpha(x, \tilde{y}) \\ &= \sum_{i=0}^{k-1} (y - \tilde{y})^i (-1)^{i-1} (|x - \tilde{y}| + \alpha)^{-i-1} + R_k(x, y) \end{aligned} \quad (24)$$

where, $R_k(x, y)$ is the remainder term in the Taylor series expansion which can be bounded as follows:

$$\begin{aligned} |R_k(x, y)| &\leq \frac{1}{|x - \tilde{y}| + \alpha} \left(\frac{|y - \tilde{y}|}{|x - \tilde{y}| + \alpha} \right)^k \\ &\leq \frac{1}{|x - \tilde{y}| + \alpha} \left(\frac{\text{diam}(X_s)}{\text{dist}(X_t, X_s)} \right)^k \\ &\leq (|x - y| + \alpha)^{-1} \eta^k \end{aligned} \quad (25)$$

provided that $\text{diam}(X_s) \leq \eta \text{ dist}(X_t, X_s)$ and α small. Repeating the same argument interchanging the roles of x and y, a similar result can be obtained with $\text{diam}(X_t) \leq \eta \text{ dist}(X_t, X_s)$.

A bound of the type Equation (25) implies that with matrices $\mathbf{U}$ and $\mathbf{V}$ defined by:

$$\mathbf{U}_{il} = (|x_i - \tilde{y}| + \alpha)^{-i-1}, \quad \mathbf{V}_{jl} = (-1)^l (y_j - \tilde{y})^l, \quad i \in t, j \in s \quad (26)$$

satisfy:

$$|\mathbf{A}_{ij} - (\mathbf{U}\mathbf{V}^T)_{ij}| \leq \eta^k |\mathbf{A}_{ij}| \quad (27)$$

and therefore using the Frobenius norm $\|\mathbf{M}\|_F^2 = \sum_{i,j} \mathbf{M}_{ij}^2$, we have the following global error bound:

$$\|\mathbf{A} - \mathbf{U}\mathbf{V}^T\|_F \leq \eta^k \|\mathbf{A}\|_F \quad (28)$$

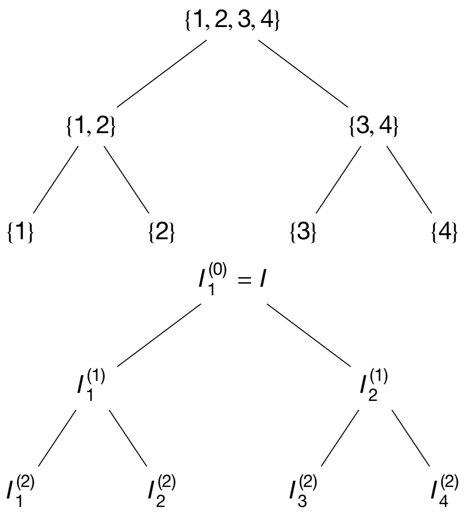

Figure 5

Simple cluster tree with 3 levels.

Center of mass of the cluster:

$$\mathbf{X} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{x}_k \in \mathbb{R}^d$$

Covariance matrix of the cluster:

$$\mathbf{C} = \sum_{k=1}^{N} (\mathbf{x}_k - \mathbf{X})(\mathbf{x}_k - \mathbf{X})^T \in \mathbb{R}^{d\times d}$$

Eigenvalues and eigenvectors:

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \qquad i = 1, \ldots, d, \qquad \lambda_1 \geq \cdots \geq \lambda_d \geq 0$$

Initialize:

$$\tau_1 := \emptyset, \tau_2 := \emptyset$$

```
for $k = 1, \ldots, N$
**if** $(\mathbf{x}_k - \mathbf{X}, v_1) \geq 0$ **then**:
    $\tau_1 := \tau_1 \bigcup \mathbf{x}_k$
**else**
    $\tau_2 := \tau_2 \bigcup \mathbf{x}_k$
**end if**
```

Figure 6

Build Cluster Tree - Split τ.

Therefore, the global relative approximation error decreases exponentially with increasing k. From this it can easily be shown that:

$$\sigma_{k+1} \leq \eta^k \|\mathbf{A}\|_F \tag{29}$$

where, σ_{k+1} is the $k+1$-th singular value of $\mathbf{A}$. This implies that the singular values of such blocks decay exponentially.

2.3 Construction of Cluster Tree and Block-Cluster Tree

Consider the index set $I = \{1, 2, \ldots, N\}$, corresponding the points $\{\mathbf{x}_i\}_{i=1}^N$. A cluster $\tau \subset I$, is a set of indices corresponding to points that are "close" together in some sense. We also define $X_\tau = \{\mathbf{x}_i | i \in \tau\}$. We have previously seen that low-rank matrices are an efficient means to represent matrices with exponentially decaying singular values. Representing the entire matrix by a low-rank matrix is not feasible in most applications. Typically, in dense covariance matrices that arise in practice, only the sub-blocks corresponding to admissible clusters (a notion that we will expand on shortly) can be approximated well by low-rank matrices. In order to partition the set of matrix indices $I \times I$ hierarchically into sub-blocks, we first recursively subdivide the index set I, which leads to the so-called cluster tree. For a formal definition, please refer to [20]. Figure 5 describes a very simple cluster tree with 3 levels. Also, the notation $I_k^{(l)}$ implies, the index set of node at level l and having index k (at level l).

A simple algorithm to construct the cluster tree is based on geometric bisection and is described in Figure 6. Briefly, this algorithm is initialized with the cluster corresponding to the index set I. The eigenvector $\mathbf{v}_1$ corresponding to the largest eigenvalue of the covariance matrix $\mathbf{C}$ is computed, and it corresponds to the direction of the longest expanse of the cluster. Then, we compute the separation plane that goes through the center of mass of the cluster $\mathbf{X}$ and is orthogonal to the eigenvector $\mathbf{v}_1$. Based on which side of the separation plane the points lie in, the cluster is split into two, more or less equal, sons. Recursive subdivision gives rise to a binary cluster tree. Theoretically, each node of the tree is split into two sons until the cardinality of the node is 1. In practice, this subdivision is stopped if the cardinality of the node is less than equal to some threshold parameter $n_{min} \sim 32$. It can be shown that for uniformly distributed clusters, the depth of the cluster tree, in other words the number of levels in the tree, is $\log_2(|I|)$. Other algorithms to construct a tree are geometric bisection [20] or the FMM cluster tree which is visualized in Figure 7.

Next, well separated cluster pairs, called *admissible* cluster pairs are identified. A cluster pair (τ, σ) is considered admissible if:

$$\min\{\mathrm{diam}(X_\tau), \mathrm{diam}(X_\sigma)\} \leq \eta \,\mathrm{dist}(X_\tau, X_\sigma) \tag{30}$$

where, the diameter of a cluster τ is the maximal distance between any pair of points in τ and the distance between two clusters τ and σ, is the minimum distance between any pair of points in $\tau \times \sigma$:

$$\mathrm{diam}(X_\tau) = \max_{\mathbf{x},\mathbf{y}\in X_\tau} \|\mathbf{x} - \mathbf{y}\|, \quad \mathrm{dist}(X_\tau, X_\sigma) = \min_{\mathbf{x}\in X_\tau, \mathbf{y}\in X_\sigma} \|\mathbf{x} - \mathbf{y}\| \tag{31}$$

A kernel is called *asymptotically smooth*, if there exist constants c_1^{as}, c_2^{as} and a real number $g \geq 0$ such that for all multi-indices $\alpha \in \mathbf{N}_0^d$ it holds that:

$$\left|\partial_{\mathbf{y}}^{\alpha} K(\mathbf{x}, \mathbf{y})\right| \leq c_1^{as} p!(c_2^{as})^p (|\mathbf{x} - \mathbf{y}|)^{-g-p}, \quad p = |\alpha| \tag{32}$$

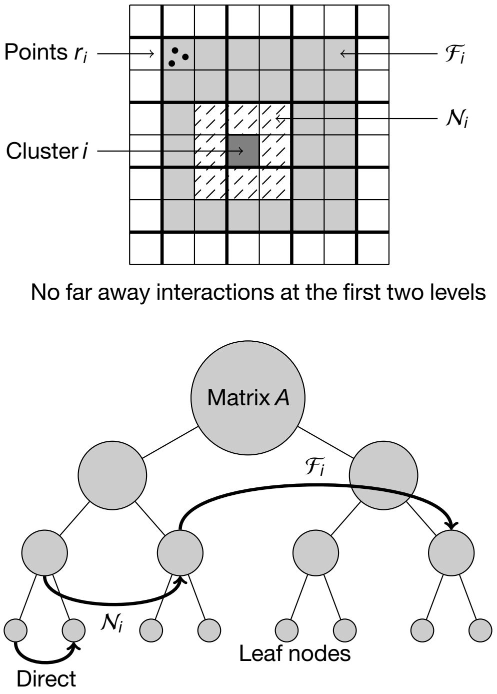

Figure 7

Fast multipole method tree.

```
if $\tau \times \sigma$ is not admissible and $|\tau| > n_{min}$ and $|\sigma| > n_{min}$
then:
    $S(\tau \times \sigma) := \{\tau' \times \sigma' : \tau' \in S(\tau), \sigma' \in S(\sigma)\}$
    for all $\tau' \times \sigma' \in S(\tau \times \sigma)$ do
        BuildBlockTree($\tau' \times \sigma'$)
    end for
else
    $S(\tau \times \sigma) := \emptyset$
end if
```

Figure 8

BuildBlockTree($\tau \times \sigma$).

The condition (30) ensures that the kernel $\kappa(\cdot, \cdot)$ is asymptotically smooth over the domain $D_\tau \times D_\sigma$, where D_σ is the convex hull of X_σ. Asymptotically smooth kernel functions can be shown to admit degenerate approximations on the kernel functions, on pairs of domains satisfying the admissibility condition (30). The implication of the kernel being asymptotically smooth on admissible clusters is that the submatrix corresponding to the blocks $\tau \times \sigma$ have exponentially decaying singular values and are well approximated by low-rank matrices. The rank of the resulting matrix is $O(k^d)$ [42], where k is the number of terms retained in the Taylor series expansion.

The cluster tree can then be used to define a block cluster tree, by forming pairs of clusters recursively. Given a cluster tree and an admissibility condition *(Eq. 30)*, the block cluster tree can be constructed using Figure 8 (see for example, Fig. 9 and 10). By calling the algorithm with $\tau = \sigma = I$, we create a block cluster tree with root $I \times I$. The leaves of the block cluster tree form a partition of $I \times I$.

For the leaves of the block-cluster tree, that are nodes which do not have any sons, if the clusters are not admissible then we store the matrix corresponding to this sub-blocks in a component-wise fashion. Else, if they are admissible, then the admissibility condition (30) guarantees that the sub-blocks will be of low-rank which can be computed in one of the methods described in the subsequent Section 2.4, and the corresponding low-rank matrices will be stored in the outer-product form described in Section 2.1.

2.4 Low Rank Approximation

We have seen in previous sections that sub-blocks of the matrices corresponding to admissible clusters can be well approximated using low-rank matrices. However, to illustrate this, we have made the assumption that it is possible to compute the Taylor series expansion of the kernel. Computing the Taylor series (or for that matter, other analytical expressions such as multipole expansions) in 3D of kernels can be a tedious task. Moreover, every time one needs to use a new covariance kernel, it might be necessary to re-derive the Taylor series for that particular kernel. Therefore, in general, we would like an approach that is, in some sense, kernel independent. By this, we mean that as long as the kernel satisfies the admissibility condition (30), the method should only require as input a numerical evaluation of the kernel $\kappa(\mathbf{x}, \mathbf{y})$ and a given pair of points $(\mathbf{x}, \mathbf{y})$. Moreoever, the number of terms required for an accurate representation up to an error of ε, is $O(k^d)$, where $k = O(1/\varepsilon)$. Another approach is to use tensor-product interpolation, which constructs a low-rank approximation using tensor-products of interpolating polynomials such as Legendre or Chebyshev polynomials [20, 36]. This approach has the advantage that it is quite general since it only uses kernel evaluations but like the Taylor series it also requires $O(k^d)$ evaluations. We now discuss a way to compute low-rank representations using adaptive cross approximation.

2.4.1 Adaptive Cross Approximation

The idea behind the cross approximation is based on the result described in [42], which states that supposing a matrix $\mathbf{A}$ is well approximated by a low-rank matrix, by a clever choice of k columns $\mathbf{C} \in \mathbb{R}^{m \times k}$ and k rows $\mathbf{R}$ of the matrix $\mathbf{A}$, we can approximate $\mathbf{A}$ of the form:

$$\|\mathbf{A} - \hat{\mathbf{A}}\| \leq \varepsilon \qquad \hat{\mathbf{A}} = \mathbf{CGR} \tag{33}$$

It relies on a result from [47], which states that if there is a sufficiently good low rank approximation to a matrix,

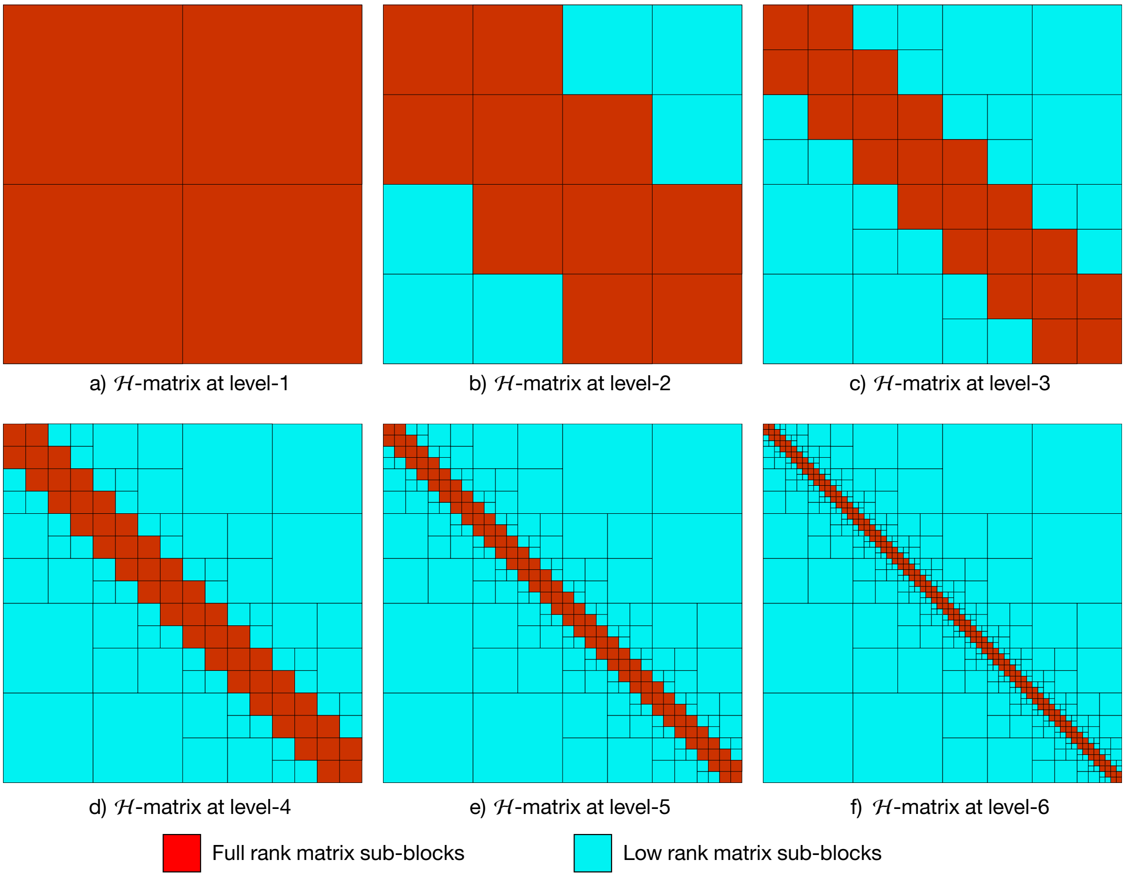
a) $\mathcal{H}$-matrix at level-1
b) $\mathcal{H}$-matrix at level-2
c) $\mathcal{H}$-matrix at level-3
d) $\mathcal{H}$-matrix at level-4
e) $\mathcal{H}$-matrix at level-5
f) $\mathcal{H}$-matrix at level-6
🟥 Full rank matrix sub-blocks
🟦 Low rank matrix sub-blocks

Figure 9

Hierarchical matrix arising out of a one-dimensional problem at different levels in the tree. The above figures were generated using HFIGS [46].

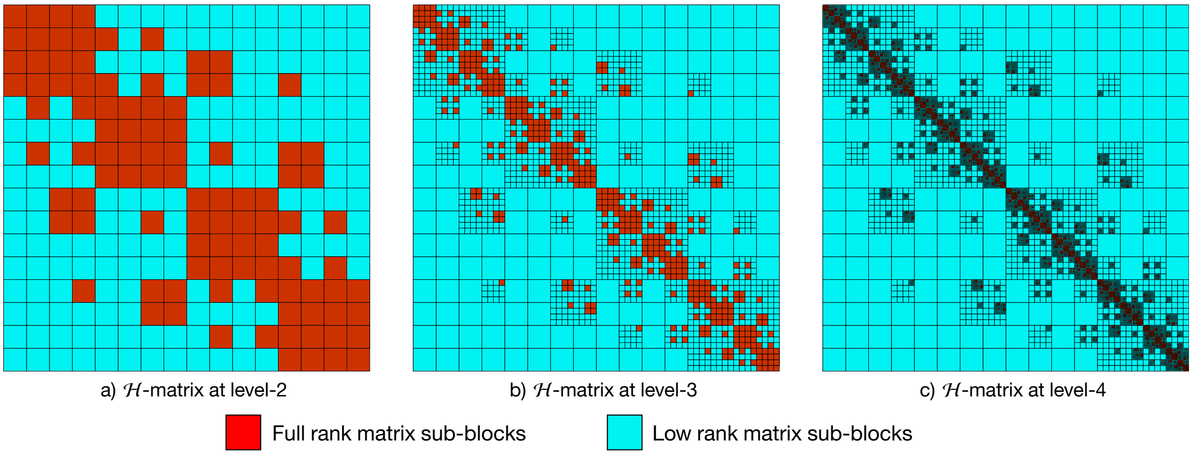
a) $\mathcal{H}$-matrix at level-2
b) $\mathcal{H}$-matrix at level-3
c) $\mathcal{H}$-matrix at level-4
🟥 Full rank matrix sub-blocks
🟦 Low rank matrix sub-blocks

Figure 10

Hierarchical matrix arising out of a two dimensional problem at different levels in the tree. The above figures were generated using HFIGS [46].

then there exists a cross-approximation with almost the same approximation quality. Figure 11 describes a simple heuristic to compute such a cross approximation that is based on successive approximations by rank–1 matrices (visualized in

```
Initialize:
        $\mathbf{R}_0 = \mathbf{A}, \qquad \mathbf{S} = \mathbf{0}$
for all $k = 0, 1, 2, \ldots$ do
    $(i^*_{k+1}, j^*_{k+1}) := \arg\max_{i,j} |(\mathbf{R}_k)_{ij}|$ and $\gamma_{k+1} = \left(\mathbf{A}_{i^*_{k+1}, j^*_{k+1}}\right)^{-1}$
    if $\gamma_{k+1} \neq 0$ then:
        Compute column $\mathbf{u}_{k+1} := \gamma_{k+1}\mathbf{R}_k\mathbf{e}_{j_{k+1}}$ and row $\mathbf{v}_{k+1} := \mathbf{R}_k^T\mathbf{e}_{i_{k+1}}$
        New residue and approximation:
            $\mathbf{R}_{k+1} = \mathbf{R}_k - \mathbf{u}_{k+1}\mathbf{v}_{k+1}^T \qquad \mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{u}_{k+1}\mathbf{v}_{k+1}^T$
    else
        Terminate algorithm with exact rank $k - 1$
    end if
end for.
```

Figure 11

Cross approximation using full pivoting.

Fig. 12). It has the property that if the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ has an exact rank $k < \min\{m, n\}$, this algorithm will terminate in k steps and defining:

$$\mathbf{S}_k = \sum_{\nu=1}^{k} \mathbf{u}_k \mathbf{v}_k^T \tag{34}$$

we have that $\mathbf{S}_k = \mathbf{A}$ in exact arithmetic. Furthermore, it exactly reproduces the k pivot rows and columns of $\mathbf{A}$. Of course, the principal disadvantage of this algorithm is that, to generate a rank–k approximation, it requires $O(kmn)$ steps, which is not feasible for large matrices. The bottleneck arises from calculating the pivot indices (i^*_k, j^*_k) which requires generating all the entries of the matrix $\mathbf{A}$.

Several heuristic strategies have been proposed to reduce the complexity of the fully pivoting cross approximation algorithm. In particular, one such algorithm is called Partially Pivoted Adaptive Cross Approximation algorithm, which maximizes $|\mathbf{A}_{ij}|$ for only one of the pairs of indices and also avoids the update of $\mathbf{A}$. It has a complexity $O(k^2(m + n))$ and is very easy to implement. Figure A.1 in the appendix, lists a practical version of the algorithm, which includes a termination criteria based on an heuristic approximation to the relative approximation in the Frobenius norm. The proofs of convergence of this algorithm can be found in [48], and they rely on approximation bounds of Lagrange interpolation and a geometric assumptions on the distribution of points, which may not be very practical. [20] lists some contrived counterexamples that show that this algorithm can produce bad pivots. To fix this issue, several other variants have been proposed such as Improved ACA (ACA+) and Hybrid Cross Approximation (HCA).

2.4.2 Further Compressing Low-Rank Representations

From Theorem 2.1, clearly the best rank k approximation to a matrix is the SVD of the matrix truncated to retain only the k largest singular values and their corresponding singular vectors. However, computing the SVD of a matrix is expensive. Indeed, for a $m \times n$ matrix $\mathbf{A}$, with $m > n$, the cost of SVD would be $O(mn^2)$. Clearly, this is an unacceptably high cost, considering that we intend to design algorithms that work in almost-linear complexity. It is for this reason that we use one of the methods described above – Adaptive Cross Approximation (ACA) or polynomial interpolation which can be computed in linear complexity. However, from Theorem 2.1, these low-rank representations are not optimal and there is further scope for compression.

The compression is performed as follows: given the low-rank outer product form for the matrix $\mathbf{A}$ of rank r, we compute the SVD of the matrix and then retain only $k(\varepsilon)$ (as defined in (16)) singular values for a given tolerance ε. We now show how to do this in an efficient manner. Given $\mathbf{A} = \mathbf{U}\mathbf{V}^T, \mathbf{U} \in \mathbb{R}^{m \times r}, \mathbf{V} \in \mathbb{R}^{n \times r}$, the truncated matrix $\tilde{\mathbf{A}}$ can be computed as follows:

- compute the truncated QR factorizations, $\mathbf{U} = \mathbf{Q}_\mathbf{U}\mathbf{R}_\mathbf{U}$ and $\mathbf{V} = \mathbf{Q}_\mathbf{V}\mathbf{R}_\mathbf{V}$;
- compute SVD $\mathbf{Q}_1\boldsymbol{\sigma}\mathbf{Q}_2^T$ of $\mathbf{R}_\mathbf{U}\mathbf{R}_\mathbf{V}^T \in \mathbb{R}^{r \times r}$;
- truncate the SVD to retain $k(\varepsilon)$ terms of $\boldsymbol{\sigma}$, and $k(\varepsilon)$ columns of $\mathbf{Q}_1$ and $\mathbf{Q}_2$;
- set $\tilde{U} = \mathbf{Q}_\mathbf{U}\tilde{\mathbf{Q}}_1\boldsymbol{\sigma}_k \in \mathbb{R}^{m \times k}$ and $\tilde{\mathbf{V}} = \mathbf{Q}_\mathbf{V}\tilde{\mathbf{Q}}_2 \in \mathbb{R}^{n \times k}$.

and now, $\tilde{\mathbf{A}} = \tilde{\mathbf{U}}\tilde{\mathbf{V}}^T$. This truncation can be computed in $O(r^2(m + n) + r^3)$.

2.5 Matrix-Vector Product

We are then in a position to define the set of hierarchical matrices with block-wise rank k.

Definition. Let $\mathcal{T}_I$ be a cluster tree and $\mathcal{T}_{I \times I}$ be the block cluster tree corresponding to the index set I. We define the set of hierarchical matrices as:

$$\mathcal{H}(\mathcal{T}_{I \times I}, k) = \{\mathbf{M} \in \mathbb{R}^{|I| \times |I|} \mid \operatorname{rank}(\mathbf{M}_{\tau \times \sigma}) \leq k \quad \forall \tau, \sigma \in \mathcal{T}_I\} \tag{35}$$

Note that $\mathcal{H}(\mathcal{T}_{I \times I}, k)$ is not a linear space. For the sub-block $\mathbf{A}_{\tau \times \sigma} \in \mathbb{R}^{\tau \times \sigma}$ which is the restriction of the matrix $\mathbf{A}$ to the sub-block $\tau \times \sigma$, and supposing that this cluster pair is admissible, *i.e.*, it satisfies condition (30), it is possible to generate a low-rank approximation for this sub-block when the kernel is asymptotically smooth (for definition see [19]). The low-rank representation for this sub-block is computed using the partially pivoted adaptive cross approximation algorithm [19, 24].

To compute the matrix-vector product involving the $\mathcal{H}-$matrix $\mathbf{A}$, with a vector $\mathbf{x}$, we use Figure 13. For a rigorous derivation of the computational complexity of storage and matrix-vector product, the reader is referred to the following sources [20,22]. Since the hierarchical matrix framework is an algebraic approach, it is possible to define a matrix arithmetics (addition, multiplication and inversion) for the hierarchical matrices. These algorithms can be found in the aforementioned references.

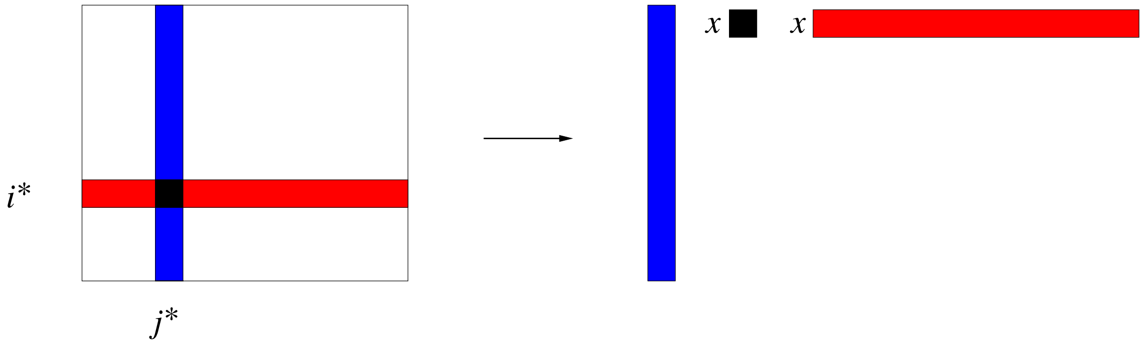

Figure 12

Illustration of the steps involved in adaptive cross approximation.

```
if $S(\tau \times \sigma) \neq 0$ then:
    for all $\tau' \times \sigma' \in S(\tau \times \sigma)$ do
        $MVM(\mathbf{A}, \tau' \times \sigma', \mathbf{x}, \mathbf{y})$
    end for
else
    $\mathbf{y}|_\tau := \mathbf{y}|_\tau + \mathbf{A}_{\tau\times\sigma} x_\sigma$, for both full and low-rank sub-blocks.
end if.
```

Figure 13

Matrix-vector product MVM($\mathbf{A}, \tau \times \sigma, \mathbf{x}, \mathbf{y}$).

3 APPLICATIONS

We now describe how to use the hierarchical matrix approach for the solution of the system of equations that result from the geostatistical approach. We then highlight the performance of the hierarchical matrices in terms of setup and computation time. The performance of our algorithm is demonstrated with an application to realtime monitoring at a CO_2 sequestration site.

3.1 Problem Specification

In this section, we describe the specific problem, which we solve by large-scale linear inversion. The problem deals with large scale crosswell tomography to monitor the CO_2 plume in CO_2 sequestration sites. The problem configuration is shown in Figure 14. The motivation for this capture geometry stems from [49]. We will compare the results we obtain using our fast large scale linear inversion algorithm with the results in [49], which were obtained by running the forward model using TOUGH2, a multiphase-flow reservoir model [50] and patchy rock physical model [51].

A cross-well tomography is setup with n_s sources along the vertical line AB and n_r receivers along the vertical line CD. The sources emit a seismic wave and the receivers measure the time taken by the seismic wave from a source to hit the receiver. This is done for each source-receiver pair.

Our goal is to image the slowness of the medium inside the rectangular domain. Slowness is defined as the reciprocal of the speed of the seismic wave in the medium. In the context of CO_2 sequestration for example, the seismic wave travels considerably slower through CO_2 saturated rock [49, 52, 53] as opposed to the rest of the medium. Hence by measuring the slowness in the medium, we can estimate the CO_2 concentration at any point in the domain (with some uncertainty) and thereby the location of the CO_2 plume.

The above configuration is typical for most of the continuous crosswell seismic monitoring sites. We go about modeling the problem as follows. As a first order approximation, the seismic wave is modeled as traveling along a straight line from the sources to receivers with no reflections/refractions. The time taken by the seismic wave to travel from the source to the target is measured. Each source-receiver pair gives us a measurement and hence there are a total of $n = n_s \times n_r$ measurements. To obtain the slowness in the domain $ABDC$, the domain is discretized into m grid points (*i.e.*, an $m_x \times m_y$ grid such that $m = m_x m_y$) and within each cell the slowness is assumed to be constant. Let t_{ij} denote the time taken by the seismic wave to travel from source i to receiver j. Let s_k be the slowness of the kth cell. For every source-receiver (i, j) pair, we then have that:

$$\sum_k l_{ij}^k s_k + v_{ij} = t_{ij} \tag{36}$$

where l_{ij}^k denotes the length traveled by the ray from source i to the source j through the kth cell and v_{ij} denotes the measurement error $i \in \{1, 2, \ldots, n_s\}$, $j \in \{1, 2, \ldots, n_r\}$, $k \in \{1, 2, \ldots, m\}$. Equation (36) can be written in a matrix-vector form:

$$\mathbf{Hs} + \mathbf{v} = \mathbf{y}$$

where $\mathbf{H} \in \mathbb{R}^{n\times m}$, $\mathbf{s} \in \mathbb{R}^{m\times 1}$, $v \in \mathbb{R}^{n\times 1}$ and $\mathbf{y} \in \mathbb{R}^{n\times 1}$. Typically, the measurement error is modeled as a Gaussian white-noise. For most problems of practical interest, the number of measurements n is much smaller than the number of unknowns m, *i.e.*, $n \ll m$. This under-determined linear system constitutes our inverse problem.

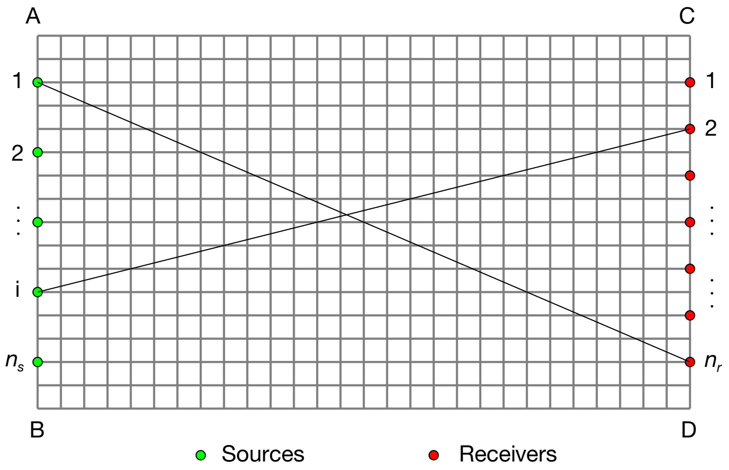

Figure 14

Capture geometry.

Operation	Cost
Compute the matrix-matrix product $\mathbf{Q}_H = \mathbf{QH}^T$	$O(nm^{3/2})$
Compute the matrix $\tilde{\Psi} = \mathbf{HQ}_H$	$O(n^2m^{1/2})$
Compute the matrix $\Psi = \tilde{\Psi} + \mathbf{R}$	$O(n)$
ompute the matrix $\phi = \mathbf{HX}$	$O(pnm^{1/2})$
Solve the linear system directly in Equation (6) to get ξ and β	$O((n+p)^3)$
Compute $\hat{s} = \mathbf{X}\beta + \mathbf{Q}_H\xi$	$O(mp + mn)$

Figure 15

Conventional technique to solve large scale linear inversion problem for the cross-well tomography problem where **H** is sparse.

We will now analyze the matrix H to figure out the structure of the linear system. Each row of the matrix H corresponds to a source-receiver pair. Consider the source i and receiver j where $i \in \{1, 2, \ldots, n_s\}$ and $j \in \{1, 2, \ldots, n_r\}$. This corresponds to the $((i-1)n_r + j)$th row of the matrix H. Since we have modeled the wave traveling from a source to a receiver as a straight line without reflections/refractions, the non-zero entries along each row correspond to the cells hit by the ray from a source to the receiver. Since the wave from the source to receiver travels along a straight line, only the cells that lie on this straight line contribute to the non-zero entries. Hence, every row of H has only $O(\sqrt{m})$ non-zeros. Hence, the matrix H is sparse since it has only $O(n\sqrt{m})$ entries as opposed to $O(nm)$ entries. We would hence like to take advantage of this sparsity to accelerate our computations. This underdetermined system is solved using the Bayesian geostatistical approach discussed in Section 3.3. Since, we would also like to characterize the fine-scale features, m can be much larger than n. The convention algorithm to solve this system is described in Figure 15. A pictorial representation of the capture geometry is shown in Figure 16.

Section 3.3 discusses the fast algorithm to solve the above problem using Bayesian geostatistical approach.

3.2 $\mathcal{H}$-matrix Approximation of Covariance Matrices

We verify the asymptotic complexity of the $\mathcal{H}$-matrix approach with regard to setup costs and computation of matrix-vector products. We focus on the following test case: from the domain $[-1, 1]^2$, we randomly sample m points which are distributed uniform randomly and for these set of points, we build the corresponding covariance matrix and compute matrix-vector products. The kernel is chosen to be a member of the Matérn covariance family, the exponential covariance kernel $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||/l)$, where l is the integral length scale and is chosen to be 1. It can be shown that this kernel satisfies the requirements of asymptotic smoothness [33] as stated in Section 2. We also make the following choices for the parameters: we pick $\eta = 0.75$ and $n_{min} = 32$. We compare the time required to perform matrix-vector products for three different tolerances $\varepsilon = 10^{-3}, 10^{-6}, 10^{-9}$. The block wise rank k is computed so that each sub-block of the matrix is approximated relatively in Frobenius norm to ε. We observe a log-linear scaling in the computation of matrix-vector products. We observe similar performance for other covariance kernels as well. The accuracy of the matrix-vector products involving an appropriate sized vector $\mathbf{x}$ can be established using the following result:

$$||\mathbf{Q}_{\mathcal{H}}\mathbf{x} - \mathbf{Q}\mathbf{x}||_2 \le \varepsilon ||\mathbf{Q}||_F ||\mathbf{x}||_2 \tag{37}$$

Various schemes to estimate the Frobenius norm of the matrix are described in [54]. For the rest of the examples, we use the exponential covariance kernel, with appropriately chosen length scales depending on the problem. We also

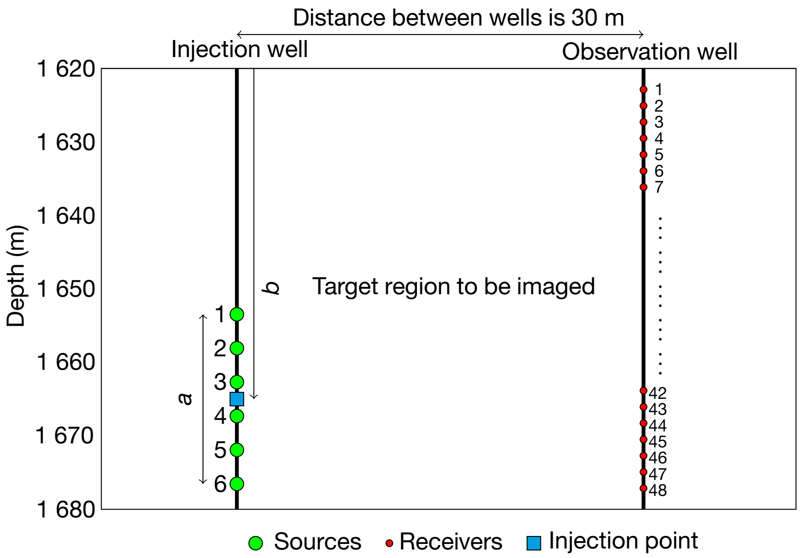

Figure 16

The crosswell geometry.

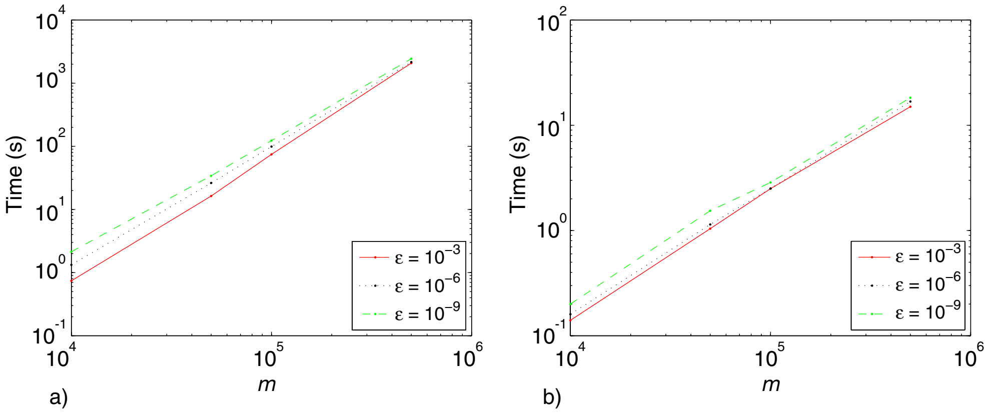

Figure 17

a) Setup time for the $\mathcal{H}$-matrix; b) time for one matrix vector product, for three different tolerances $\varepsilon = 10^{-3}, 10^{-6}, 10^{-9}$.

pick $\varepsilon = 10^{-9}$ and $\eta = 0.75$. The choice of the approximation ε depends on the application at hand. Typically, a choice of $\varepsilon = 10^{-9}$ is a good choice and ensures that the matrix-vector product are sufficiently accurate (see *Fig. 17*).

3.3 Algorithm

In this section, we discuss in detail the fast algorithm for the large scale linear inverse problem using Bayesian geostatistical approach and how this algorithm is applied to the cross-well tomography problem. We would like to point out that the algorithm is far more general and can be used for other linear inverse problems and not just the cross-well tomography case.

The main bottleneck in the entire computation is constructing the matrix product $\mathbf{Qz}$, where $\mathbf{Q} \in \mathbb{R}^{m\times m}$ is the covariance matrix. As mentioned earlier, the covariance matrix $\mathbf{Q}$ has a hierarchical matrix structure since $\mathbf{Q}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the grid points, and the interaction between the well-separated clusters can be efficiently approximated by low-rank interactions. We also take into account the sparsity of the measurement operator $\mathbf{H}$, to accelerate matrix-vector product computations.

The system of Equations (7) is solved iteratively using a Krylov subspace method such as restarted GMRES (Generalized Minimum RESidual). The key advantage of using Krylov subspace methods, is that they never require the explicit entries of the matrix but only rely on matrix-vector

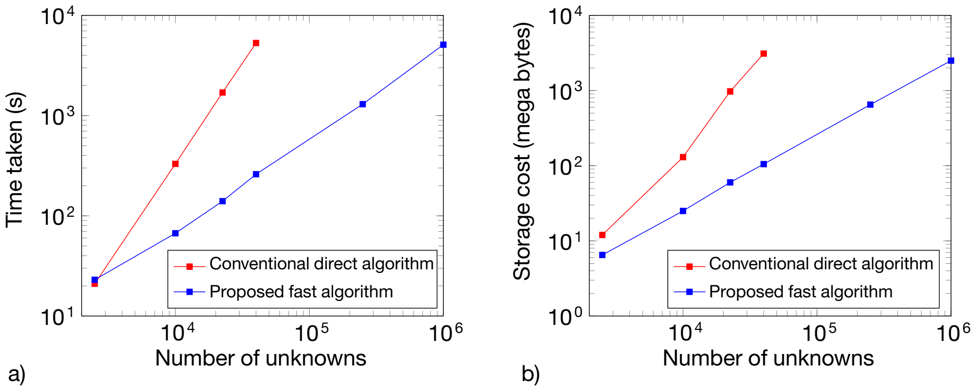

Figure 18

a) Comparison of the time taken by the fast algorithm *vs* the conventional direct algorithm; b) comparison of the storage cost.

Compute $\mathbf{z} = \mathbf{H}^T\mathbf{x}_1$ using the sparsity of $\mathbf{H}$	$O(n\sqrt{m})$
Compute $\mathbf{w} = \mathbf{Qz}$ using the $\mathcal{H}$-matrix approach	$O(km \log m)$
Compute $\mathbf{y} = \mathbf{Hw}$ using the sparsity of $\mathbf{H}$	$O(n\sqrt{m})$

Figure 19

$\mathbf{y} = \mathbf{HQH}^T\mathbf{x}_1$ using hierarchical matrix approach for the cross-well tomography problem where $\mathbf{H}$ is sparse.

products involving the matrix or its transpose. The matrix $\mathbf{A}$ is never explicitly constructed, where:

$$A = \begin{pmatrix} \mathbf{HQH}^T + \mathbf{R} & \mathbf{HX} \\ (\mathbf{HX})^T & 0 \end{pmatrix} \tag{38}$$

This is so since $\mathbf{Q}$ is never constructed explicitly. At each step in the iterative technique, we need the matrix vector product $\mathbf{Ax}$. If we set:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}, \text{ then } \mathbf{Ax} = \mathbf{A}\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{HQH}^T\mathbf{x}_1 + \mathbf{Rx}_1 + \mathbf{HXx}_2 \\ (\mathbf{HX})^T\mathbf{x}_1 \end{pmatrix} \tag{39}$$

The bottleneck in computing $\mathbf{Ax}$ is in computing $\mathbf{HQH}^T\mathbf{x}_1$. Hence, the goal is to accelerate this computation using the hierarchical matrix approach. The algorithm to accelerate the computation of $\mathbf{HQH}^T\mathbf{x}_1$ is described in Figure 19. If the iterative solver converges in κ iterations, then the total computational cost of the proposed fast algorithm is $O\left(\kappa m \log m + \kappa n\sqrt{m}\right)$.

While implementing the iterative solver, we observed that in certain instances, such as when the number of measurements are high, the number of iterations required for convergence of our system were quite large and thus, we devise a pre-conditioner to reduce the number of iterations. To deal with this, we proposed a pre-conditioner that serves to cluster the eigenvalues of the preconditioned operator near 1 thus resulting in fewer iterations. We will not describe this algorithm here. For the applications that we will consider, the number of measurements is modestly high and the algorithm converges in fewer iterations than the number of measurements, even without the use of a pre-conditioner. An alternate strategy would be to compute $\mathbf{QH}^T$ using fast matrix-vector products using the $\mathcal{H}$-matrix approach. Subsequently, we form the dense matrix $\mathbf{\Psi}$ and solve the system of Equations (7) using a direct solver such as Gaussian elimination. This strategy has an advantage when the entries of the matrix $\mathbf{H}$ are available explicitly, and the number of measurements are small. This approach has been discussed in [17].

The algorithm is implemented in C++ with the aid of PETSc [55-57], which has a collection of data structures and linear solvers that are extremely useful for large scale implementations in scientific computing.

3.4 Numerical Benchmark

In Figure 18, we show the reconstruction of the CO_2 concentration, for which the measurements are taken 5 days after the injection. The number of measurements were 288 and the number of unknowns varied up to a million. The algorithm converged in fewer than 200 iterations in all cases, for the residual to satisfy $\|\mathbf{r}_k\|_2/\|\mathbf{r}_0\|_2 \leq 10^{-6}$.

For $a = 20$ m and $b = 1\,665$ m, we plot the image of the estimated slowness. This is shown in Figure 20. There seems to be a good agreement with the true data shown in Figure 20. The uncertainty in the image reconstructed is shown in the right panel of Figure 20.

CONCLUSIONS AND DISCUSSIONS

We have described an efficient numerical method to compute the best estimate for a linear under determined set of equations using the Stochastic Bayesian approach for geostatistical applications. We emphasize here, the generality of our

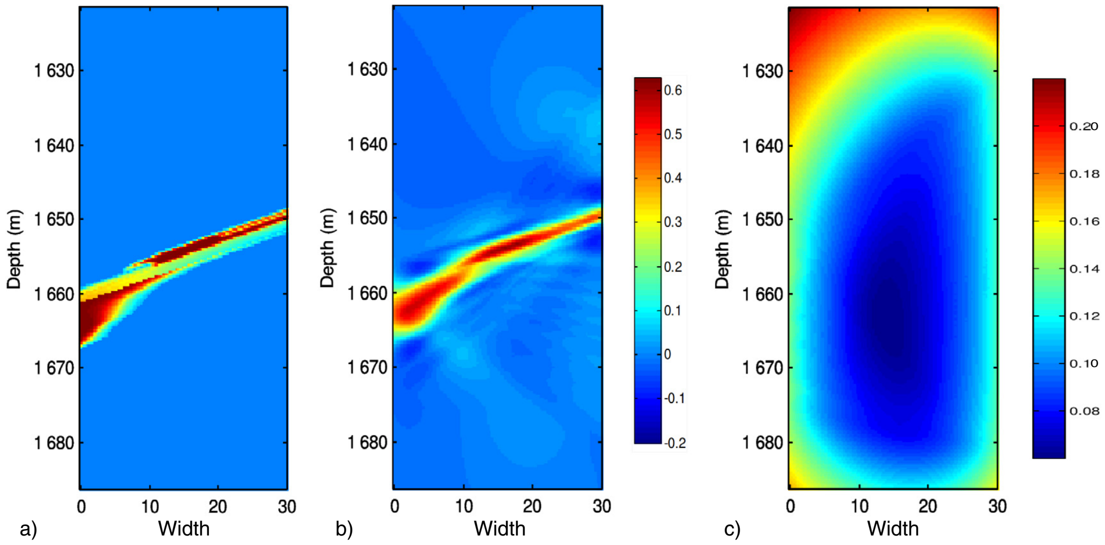

Figure 20

a) True slowness using TOUGH2/ECO2N [50, 58] simulations, 120 hours after injection; b) reconstructed slowness using our proposed algorithm for the optimal choice of *a* and *b*; c) uncertainty in the estimated solution. All these plots are for a 237 × 217 grid. Each unit of slowness corresponds to 10^4 s/m.

formulation, in the sense that it is capable of handling a wide variety of generalized covariance functions $\kappa(\cdot,\cdot)$ with a minimum amount of recoding and the fact that this is applicable to situations in which the unknowns can be distributed on an irregularly spaced grid. One important issue that has not been discussed is the quantification of uncertainty associated with the solution of the inverse problem. A commonly used strategy [14, 59] to quantify the uncertainty associated with the estimate of the solution, is *via* computing conditional realizations. This method avoids the computation of the posterior covariance matrix, which is expensive for large-scale problems [18]. An extension to solve quasi-linear problems and for solving 3D problems is underway.

ACKNOWLEDGEMENTS

The authors were supported by NSF Award 0934596, Subsurface Imaging and Uncertainty Quantification, "Army High Performance Computing Research Center" (AHPCRC, sponsored by the U.S. Army Research Laboratory under contract No. W911NF-07-2-0027) and "The Global Climate and Energy Project" (GCEP) at Stanford. The authors would also like to thank Hatef Monajemi for his help with the applications.

REFERENCES

1 Cardiff M., Barrash W. (2011) 3-D transient hydraulic tomography in unconfined aquifers with fast drainage response, *Water Resour. Res.* **47**, W12518.

2 Liu S., Yeh T.C.J., Gardiner R. (2002) Effectiveness of hydraulic tomography: Sandbox experiments, *Water Resour. Res.* **38**, 4, 5-5.

3 Liu X., Kitanidis P.K. (2011) Large-scale inverse modeling with an application in hydraulic tomography, *Water Resour. Res.* **47**, 2, W02501.

4 Zhu J., Yeh T.C.J. (2005) Characterization of aquifer heterogeneity using transient hydraulic tomography, *Water Resour. Res.* **41**, 7, W07028.

5 Berryman J.G. (2000) Analysis of approximate inverses in tomography i. resolution analysis of common inverses, *Optim. Eng.* **1**, 1, 87-115.

6 Berryman J.G. (2000) Analysis of approximate inverses in tomography ii. iterative inverses, *Optim. Eng.* **1**, 4, 437-473.

7 Lazaratos S.K., Marion B.P. (1996) Crosswell seismic imaging of reservoir changes caused by CO_2 injection, in *1996 SEG Annual Meeting*, Denver, Colorado, 10-15 Nov.

8 Daily W., Ramirez A., LaBrecque D., Nitao J. (1992) Electrical resistivity tomography of vadose water movement, *Water Resour. Res.* **28**, 5, 1429-1442.

9 Kemna A., Kulessa B., Vereecken H. (2002) Imaging and characterisation of subsurface solute transport using electrical resistivity tomography (ert) and equivalent transport models, *J. Hydrol.* **267**, 3-4, 125-146.

10 Akçelik V., Biros G., Ghattas O., Long K.R., Waanders B.B. (2003) A variational finite element method for source inversion for convective-diffusive transport, *Finite Elements Anal. Des.* **39**, 683-705.

11 Akçelik V., Biros G., Draganescu A., Hill J., Ghattas O., Waanders B.V.B. (2005) Dynamic data-driven inversion for terascale simulations: Real-time identification of airborne contaminants, *Proceedings of the ACM/IEEE 2005 conference on Supercomputing*, Washington, 12-18 Nov., 43. IEEE Computer Society.

12 Flath H.P., Ghattas O. (2011) Fast algorithms for bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial hessian approximations, *SIAM J. Sci. Comput.* **33**, 1, 407-432.

13 Michalak A.M., Kitanidis P.K. (2003) A method for enforcing parameter nonnegativity in bayesian inverse problems with an application to contaminant source identification, *Water Resour. Res.* **39**, 2, 1033.

14 Kitanidis P.K. (1995) Quasi-linear geostatistical theory for inversing, *Water Resour. Res.* **31**, 10, 2411-2419.

15 Kitanidis P.K. (2007) On stochastic inverse modeling, in *Subsurface Hydrology: Data Integration for Properties and Processes*, Hyndman D.W., Day-Lewis F.D., Singha K. (eds), American Geophysical Union (AGU), Washingtom, D.C., *Geophysical Monogr. Ser.* **171**, 19-30, doi: 10.1029/171GM04.

16 Kitanidis P.K. (2011) Bayesian and geostatistical Approaches to Inverse Problems, in *Large-Scale Inverse Problems and Quantification of Uncertainty*, Biegler L. (ed.), John Wiley and Sons, Ltd, Chichester, UK, doi: 10.1002/9780470685853.ch4.

17 Ambikasaran S., Li J.Y., Kitanidis P.K., Darve E.F. (2012) Large-scale stochastic linear inversion using hierarchical matrices. under review, *Comput. Geosci.*

18 Saibaba A.K., Kitanidis P.K. (2012) Efficient methods for large-scale linear inversion using a geostatistical approach, *Water Resour. Res.* **48**, 5, W05522.

19 Bebendorf M. (2008) *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, *Lecture Notes in Computational Science and Engineering (LNCSE)* **63**. Springer-Verlag, ISBN 978-3-540-77146-3.

20 Börm S., Grasedyck L., Hackbusch W. (2003) Hierarchical matrices, *Lecture Notes* 21/2003.

21 Börm S., Grasedyck L., Hackbusch W. (2003) Introduction to hierarchical matrices with applications, *Eng. Anal. Bound. Elem.* **27**, 5, 405-422.

22 Grasedyck L., Hackbusch W. (2003) Construction and arithmetics of h-matrices, *Computing* **70**.

23 Hackbusch W., Grasedyck L., Börm S. (2001) *An introduction to hierarchical matrices*, Max-Planck-Inst. für Mathematik in den Naturwiss.

24 Rjasanow S., Steinbach O. (2007) *The fast solution of boundary integral equations. Mathematical and Analytical Techniques with Applications to Engineering*, Springer, New York.

25 Furrer R., Genton M.G., Nychka D. (2006) Covariance tapering for interpolation of large spatial datasets, *J. Comput. Graphical Stat.* **15**, 3, 502-523.

26 Kaufman C.G., Schervish M.J., Nychka D.W. (2008) Covariance tapering for likelihood-based estimation in large spatial data sets, *J. Am. Stat. Assoc.* **103**, 484, 1545-1555.

27 Huang H.C., Cressie N., Gabrosek J. (2002) Fast, resolution-consistent spatial prediction of global processes from satellite data, *J. Comput. Graphical Stat.* **11**, 1, 63-88.

28 Johannesson G., Cressie N. (2004) Variance-covariance modeling and estimation for multi-resolution spatial models, *geoENV IV-Geostatistics for Environmental Applications*, pp. 319-330.

29 Johannesson G., Cressie N., Huang H.C. (2007) Dynamic multi-resolution spatial models, *Environ. Ecol. Stat.* **14**, 1, 5-25.

30 Cressie N., Johannesson G. (2008) Fixed rank kriging for very large spatial data sets, *J. R. Stat. Soc.: Ser. B Stat. Methodol.* **70**, 1, 209-226.

31 Christakos G. (1984) On the problem of permissible covariance and variogram models, *Water Resour. Res.* **20**, 2, 251-265.

32 Matheron G. (1973) The intrinsic random functions and their applications, *Adv. Appl. Prob.* 439-468.

33 Khoromskij B.N., Litvinenko A., Matthies H.G. (2009) Application of hierarchical matrices for computing the karhunen–loève expansion, *Computing* **84**, 1, 49-67.

34 Stein M.L. (1999) *Interpolation of Spatial Data: some theory for kriging*, Springer Verlag.

35 Matheron G. (1963) Principles of geostatistics, *Econ. Geol.* **58**, 8, 1246-1266.

36 Fong W., Darve E. (2009) The black-box fast multipole method, *J. Comput. Phys.* **228**, 23, 8712-8725.

37 Greengard L., Rokhlin V. (1987) A fast algorithm for particle simulations, *J. Comput. Phys.* **73**, 2, 325-348.

38 Ying L., Biros G., Zorin D. (2004) A kernel-independent adaptive fast multipole algorithm in two and three dimensions, *J. Comput. Phys.* **196**, 2, 591-626.

39 Nowak W., Cirpka O.A. (2006) Geostatistical inference of hydraulic conductivity and dispersivities from hydraulic heads and tracer data, *Water Resour. Res.* **42**, 8, 8416.

40 Fritz J., Neuweiler I., Nowak W. (2009) Application of fft-based algorithms for large-scale universal kriging problems, *Math. Geosci.* **41**, 5, 509-533.

41 Golub G.H., Van Loan C.F. (1996) *Matrix computations*, 3rd ed., Johns Hopkins University Press.

42 Bebendorf M. (2000) Approximation of boundary element matrices, *Numerische Mathematik* **86**, 4, 565-589.

43 Chandrasekaran S., Gu M., Lyons W. (2005) A fast adaptive solver for hierarchically semiseparable representations, *Calcolo* **42**, 3, 171-185.

44 Chandrasekaran S., Gu M., Pals T. (2006) A fast ulv decomposition solver for hierarchically semiseparable representations, *SIAM J. Matrix Anal. Appl.* **28**, 3, 603.

45 Xia J., Chandrasekaran S., Gu M., Li X.S. (2010) Fast algorithms for hierarchically semiseparable matrices, *Numer. Linear Algebra Appl.* **17**, 6, 953-976.

46 Ambikasaran S. (2012) HFIGS, URL http://www.stanford.edu/~sivaambi/Hierarchical_matrix_FIGures.html.

47 Goreinov S.A., Tyrtyshnikov E.E., Zamarashkin N.L. (1997) A theory of pseudoskeleton approximations, *Linear Algebra Appl.* **261** 1, 1-21.

48 Bebendorf M., Rjasanow S. (2003) Adaptive low-rank approximation of collocation matrices, *Computing* **70**, 1, 1-24, ISSN 0010-485X.

49 Daley T.M., Solbau R.D., Ajo-Franklin J.B., Benson S.M. (2007) Continuous active-source seismic monitoring of formula injection in a brine aquifer, *Geophysics* **72**, 5, A57.

50 Pruess K., Oldenburg C., Moridis G. (1999) TOUGH2 user's guide, version 2.0.

51 White J.E. (1975) Computed seismic speeds and attenuation in rocks with partial gas saturation, *Geophysics* **40**, 2, 224-232.

52 Daley T.M., Ajo-Franklin J.B., Doughty. C.M. (2008) Integration of crosswell CASSM (Continuous Active Source Seismic Monitoring) and flow modeling for imaging of a CO_2 plume in a brine aquifer, in *2008 SEG Annual Meeting*, Las Vegas, 9-14 Nov.

53 Doughty C., Freifeld B.M., Trautz R.C. (2008) Site characterization for CO_2 geologic storage and *vice versa*: the Frio brine pilot, Texas, USA as a case study, *Environ. Geol.* **54**, 8, 1635-1656.

54 Bradley A.M. (2011) H-matrix and block error tolerances, *Arxiv preprint arXiv:1110.2807.*

55 Balay S., Gropp W.D., McInnes L.C., Smith B.F. (1997) Efficient management of parallelism in object oriented numerical software libraries, in *Modern Software Tools in Scientific Computing*, Arge E., Bruaset A.M., Langtangen H.P. (eds), Birkhäuser Press, pp. 163-202.

56 Balay S., Buschelman K., Eijkhout V., Gropp W.D., Kaushik D., Knepley M.G., McInnes L.C., Smith B.F., Zhang H. (2008) PETSc users manual. Technical Report ANL-95/11 - Revision 3.0.0, Argonne National Laboratory.

57 Balay S., Buschelman K., Gropp W.D., Kaushik D., Knepley M.G., McInnes L.C., Smith B.F., Zhang H. (2009) PETSc Web page. http://www.mcs.anl.gov/petsc.

58 Pruess K. (2005) *ECO2N: A TOUGH2 fluid property module for mixtures of water, NaCl, and CO_2*, Lawrence Berkeley National Laboratory.

59 Zanini A., Kitanidis P.K. (2009) Geostatistical inversing for large-contrast transmissivity fields, *Stoch. Environ. Res. Risk Assess.* **23**, 5, 565-577.

60 Darve E. (2000) The fast multipole method: numerical implementation, *Journal of Computational Physics* **160**, 1, 195-240, Elsevier.

61 Darve E. (2000) The fast multipole method I: Error analysis and asymptotic complexity, *SIAM Journal on Numerical Analysis* **38**, 1, 98-128, Society for Industrial and Applied Mathematics.

62 Darve E. (1997) Fast-multipole method: a mathematical study, *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* **325**, 9, 1037-1042, Elsevier.

Final manuscript received in June 2012
Published online in November 2012

APPENDIX PARTIALLY PIVOTED ADAPTIVE CROSS APPROXIMATION

Given, the matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, the output $\mathbf{S}_k = \sum_{i=1}^{k} \mathbf{u}_i \mathbf{v}_i^T$ is a rank k approximation of $\mathbf{A}$. It does not explicitly require the entries of the matrix to be known beforehand. Figure A.1 describes the algorithm that is used in practice. Here, it should be noted that from an implementation point of view, neither $\mathbf{R}_k$, nor $\mathbf{S}_k$ are formed explicitly, their purpose is only to simplify notation. Since the matrix $\mathbf{A}$ is not completely generated, we can use the norm of the approximation $\mathbf{S}_k$ to define a stopping criterion. This norm can be computed recursively as follows:

$$||\mathbf{S}_{k+1}||_F^2 = ||\mathbf{S}_k||_F^2 + 2\sum_{j=1}^{k} \mathbf{u}_{k+1}^T \mathbf{u}_j \mathbf{v}_j^T \mathbf{v}_{k+1} + ||\mathbf{u}_{k+1}||_F^2 ||\mathbf{v}_{k+1}||_F^2 \tag{A.1}$$

and the stopping criterion for some r becomes:

$$||\mathbf{u}_r||_F ||\mathbf{v}_r||_F \leq \varepsilon ||\mathbf{S}_r||_F \tag{A.2}$$

This algorithm requires only $O(r^2(m+n))$ arithmetic operations and its memory requirement is $O(r(m+n))$. For more details, we refer the reader to the following references [24, 42, 48].

```
Initial approximation:
    $\mathbf{S} := \mathbf{0} \qquad i_1 = 1$
for $k = 0, 1, 2, \dots$ do
    1. Generation of the row:
        $\mathbf{a} = \mathbf{A}^T \mathbf{e}_{i_{k+1}}$
    2. Row of the residuum and the pivot column:
        $(\mathbf{R}_k)^T \mathbf{e}_{i_{k+1}} = a - \sum_{l=1}^{k} (\mathbf{u}_k)_{i_{k+1}} \mathbf{v}_k$
        $j_{k+1} = \arg\max_j |(\mathbf{R}_k)_{i_{k+1} j}|$
    3. Normalizing constant:
        $\gamma_{k+1} = \left((\mathbf{R}_k)_{i_{k+1} j_{k+1}}\right)^{-1}$
    4. Generation of the column:
        $\mathbf{a} = \mathbf{A}\mathbf{e}_{j_{k+1}}$
    5. Column of the residual and the pivot row:
        $\mathbf{R}_k \mathbf{e}_{j_{k+1}} = \mathbf{a} - \sum_{l=1}^{k} (\mathbf{v}_k)_{j_{k+1}} \mathbf{u}_k$
        $i_{k+2} = \arg\max_i |(\mathbf{R}_k)_{i j_{k+1}}|$
    6. New approximation:
        $\mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{u}_{k+1} \mathbf{v}_{k+1}^T$
end for.
```

Figure A.1

Partially pivoted adaptive cross approximation.