Stochastic Algorithms in Linear Algebra - beyond the Markov Chains and von Neumann - Ulam Scheme*

Karl Sabelfeld

Institute Comp. Math. & Math. Geoph., Novosibirsk, Lavrentiev str, 6, 630090 Novosibirsk, Russia
sabelfeld.karl@yahoo.de

Abstract. Sparsified Randomization Monte Carlo (SRMC) algorithms for solving systems of linear algebraic equations introduced in our previous paper [34] are discussed here in a broader context. In particular, I present new randomized solvers for large systems of linear equations, randomized singular value (SVD) decomposition for large matrices and their use for solving inverse problems, and stochastic simulation of random fields. Stochastic projection methods, which I call here "random row action" algorithms, are extended to problems which involve systems of equations and constrains in the form of systems of linear inequalities.

1 Introduction

The use of Monte Carlo methods for solving large systems of linear equations is intimately tied the Neumann-Ulam scheme, e.g., see [15], [16], [20], [37], [31], [32], [5], [6], [7]. It can be interpreted as follows: (1) first, take the representation of the solution in a form of the Neumann series, then, (2) represent the solution (one component of the vector, in the case of a system of algebraic equations $x = Ax + b$) as an expectation over some Markov chain associated in a sense to the matrix A, (3) the expectation is then calculated by taking an ensemble average (numerically, the arithmetic mean) of a random estimator defined on the constructed Markov chains.

The nice feature of this method has always its parsimonious memory usage: the method takes almost no memory, independent of the size of the matrix. However a serious drawback is its weak convergence: the error decreases as $O(N^{-1/2})$ where N is the number of independent samples of the Markov chains. Quasi-Monte Carlo methods may sometimes improve the rate of convergence, however in practice the improvement is often too small.

Nowadays, there has been dramatic progress in solving the storage problem, and it is natural to involve other stochastic ideas beyond the von

* The author thanks the organizers of the conference, and acknowledges the support of the RFBR under Grants N 06-01-00498, 09-01-12028-ofi-m, and a joint BMBF and Bortnik Funds Grant N 7326.

I. Dimov, S. Dimova, and N. Kolkovska (Eds.): NMA 2010, LNCS 6046, pp. 14–28, 2011.

Neumann-Ulam-Markov chain paradigm. As an example, we mention conventional deterministic iteration methods where however the weights are chosen at random (e.g., see [42], [36]). Another important example is the projection method where one takes projections onto randomly sampled subspaces (e.g., see [41], [33]). Sampling from random subspaces is the main idea also in the randomized singular value decomposition technique (e.g., see [18], [10]-[12]). A general idea behind these methods appeals to the fundamental result of Johnson and Lindenstrauss [21] which says that any n point subset of Euclidean space can be embedded in $k = O(\log\{n\}/\varepsilon^2)$ dimensions without distorting the distances between any pair of points by more than a factor of $(1 \pm \varepsilon)$, for any $0 < \varepsilon < 1$. So the result of Johnson and Lindenstrauss asserts that any set of n points in d-dimensional Euclidean space can be embedded into k-dimensional Euclidean space where k is logarithmic in n and independent of d so that all pairwise distances are maintained within an arbitrarily small factor. The linear transformation can be done by a random matrix whose entries are independent standard Gaussian random variables.

This transformation was essentially simplified in [1] by showing that this matrix can be changed with a matrix whose entries r_{ij} are independent discrete random variables with the distribution $P(\pm 1) = 1/6$, $P(0) = 1/3$ which greatly sparsifies the matrix. More precisely, Achlioptas' Theorem is formulated as follows.

Suppose that A is an $n \times d$ matrix of n points in $\boldsymbol{R}^d$. Fix constants $\varepsilon, \beta > 0$, and choose an integer k such that

$$k \geq \frac{4 + 2\beta}{\varepsilon^2 - \varepsilon^3/3} \log n \ .$$

Suppose that R is a random $k \times d$ matrix with entries r_{ij} belonging to the distribution

$$r_{ij} = \sqrt{3} \begin{cases} +1 & p{=}1/6 \\ 0 & p{=}2/3 \\ -1 & p{=}1/6 \ , \end{cases} \tag{1}$$

Define the $n \times k$ matrix $Q = \frac{1}{\sqrt{k}} AR$, which is considered as a projection of A onto a k-dimensional subspace. For any row $\mathbf{u}$ in A, let $f(\mathbf{u})$ be the corresponding row in Q. Then, for any distinct rows $\mathbf{u}, \mathbf{v}$ of A, we have

$$(1 - \varepsilon)||\mathbf{u} - \mathbf{v}||^2 \leq ||f(\mathbf{u}) - f(\mathbf{v})||^2 \leq (1 + \varepsilon)||\mathbf{u} - \mathbf{v}||^2$$

with probability at least $1 - n^{-\beta}$.

In [2], the authors suggested a low-distortion embedding of L_2^d into $L_p^{O(\log n)}$ $(p = 1, 2)$, called the Fast-Johnson-Lindenstrauss- Transform (FJLT). The FJLT is faster than standard random projections and just as easy to implement. It is based upon the preconditioning of a sparse projection matrix with a randomized Fourier transform.

In all this methods we deal with conventional numerical methods, but introduce some randomness to improve the convergence and more, to turn to

very high dimensions which can not be treated pure deterministically. So for instance, it is well known that the computational cost of a full SVD for large matrices is rapidly increasing with the matrix dimension. The randomized SVD solves this problem by a random sampling of small size submatrices for which the SVD is computed. In the case of projection methods, one projects the points only to a random set of subspaces. This type of methods treats the dimension problem in a non-trivial manner. But the main advantage of these methods is in their convergence rate: it is dramatically increased compared to the conventional Monte Carlo methods, and is actually comparable with the best deterministic methods.

The computational cost of most simulation algorithms in dimension m is increasing exponentially in m. Note that even simply accessing a vector in dimension m requires N^m operations, where N is the number of entries in each direction. This complexity growth is often mentioned as *Curse of Dimensionality* [4]. Given an equation in m dimensions, one can try to approximate its solution $u(x_1, \ldots, x_m)$ by a separable function, say, as $u(x_1, \ldots, x_m) \approx f_1(x_1) \ldots f_m(x_m)$, hence, radically reducing the complexity to a linear function of m. More generally, a *separation representation* is defined as [4]

$$u(x_1, \ldots, x_m) = \sum_{i=1}^{s} \lambda_i f_1^{(i)}(x_1) \ldots f_m^{(i)}(x_m) + O(\varepsilon) \ .$$

Setting an accuracy goal ε first, and then adapting $\{\lambda_i\}$, $\{f_i(x_i)\}$ and s to achieve this goal with minimal separation rank s is the idea behind many algorithms based on the separation representation approach.

In Monte Carlo methods, one often has to deal with very large dimensions, in problems like the integration, solution of integral equations, PDEs, simulation of random fields, etc. It is customary to think that the Monte Carlo methods are able to resolve problems for very high dimensions, however it is true only under the following conditions: (1) the variance of the MC estimator is small, (2) the desired accuracy is not high, (3) the complexity of construction of the random estimator is a slow function of the dimension m. Condition (3) can be often satisfied, however the conditions (1) and (2) are the main concern, because the convergence rate of MC methods is slow, scaling as $\sigma/\sqrt{M}$ where σ is the standard deviation, and M is the sample size. Therefore, any approach, method or algorithm capable to influence one of the above three conditions is of great interest in Monte Carlo methods. In particular, one often says, in a very general sense, that a *variance reduction* is developed when certain deterministic transformations lead to transformed random estimator with smaller variance. The dimension is of less concern, though the dimension reduction is desirable in relation, again, with the variance reduction. For instance, the variance is reduced if an exact (or an efficient deterministic) integration over a part of variables is possible.

In linear algebra, a fundamental approach to separation representations for matrices is based on SVD [19], see also an excellent tutorial presentation [38]. The literature on the numerical construction of SVD is vast, we mention only some

of them, e.g., [19], [24], [26], [29], [40], [44]. Recently, different matrix operations like matrix multiplication and SVD for large matrices based on randomization idea has been suggested in different papers, for different application fields, e.g., see [18], [8], [9], [10], [11], [12], [34], [13], [23], [4], [44], [25].

Where can these computational techniques be employed ? Essentially in all fields where computation is extensively used, especially when dealing with very high dimensions, such as with high-dimensional PDEs, integral equations of the 3D potential theory, inverse problems of tomography and crystallography, solving the Schrödinger equation, turbulence simulations. These techniques prove useful not only in the computational mathematics, but also problems from information retrieval and Web analysis, such as Google PageRank problem and latent semantic indexing, have strongly motivated the research in the field of design and analysis of linear algebra algorithms involving massive data sets. The list of applications can be easily extended by Data clustering, information retrieval, property testing of graphs, image processing, among others.

2 Sparsified Randomization Algorithms for Linear Systems

Let us consider a system of liner algebraic equations with a $n \times n$ matrix A,

$$\mathbf{x} = A\mathbf{x} + \mathbf{b}, \tag{2}$$

$\mathbf{x} = (x_1, \ldots, x_n)^T$, $\mathbf{b} = (b_1, \ldots, b_n)^T \in R^n$, and $A = \{A_{ij}\}_{i,j=1}^n$, where ^{T} stands for the transpose operation, and n is supposed to be large enough.

For simplicity, we assume that the spectral radius of the matrix A is less than unity, so that the solution of (2) can be calculated by the simple iteration method

$$\mathbf{x}^{(m+1)} = A\mathbf{x}^{(m)} + \mathbf{b}; \quad \mathbf{x}^{(0)} = \mathbf{b}; \quad m = 0, 1, 2, \ldots \ . \tag{3}$$

Generalizations to other iteration methods are presented in our paper [34].

Sampling of Columns without Replacement. Let G be an unbiased estimator for the matrix A which is defined as a random matrix such that $E\,G = A$, and let $G^{(0)}$, $G^{(1)}, \ldots,$ $G^{(M-1)}$ be a sequence of independent samples chosen from the random estimator G. The iterative procedure is defined by

$$\boldsymbol{\xi}^{(m+1)} = G^{(m)}\boldsymbol{\xi}^{(m)} + \mathbf{b}, \quad m = 0, 1, \ldots, M-1 \tag{4}$$

where $\boldsymbol{\xi}^{(0)} = \mathbf{b}$. Since $G^{(m)}$, $m = 0, 1, \ldots$ are all independent of each other, we get from (4) that $\mathbf{E}\boldsymbol{\xi}^{(M)} = \mathbf{x}^{(M)}$.

Let us consider the particular case when G is chosen as a sparse matrix. We will construct the matrix G column-wise: fix an arbitrary integer l which is much less than n, and choose a random set $\mathbf{J}$ of l integers uniformly from 1 to n without replacement, that is, we choose j_1 as an integer uniformly among

$1, 2, \ldots, n$, then, j_2 uniformly among the rest of $n-1$ integers, etc., the last being j_l, and define the entries of G by

$$G_{ik} = \begin{cases} \frac{n}{l} A_{ik} & \text{for} \quad k \in \mathbf{J} \\ 0 & \text{else} \end{cases}$$

for $i = 1, 2, \ldots, n$.

Thus, the random matrix G has exactly l nonzero columns of the matrix A, and obviously that for any i, k we have

$$\mathbf{E} G_{ik} = G_{ik}\ \mathbf{P}\{k \in \mathbf{J}\} = A_{ik}.$$

Note that for calculation of the components of the vector $\boldsymbol{\xi}^{(m+1)}$ we need only l components of the vector $\boldsymbol{\xi}^{(m)}$ and in order to calculate them we need only l components of $\boldsymbol{\xi}^{(m-1)}$, and so on. Consequently, we need l^2 operations in every step. For approximation of $\mathbf{x}^{(M)}$ we need NMl^2 operations, where N is the necessary statistics and M is the length of the cut-off of the Neumann series.

Non-uniform Sampling of Columns with Replacement. Let us present a different version of the sparsification algorithm, where the random choice of columns is not uniform, but it is carried out as a sampling with replacement. In addition, for generality, we describe the evaluation of AB where B is a vector or a matrix.

Starting with the remark that a product of two matrices, A and B, can be represented as follows, $AB = \sum_{\tau=1}^{n} A_{(\tau)} B^{(\tau)}$ where we use the notation $A_{(\tau)}$ for the τ-th column of A, and $B^{(\tau)}$ for the τ-th row of B we come to the randomized calculation of AB.

Let us choose a probability distribution $p_1, p_2, \ldots, p_n$ for sampling from the indices $1, 2, \ldots, n$. The randomized evaluation of the product AB is formulated as follows:

1. For $\tau = 1$ to l we sample independently a random number i_τ in $(1, \ldots, n)$ according to the probability distribution $Prob(i_\tau = k) = p_k, \quad k = 1, \ldots, n$

- a column of S is chosen as $A_{(i_\tau)}/\sqrt{l\, p_{i_\tau}}$, and the relevant row in the matrix R is taken as $B^{(i_\tau)}/\sqrt{l\, p_{i_\tau}}$.

2. The unbiased estimator for AB is the matrix SR.

The estimator SR is obviously unbiased: $E\left[(SR)_{ij}\right] = (AB)_{ij}, \quad i, j = 1, \ldots, n$.

A criterion for the best choice of the distribution $\{p_k\}$ can be of course different. It is convenient to use the mean error in the Frobenius norm, so we have to minimize the quantity $E\left[||AB - SR||_F^2\right]$.

It can be shown (see [34]) that the choice

$$p_k = \frac{|A_{(k)}|\,|B^{(k)}|}{\sum_{k=1}^{n} |A_{(k)}|\,|B^{(k)}|} \tag{5}$$

minimizes the variance of the error which takes in this case the form:

$$E\big[||AB - SR||_F^2\big] = \frac{1}{l}\Big(\sum_{k=1}^{n} |A_{(k)}|\,|B^{(k)}|\Big)^2 - \frac{1}{l}\,||AB||_F^2\ . \tag{6}$$

In conclusion we summarize that in the Sparsified Algorithm we have the following input parameters: n, the size of the matrix A, m, the number of iterations, and l, the size of the sampled submatrices which characterizes how sparse the random matrices in the randomization algorithm are.

3 SVD and Randomized Versions

3.1 SVD Background

Let A be a rectangular $m \times n$ matrix with m rows and n columns, having rank r. From the fundamental theorem of linear algebra we know (e.g., see [38]) that the matrix can be represented as a sum of r matrices of rank 1:

$$A = \sum_{i=1}^{r} \sigma_i\, u^{(i)} v^{(i)T} \tag{7}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$ are the singular values, and $u^{(i)} \in \boldsymbol{R}^m$, $v^{(i)} \in \boldsymbol{R}^n$, $i = 1, \ldots, r$ are its left and right singular column-vectors, respectively. The families $\{u^{(i)}\}$, $\{v^{(i)}\}$ are orthogonal sets of vectors: $u^{(i)T} \cdot u^{(j)} = \delta_{ij}$, and the same for $\{v^{(i)}\}$.

In matrix form, the SVD representation (7) reads:

$$A = U \Sigma V^T \tag{8}$$

where U and V are orthonormal matrices with left and right singular vectors of A, respectively, and Σ is a diagonal matrix: $\Sigma = diag(\sigma_1, \ldots, \sigma_r)$.

Recall that $U^T U = I_{r\times r}$ and $V^T V = I_{n\times n}$. The Frobenius norm $||A||_F$ and the spectral norm $||A||_2$ are defined by

$$||A||_F = \Big(\sum_{ij} a_{ij}^2\Big)^{1/2}, \qquad ||A||_2 = \max_{|x|_2=1} |Ax|_2 = \sigma_1\ . \tag{9}$$

The following fundamental result is well known from linear algebra as the *Eckhart-Young theorem* [14]. *If we are interested in the best approximation (in the norms $||\cdot||_F$ and $||\cdot||_2$) of A among all matrices D of rank k, then the solution is $A_k = \sum_{i=1}^{k} \sigma_i u^{(i)} (v^{(i)})^T$, i.e., for all k rank matrices D, $||A - A_k||_2 \leq ||A - D||_2$, $||A - A_k||_F \leq ||A - D||_F$. The matrix A_k admits the representation:*

$$A_k = U_k \Sigma_k V_k^T = A V_k V_k^T = U_k U_k^T A$$

where U_k, V_k are submatrices of U and V which contain only the top left and right singular vectors, respectively.

A matrix A has a good rank k approximation if $||A-A_k||$ is small in Frobenius and 2-norms. To estimate the errors, one may use the well known equalities:

$$||A - A_k||_F = \Big(\sum_{i=k+1}^{r} \sigma_i^2(A) \Big)^{1/2} , \quad ||A - A_k||_2 = \sigma_{k+1}(A) \ .$$

3.2 Randomized SVD Algorithm

So let us assume that the matrix A is large enough, and we want to construct a randomized approximation of the first k right singular values and corresponding right singular vectors. The idea behind many versions of randomized algorithms for SVD is to sample randomly s rows of A, then to form an $s \times s$ matrix S and compute its right singular vectors. Let us give the following version presented in [10].

Let us choose a discrete probability distribution $p_1, \ldots, p_m$ for sampling from the rows $A_{(1)}, \ldots, A_{(m)}$ of A: $\sum_{i=1}^{m} p_i = 1$.

Randomized SVD Algorithm
0. Fix an integer s such that s is much larger than k, where ε is an error measure, but $s \le m$.
1. for $j = 1$ to s do:
sample a random index $\{1, \ldots, m\}$ of the row of A according to the probability distribution $\{p_j\}_{j=1}^{m}$, and include $A_{(j)}/\sqrt{sp_j}$ as a row of S,
2. Compute $S\,S^T$ and its SVD:

$$S\,S^T = \sum_{j=1}^{s} \lambda_j^2 w^{(j)} w^{(j)T}$$

3. Compute $h^{(j)} = S^T w^{(j)}/|S^T w^{(j)}|$ for $j = 1, \ldots, k$. Construct H_k as a matrix whose columns are the $h^{(j)}$, and $\lambda_1, \ldots \lambda_k$ are our approximations to the first k singular values of A. Thus we get a rank (at most) k approximation to A is $AH_kH_k^T$.

Note that we could turn to sample columns of A instead of rows, and compute approximations of the left singular vectors, then, H_k were a matrix RR^TA where R is a $m \times k$ matrix containing approximations to the top k left singular vectors.

Let us give the error estimators presented in [12].

Assume that we construct a k rank approximation $AH_kH_k^T$ to our matrix A by the above algorithm where the sampling of s random rows is carried out according to a probability distribution $\{p_i\}_{i=1}^{m}$ satisfying the condition $p_i \ge \beta|A_{(i)}|^2/||A||_F^2$ for some positive $\beta \le 1$, and let $\varepsilon > 0$. If $s \ge 4k/\beta\varepsilon^2$ then the following estimation of the mean is true

$$E\Big[||A - AH_kH_k^T||_F^2\Big] \le ||A - A_k||_F^2 + \varepsilon||A||_F^2 \ . \tag{10}$$

Error estimation in probability is also possible. Let $\eta = 1 + \sqrt{8\log(1/\delta)/\beta}$. If $s \geq 4k\eta^2/\beta\varepsilon^2$ then with probability at least $1-\delta$

$$||A - AH_kH_k^T||_F^2 \leq ||A - A_k||_F^2 + \varepsilon||A||_F^2 \ . \tag{11}$$

The same estimations in the spectral norm hold also true, with omitting the factor k in the conditions $s \geq 4k/\beta\varepsilon^2$ and $s \geq 4k\eta^2/\beta\varepsilon^2$.

From the description of the above algorithm it is clear that the steps 1 and 2 are crucial for the efficiency of the method. In the step 1, we could of course use the uniform sampling which means, one call of the RAND generator will be used only, not depending on the dimension n. However this would work well only if the "weights" of the rows, $|A_{(i)}|$ are more or less equal for all $i = 1, \ldots, n$. Generally, according to the estimates (10), (11), it is reasonable to sample the rows according to the probability distribution $p_i = \beta|A_{(i)}|^2/||A||_F^2$. In [8], the authors suggest to use the conventional sampling algorithm which needs about $n \log n$ operations. But we can use Walker's algorithm [43] (see the Fortran code in our recent paper [34]) which even in the general case needs only one call to RAND generator, not depending on the dimension of the matrix. Out of the loop, we need only a preparation of two additional arrays of dimension n which are calculated in $O(n)$ operations. This method works of course if we use the sampling of rows with replacement which is always the case since we deal with matrices of large dimension. Thus this sampling algorithm is practically equivalent in efficiency to the uniform sampling of rows !

4 Simulation of Random Fields Based on the Karhunen-Loève Expansion

Let us now consider a real-valued inhomogeneous random field $u(x), \quad x \in G$ defined on a probability space (Ω, A, P) and indexed on a bounded domain G.

Assume (without loss of generality) that the field has a zero mean and a variance $E\,u^2(x)$ that is bounded for all $x \in G$. The Karhunen-Loève expansion has the form

$$u(x) = \sum_{k=1}^{\infty} \sqrt{\lambda_k}\,\xi_k\,h_k(x) \ , \tag{12}$$

where λ_k and $h_k(x)$ are the eigen-values and eigen-functions of the covariance function $B(x_1, x_2) = \langle u(x_1)\,u(x_2)\rangle$, and ξ_k is a family of random variables.

Thus λ_k and $h_k(x)$ are the eigen-values and eigen-functions are the solutions of the following eigen-value problem for the correlation operator:

$$\int_G B(x_1, x_2)\,h_k(x_1)\,dx_1 = \lambda_k\,h_k(x_2) \ . \tag{13}$$

The eigen-functions form a complete orthogonal set $\int_G h_i(x)\,h_j(x)\,dx = \delta_{ij}$ where δ_{ij} is the Kronecker delta-function. The family $\{\xi_k\}$ is a set of uncorrelated random variables which are obviously related to h_k by

$$\xi_k = \frac{1}{\sqrt{\lambda_k}} \int\limits_G u(x)\, h_k(x)\, dx \ , \qquad E\,\xi_k = 0, \qquad E\xi_i\,\xi_j = \delta_{ij} \ . \tag{14}$$

It is well known that the Karhunen-Loève expansion presents an optimal (in the mean square sense) convergence for any distribution of $u(x)$. If $u(x)$ is a zero mean Gaussian random field, then $\{\xi_k\}$ is a family of standard Gaussian random variables. Some generalizations to non-gaussian random fields are reported in [27].

4.1 Discrete Approximation of the Karhunen Loève Expansion

Exact solution of the eigen-value problem (13) can be obtained only for some simple cases, but generally, one has to solve it numerically, using quadrature-based methods, e.g., the Nyström method [3].

Assume for simplicity the random process $u(x)$ is defined on a bounded interval $G = [\ a, b\ ]$, and $x_i, i = 1, \ldots, n$ are points of a subdivision of this interval, and we are seeking for a discrete approximation $v \approx u(x)$ where the component v_j of the vector v approximates the value $u(x_j)$, $j = 1, \ldots, n$.

Then the covariance $n \times n$ matrix $B_v = \langle v\, v^T \rangle$ of the vector v should approximate the given correlation function $B(x_i, x_j)$ in the sense that $(B_v)_{ij} \approx B(x_i, x_j)$. This implies that the continuous eigenvalue problem (13) is approximated by the eigenvalue problem for the correlation matrix B_v:

$$B_v g_k = \lambda_k\, g_k \tag{15}$$

where λ_k are the eigenvalues, and g_k the relevant eigenvectors. Since B_v is symmetric and positive definite, all eigenvalues $\lambda_1, \ldots, \lambda_n$ are non-negative, and the spectral representation for the matrix B_v reads

$$B_v = \sum_{k=1}^{n} \lambda_k g_k g_k^T .$$

This leads us to the discrete K-L expansion of the random vector v:

$$v = \sum_{k=1}^{n} \sqrt{\lambda_k}\, \xi_k\, g_k$$

where $\{\xi_k\}_{k=1,\ldots n}$ is a sequence of independent standard Gaussian random variables.

So what remains here, is to solve the eigenvalue problem (15). If the dimension of B_h is not large, one may use standard numerical methods, e.g., the Lanczos algorithm. However to approximate random fields with high accuracy, one needs to take a subdivision which is fine enough, so the matrix B_v can be of very large size. Then, we can use the randomized low rank approximation method described in section 2.2. It should be noted that the method can be very efficient if the matrix B_v admits a good low rank approximation which is in many practical cases true when the correlation is not too long-ranged.

Lorenzian Random Field. In [34], we have presented the following results of simulation obtained by the randomized SVD based algorithm described.

Let us consider the following example [30] where we have considered the following random boundary value problem: in the upper half-plane $G = \{(x, y) : y \geq 0\}$, find a solution to the Laplace equation $\Delta u(x, y) = 0$ with the boundary conditions $u|_{y=0} = g(x)$ where $g(x)$ is a Gaussian zero mean white noise. We have constructed the solution explicitly, which says that the solution $u(x, y)$ is a partially homogeneous (i.e., homogeneous with respect to the longitudinal coordinate x) Gaussian random field which is uniquely defined by its correlations at two pints (x_1, y_1), (x_2, y_2), and the correlation function has the following Lorenzian form

$$B(x_1, y_1; x_2, y_2) = \langle u(x_1, y_1)\, u(x_2, y_2) \rangle = \frac{1}{\pi} \frac{y_1 + y_2}{(y_1 + y_2)^2 + (x_1 - x_2)^2} \,. \qquad (16)$$

Thus the random process $u(x, y)$ is inhomogeneous in transverse direction. In [30], we have found an explicit K-L expansion of this solution, so it was used to validate our randomized SVD based algorithm. The solution $u(x, y)$ on a rectangular G with a grid with 500×500 nodes was simulated, and the rank $k = 20$ approximation was already enough to calculate the solution with 1%-accuracy. The number of randomly sampled rows in the randomized SVD algorithm was $s = 200$. The reason why the rank $k = 20$ was enough is in the relative rapid decrease of the correlations. In the next example we deal with a long-range correlation function of the fractional Wiener process.

Fractional Wiener Process. Let us consider the fractional Wiener process $W^H(t)$ of index H, $H \in (0, 1)$ (Hurst parameter) which is defined as a centered Gaussian inhomogeneous random process on $[0, 1]$ with the following correlation function

$$B_H(s, t) = E[W^H(s) W^H(t)] = \frac{1}{2}\Big(s^{2H} + t^{2H} - |t - s|^{2H}\Big) \,.$$

Simulation results for the fractional Wiener process on the interval $[0, 2.5]$ with the Hurst constant $H = 0.3$ are presented in [35], the randomized SVD algorithm with $k = 80$ rank approximation was constructed by sampling 160 random rows, in the 240×240 correlation matrix.

5 Solution of Integral Equations

5.1 Singular Approximations

The low rank approximation can be used to transform the original integral equation to an equivalent integral equation with a new kernel whose properties are better in certain sense. For instance, in [31], Sect. 2.2 we present a singular approximations based method where the norm of the new kernel of the transformed equations is less than 1. This can be achieved by the randomized SVD

with very low rank approximations. Let us present the method for a system of linear algebraic equations, for details of numerical simulation see [35].

Thus we consider a large system of linear equations with an $m \times m$ matrix and right-hand side vector $b = (b_1, \ldots, b_m)^T$, and it is assumed that $||A|| \geq 1$, hence the Neumann series diverges. We introduce a matrix

$$B = A - \sum_{i=1}^{r} \alpha_i \beta_i^T \tag{17}$$

where $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_m$ are arbitrary column-vectors, i.e., the matrix B is obtained by substraction from A a sum of singular matrices of the form $\alpha_i \beta_i^T$. Suppose such matrices are found, and we are interested in the relation between the solution x and the solution of the equation with the matrix B.

Consider $r + 1$ auxiliary linear systems with the matrix B for different right-hand sides:

$$x_0 = Bx_0 + b, \quad x_1 = Bx_1 + \alpha_1, \quad \ldots\ldots \quad x_r = Bx_r + \alpha_r \ . \tag{18}$$

Then

$$x = x_0 + \sum_{i=1}^{r} J_i x_i \tag{19}$$

where $J_1, \ldots, J_r$ are components of the vector J which satisfies the equation $J = TJ + t$ where T is the matrix with entries $T_{ij} = \beta_i^T x_j, \ \ i, j = 1, \ldots r$, and t is a vector with components $t_i = \beta_i^T x_0, \ \ i = 1, \ldots, r$.

Practical implementation of this method has a sense if for small value of r we can find the expansion (17) with $q_B = ||B|| < 1$. Note that the randomized SVD algorithm suggests such a solution, and we can try, step by step, to increase the number of terms till the condition $q_B = ||B|| < 1$ is satisfied. For example, in the boundary integral equation formulation of the Laplace equation for a convex domain one may take $r = 1$ (e.g., see [17]). For non-convex domains, r can be chosen quite small, as our calculations presented in the next section show. This is true for quite general singular kernels of the potential theory which appear in the relevant boundary integral equations, see, e.g., [25], [28], [29].

5.2 Inconsistent Systems, Linear Least Squares, and Ill-Posed Problems

The general formulation of a linear least squares problem is the following: we have a set of vectors which we wish to combine linearly to provide the best possible approximation to a given vector. If the set of vectors is $\{a_1, a_2, \ldots, a_n\}$ and the given vector is b, we seek coefficients $x_1, x_2, \ldots, x_n$ which produce a minimal error $\left|b - \sum_{i=1}^{n} x_i a_i\right|$.

We have to choose the vector x so as to minimize $|Ax - b|$. Let the SVD of A be $U \Sigma V^T$ (where U and V are square orthogonal matrices, and Σ is rectangular with the same dimensions as A). Then we have

$$Ax - b = U\Sigma V^T x - b = U(\Sigma V^T x) - U(U^T b) = U(\Sigma y - c) \tag{20}$$

where $y = V^T x$ and $c = U^T b$. Note that U is an orthogonal matrix, and so preserves lengths, i.e., $|U(\Sigma y - c)| = |\Sigma y - c|$, and hence $|Ax - b| = |\Sigma y - c|$. This suggests a method for solving the least squares problem. First, determine the SVD of A and calculate c as the product of U^T and b. Then, solve the least squares problem for Σ and c, i.e., find a vector y so that $|\Sigma y - c|$ is minimal which is obviously trivial since Σ is diagonal. Now, $y = V^T x$ so we can determine x as Vy. That gives the solution vector x as well as the magnitude of the error, $|\Sigma y - c|$.

6 Random Row Action Iteration Process

We describe here a randomized version of the projection methods belonging to the class of a "row-action" methods which work well both for systems with singular matrices and for overdetermined systems. These methods belong to a type known as *Projection on Convex Sets* methods. Here we present a method beyond the conventional Markov chain based Neumann–Ulam scheme. The main idea is in a random choice of the row in the projection method so that in average, the convergence is improved compared to the conventional periodic choice of the rows. We extend this randomized method for solving linear systems coupled with systems of linear inequalities.

The row action iteration process also known as the projection method suggested first by Kaczmarz [22] can be proved to converge for any system of linear equations with nonzero rows, even when it is singular and inconsistent and the arithmetic operations required in an iteration of the method are comparatively few.

Let us consider a system of linear algebraic equations

$$Ax = b \tag{21}$$

where A is a rectangular $m \times n$ matrix with $m \geq n$, and $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$.

We further denote by a_i the i-th row of A, and a_i^T is the relevant column-vector, the transpose of a_i.

Our stochastic iterative process is written as follows

$$x_{k+1} = x_k + \omega_k \, E \, \frac{b_{\nu(i)} - (a_{\nu(i)} \cdot x_k)}{||a_{\nu(i)}||^2} \, a_{\nu(i)}^T \, , \qquad k = 1, 2, \ldots \tag{22}$$

where ω_k are some parameters (could be random), the expectation E is taken over the distribution of random indices $\nu(i)$ whose values are sampled at random among random subsets of indices lying in $(1, 2, \ldots, m)$. We show that the distribution can be chosen so that the method converges with expected exponential rate, not depending on the number of equations in the system. The solver does not even need to know the whole system, but only some random rows of the matrix, therefore, it is well suited for solving very large systems of linear algebraic equations. Moreover, this method can be used for solving systems of linear equations coupled with systems of linear inequalities. Remarkably, the structure of the algorithm remains practically the same. We note that an example of non-uniform sampling of the random rows in the row action process was suggested

in [39] which is quite costly, because it requires recalculation of the sampling probabilities in each iteration process.

So assume we solve a coupled system of linear equations and inequalities

$$a_i^T\, x \le b_i \qquad i \in I_{\le}, \tag{23}$$

$$a_i^T\, x = b_i \qquad i \in I_{=}\ . \tag{24}$$

Let

$$\gamma_k^{(i)} = \begin{cases} [(a_i \cdot x_k) - b_i]^+ & \text{if} \quad i \in I_{\le} \\ (a_i \cdot x_k) - b_i & \text{if} \quad i \in I_{=}\ , \end{cases}$$

and write the iteration process in the form:

$$x_{k+1} = x_k - \frac{\gamma_k^{(\nu(i))}}{||a_{\nu(i)}||^2}\, a_{\nu(i)}^T\ , \qquad k = 1, 2, \dots\ . \tag{25}$$

It can be shown that this process is convergent, and

$$E\big[d^2(x_{k+1}, S)\big] \le \Big(1 - \frac{1}{L^2 ||A||_F^2}\Big)\, d^2(x_k, S)\ .$$

Here L is the Hoffmann constant defined by

$$d(x, S_b) \le L||\, e(Ax - b)||$$

where S_b is the set of possible solutions of our systems, $d(x, S_b)$ is the Euclidean distance from x to the set S_b, and $e(y)$ defines the error in the relevant line of our system of equations and inequalities

$$e(y)_i = \begin{cases} y_i^+ & (i \in I_{\le}) \\ y_i & (i \in I_{=}) \end{cases}$$

References

1. Achlioptas, D., McSherry, F.: Fast computation of low rank matrix approximations. In: Proceedings of the 33rd Annual Symposium on Theory of Computing (2001)
2. Ailon, N., Chazelle, B.: The fast JohnsonLinderstrauss transform and approximate nearest neighbors. SIAM J. Comput. 39(1), 302–322 (2009)
3. Belongie, S., Fowlkes, C., Chung, F., Malik, J.: Spectral Partitioning with Indefinite Kernels Using the Nystreom Extension. In: Heyden, A., et al. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 531–542. Springer, Heidelberg (2002)
4. Beylkin, G., Mohlenkam, M.J.: Algorithms for numerical analysis in high dimension. SIAM Journal on Scientific Computing 26(6), 2133–2159 (2005)
5. Dimov, I., Philippe, B., Karaivanova, A., Weihrauch, C.: Robustness and Applicability of Markov Chain Monte Carlo Algorithms for Eigenvalue Problem. Journal of Applied Mathematical Modelling 32, 1511–1529 (2008)
6. Dimov, I., Alexandrov, V., Papancheva, R., Weihrauch, C.: Monte Carlo Numerical Treatment of Large Linear Algebra Problems. In: Shi, Y., et al. (eds.) ICCS 2007. LNCS, vol. 4487, pp. 747–754. Springer, Heidelberg (2007)

7. Dimov, I.T.: Monte Carlo Methods for Applied Scientists, p. 291. World Scientific, Singapore (2008)
8. Drineas, P., Frieze, A., Kannan, R., Vempala, S., Vinay, V.: Clustering Large Graphs via the Singular Value Decomposition. Machine Learning 56(13), 9–33 (2004)
9. Drineas, P., Kannan, R.: Pass Efficient Algorithms for Approximating Large Matrices. In: Proceedings of the 14th Annual Symposium on Discrete Algorithms (Baltimore, MD), pp. 223–232 (2003)
10. Drineas, P., Drinea, E., Huggins, P.S.: An experimental evaluation of a Monte Carlo algorithm for singular value decomposition. In: Manolopoulos, Y., Evripidou, S., Kakas, A.C. (eds.) PCI 2001. LNCS, vol. 2563, pp. 279–296. Springer, Heidelberg (2003) ISSN 0302-9743
11. Drineas, P., Kannan, R.: Fast Monte Carlo algorithms for approximate matrix multiplication. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, p. 452 (2001) ISBN: 0-7695-1390-5
12. Drineas, P., Kannan, R., Mahoney, M.W.: Fast Monte Carlo algorithms for matrices I: approximating matrix multiplication. SIAM J. Comput. 36(1), 132–157 (2006)
13. Eberly, W., Kaltofen, E.: On Randomized Lanczos Algorithms. In: International Conference on Symbolic and Algebraic Computation Archive Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation, pp. 176–183 (1997)
14. Eckhart, C., Young, G.: A principal axis transformation for non-Hermitian matrices. Bulletin of the American Mathematical Siciety 45, 118–121 (1939)
15. Ermakov, S.M., Mikhailov, G.A.: Statistical modeling. Nauka, Moscow (1982) (in Russian)
16. Ermakov, S.M.: Monte Carlo Method in Computational Mathematics. An Introductory course. BINOM publisher, St. Pitersburg (2009) (in Russian)
17. Ermakov, S.M., Sipin, A.S.: A new Monte Carlo scheme for solving problems of mathematical physics. Soviet Dokl. 285(3) (1985) (Russian)
18. Frieze, A., Kannan, R., Vempala, S.: Fast Monte Carlo algorithms for finding low-rank approximations. J. ACM 51(6), 1025–1041 (2004)
19. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
20. Hammersley, J.M., Handscomb, D.C.: Monte Carlo Methods. Chapman and Hall, London (1964)
21. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz maps into a Hilbert space. Contemp. Math. 26, 189–206 (1984)
22. Kaczmarz, S.: Angenaeherte Aufloesung von Systemen linearer Gleichungen. Bull. Acad. Polon. Sciences et Lettres, A, 355–357 (1937)
23. Kobayashi, M., Dupret, G., King, O., Samukawa, H.: Estimation of singular values of very large matrices using random sampling. Computers and Mathematics with Applications 42, 1331–1352 (2001)
24. Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of Research of the National Bureau of Standards 45(4), 255–282 (1950)
25. Liberty, E., Woolfe, F., Martinsson, P.-G., Rokhlin, V., Tygert, M.: Randomized algorithms for the low-rank approximation of matrices. Yale Dept. of Computer Science Technical Report 1388
26. Muller, N., Magaia, L., Herbst, B.M.: Singular Value Decomposition, Eigenfaces, and 3D Reconstructions. SIAM Review 46(3), 518–545 (2004)

27. Phoon, K.K., Huang, H.W., Quek, S.T.: Simulation of strongly non-Gaussian processes using Karhunen-Loeve expansion. Probabilistic engineering Mechanics 20, 188–198 (2005)
28. Rokhlin, V.: Rapid solution of integral equations of classical potential theory. J. Comp. Phys. 60, 187–207 (1985)
29. Rokhlin, V., Szlam, A., Tygert, M.: A randomized algorithm for principal component analysis. SIAM J. Matrix Anal. Appl., arxiv.org (2009)
30. Expansion of random boundary excitations for some elliptic PDEs. Monte Carlo Methods and Applications 13(5-6), 403–451 (2007)
31. Sabelfeld, K.K.: Monte Carlo Methods in Boundary Value Problems. Springer, Heidelberg (1991)
32. Sabelfeld, K.K., Simonov, N.A.: Random Walks on Boundary for Solving PDEs. VSP, The Netherlands, Utrecht (1994)
33. Sabelfeld, K., Loshina, N.: Fast stochastic iterative projection methods for very large linear systems. In: Seventh IMACS Seminar on Monte Carlo Methods (MCM 2009), Brussels, September 6-11 (2009)
34. Sabelfeld, K., Mozartova, N.: Sparsified Randomization Algorithms for large systems of linear equations and a new version of the Random Walk on Boundary method. Monte Carlo Methods and Applications 15(3), 257–284 (2009)
35. Sabelfeld, K., Mozartova, N.: Sparsified Randomization Algorithms for low rank approximations and applications to integral equations and inhomogeneous random field simulation. Mathematics and Computers in Simulation (2010) (submitted)
36. Sabelfeld, K., Shalimova, I., Levykin, A.: Random Walk on Fixed Spheres for Laplace and Lamé equations. Monte Carlo Methods and Applications 12(1), 55–93 (2006)
37. Sobol, I.M.: Numerical Monte Carlo Methods. Nauka, Moscow (1973) (in Russian)
38. Strang, G.: The fundamental Theorem of linear algebra. The American Mathematical Monthly 100(9), 848–855 (1993)
39. Strohmer, T., Vershynin, R.: A randomized Kaczmarz algorithm with exponential convergence. Journal of Fourier Analysis and Applications 15, 262–278 (2009)
40. Stewart, G.W.: On the Early History of the Singular Value Decomposition. SIAM Review 35(4) (1993)
41. Vempala, S.S.: The Random projection method. AMS (2004)
42. Vorobiev, Ju.V.: Stochastic iteration process. J. Comp. Math. and Math. Physics 4(6), 5(5), 1088–1092, 787-795 (1964) (in Russian)
43. Walker, A.J.: New fast method for generating discrete random numbers with arbitrary friquency distributions. Electronic Letters 10, 127–128 (1974)
44. Woolfe, F., Liberty, E., Rokhlin, V., Tygert, M.: A fast randomized algorithm for the approximation of matrices. Applied and Computational Harmonic Analysis 25, 335–366 (2008)