INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 15, Number 3, Pages 340–352

AN APPROXIMATE ALGORITHM TO SOLVE LINEAR SYSTEMS BY MATRIX WITH OFF-DIAGONAL EXPONENTIAL DECAY ENTRIES

QIANGSHUN CHANG, YANPING LIN, AND SHUZHAN XU

Abstract. We present an approximate algorithm to solve only one variable out of a linear system defined by a matrix with off-diagonal exponential decay entries (including the practically most important class of band limited matrices) via a sub-linear system. This approach thus enables us to solve any subset of solution variables. Parallel implementation of such approximate schemes for every variable enables us to solve the linear system with computational time independent of the matrix size.

Key words. Linear equation, numerical solution, sub-linear system, decomposition.

1. Introduction and motivations

How to solve a large linear system $Ax = f$ is a key topic of practical importance. Direct approaches (like Gauss elimination and various decomposition schemes, such as LU and QR, are used mainly for small matrices) are theoretically precise yet practically forbidden for large linear systems in general due to computational cost. Iterative approaches (such as Jacobi, Gausss-Seidel, SOR and CG iteration, [8, 19]) and multigrid methods [17] are approximate methods and basically "the practical schemes". For iterative methods, the matrix condition number (defined as $cond(A) = \max_{||x||_2=1} \frac{||Ax||_2}{\|x\|_2} = \frac{\lambda_{max}}{\lambda_{min}}$, where λ_{max} and λ_{min} are the maximum and the minimum eigen-values of A respectively) usually decides their convergence rates and the matrix "sparsity" (i.e. number of non-zero entries in A) decides their computation costs. Iteration by nature, the convergence speed of multigrid methods is independent of the matrix condition numbers though [12, 17]. For linear systems derived from numerical differential equations via finite difference [16] or finite element [14], the matrix sizes are generally decided by the size of domain and the approximation accuracies required. Domain decomposition ([12, 13], to split the original problem into problems with smaller domains) and preconditioning ([11], to transform the matrix for better condition number) studies are trying to deal with large matrices and poor matrix condition numbers. They are usually "geometrical" methods linked to the original problem. Algebraic multigrid, a special iterative method [17], is based on the algebraic properties of the final linear systems mainly.

Heading in a different direction, there are also lot of recent progress trying to reduce the number of computations and to split the matrices (i.e. different decomposition schemes). Almost purely algebra in nature, these works try to solve the linear system efficiently by looking at the matrix structure directly. The first approach is the work on semiseparable matrices (also been used for symmetric eigen-value problems) [18]. A symmetric matrix can be transformed into tridiagonal, semiseparable or with diagonal plus semiseparable form (free diagonal choice) via orthogonal similarity or Lanczos-like reduction. Efficient algorithms can then be

Received by the editors March 10, 2016 and, in revised form, May 20, 2017.

2000 *Mathematics Subject Classification.* 15XX, 66XX.

devised accordingly [18]. For example, Crout algorithm can be applied to solve the tridiagonal form and efficient QR-factorization approach can be used to solve the semiseparable form. Another approach is H-matrices [2] with the aim of enabling matrix operations in almost linear complexity. The key technique is to applying local matrix approximation via matrices that is product of two vectors. Based on Taylor series analysis of kernel $\log|x-y|$, local matrix approximation can be applied with low rank approximation of matrix blocks. Almost linear complexity algorithm can then be devised via cluster tree partition technically. We emphasize that these techniques, easier to apply with efficiency for small matrices, can be applied on top of our scheme since our approach is to decompose the system into smaller systems first. Future work combining these ideas with decomposition most likely will further refine our algorithm.

Last but not least, we mention algebraic Schwarz or algebraic domain decomposition methods which are mostly related to our work [1, 15, 20]. They are iterative approaches via Schwarz alternating in algebraic form and can actually be viewed more clearly in its elliptic problem theoretical background. H. A. Schwarz's study of Dirichlet problem on overlapping regions provided the fundamental alternating solution approach (numerically a different way of iteration). The elegant and insightful analysis of P. L. Lions and O. B. Widlund [5, 9, 10] are recent reinterpretations and further developments (for example parallel algorithms) of this classical direction. Amazingly enough, our very first feeling is that these projection analysis techniques might be borrowed and modified for the iterative turbo decoding analysis. It is also interesting to recall that turbo codes was invented by C. Berrou, A. Glavieux and P. Thitimajshima in 1993 from France (see references in [7]). Secondly and most importantly, we feel that the connection between iterative algebraic domain decomposition and our direct approach to be presented deserves serious further investigation (in particular the Dirichlet problem counter part analysis).

In our effort starting with algebraic multigrid looking for schemes to make the matrix to have better condition number and to decompose large linear systems, we found a fast approximate algorithm capable of breaking the matrix size (for special classes of matrices of course) to be presented here. This algorithm seems can be used in many areas beyond numerical partial differential equations. It thus justifies an independent paper. Our main contribution is the algorithm capable of solving a single variable by solving a smaller linear system in some special large linear systems with controllable error. Even can be further elaborated and extended, we mainly study matrices with off-diagonal exponential decay entries for simplicity and practical efficiency. Counter examples show easily that our algorithm is not valid for all matrices. For practical implementation concerns, we also present exact conditions for a matrix to be with off-diagonal exponentially decay entries.

Let us look at some simple symmetric positive definite matrix examples with exponential decay entries to build up our intuitions for further analysis. For $A = \left(\begin{smallmatrix} a & c \\ c & d \end{smallmatrix}\right)$, where $a > 0$, $d > 0$, $a >> c$, and $d >> c$. Linear equation $Ax = b_1$ has solutions $x = (db_1 - cb_2)/(ad - c^2)$ and $x_2 = (ab_2 - cb_1)/(ad - c^2)$. We can see that $\lim_{c \to 0} x_1 = \dfrac{b_1}{a}$ and $\lim_{c \to 0} x_2 = \dfrac{b_2}{d}$. That is solutions of $\left(\begin{smallmatrix} a & c \\ c & d \end{smallmatrix}\right)\left(\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right) = \left(\begin{smallmatrix} b_1 \\ b_2 \end{smallmatrix}\right)$ are close to solutions of $\left(\begin{smallmatrix} a & 0 \\ 0 & d \end{smallmatrix}\right)\left(\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right) = \left(\begin{smallmatrix} b_1 \\ b_2 \end{smallmatrix}\right)$, which is a compressed form with c set to zero. Let's ponder on this observation and extend our analysis to a larger matrix. Suppose $A = \left(\begin{smallmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{smallmatrix}\right)$ is symmetric positive definite and with off-diagonal exponential decay entries, we have (from the explicit expression of $Ax = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ solutions) the following approximations

$$x_1 \approx \frac{a_{33} \begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{a_{33} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}, \tag{1}$$

$$x_3 \approx \frac{a_{11} \begin{vmatrix} a_{22} & b_2 \\ a_{32} & b_3 \end{vmatrix}}{a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}} = \frac{\begin{vmatrix} a_{22} & b_2 \\ a_{32} & b_3 \end{vmatrix}}{\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}}. \tag{2}$$

As for the second variable, we have the following approximation form

$$x_1 \approx \frac{-a_{21} \begin{vmatrix} b_1 & a_{13} \\ b_3 & a_{33} \end{vmatrix} + b_2 \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}}{a_{22} \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} - a_{23} \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{33} \end{vmatrix}}. \tag{3}$$

Equivalently, the approximate solution can be given by the following systems: (1) use $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ to solve x_1, (2) use $\begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_2 \\ b_3 \end{pmatrix}$ to solve x_3, and (3) use system $\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ to solve x_2. What we have observed is that matrix $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ can be approximated by $\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix}$. The solutions of the linear system $\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ can be approximated by the three sub-linear systems. Each corresponds to a sub-block of the original matrix: the upper-left corner matrix $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, matrix $\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix}$ in the middle, and the lower-right corner $\begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix}$ with the corresponding b_i's to solve x_i's respectively. Observing from $\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, we see $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, $\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ and $\begin{pmatrix} b_2 \\ b_3 \end{pmatrix}$ are used respectively. It is actually using a ”window” of unit length from (or centered at) the variable to be solved. Just look at variable indexes, we see they correspond to $\{\underline{1}, 2\}, \{1, \underline{2}, 3\}$ and $\{2, \underline{3}\}$ respectively to be precise. The most noticeable fact is that this approach actually reduces the solution spaces dimensions with tolerable errors. The previous observations are also true for non-symmetric matrices. We ignore the analysis details to avoid lengthy expressions.

For a linear system defined by matrix with off-diagonal exponential decay entries, we can simply approximate the first a few rows with the ”tail” part (looking from left to right) off as zeros. This approximation will not affect the solution much due to the limit properties. Yet it is enough to solve the first variable already. If we keep applying this ”windowing” technique to each variable of the system, the solutions will actually be quite close to the original precise solutions. Due to the edge effect (errors caused by the variables near the window edge), only the solution variable at the window center is accurate enough. We will show that this intuition is actually right.

Another important source of our intuition comes also from the approximate Viterbi, BCJR and turbo decoding algorithms (also named as local decoding schemes by Xu and Stark in 3-4 of [7]) for convolutional and turbo codes studied and summarized in [7]. Viterbi, BCJR and turbo algorithms are fundamental schemes for bit error correction in digital communications (see references in [7]). These approximate numerical schemes, virtually "windowing techniques" using lower dimension approximations, have important practical values. With parallel lay out of these approximate decoders, fast algorithms can be devised with speed that is independent of frame size [7]. For the theoretically and practically fundamental linear codes (include convolutional, turbo and LDPC codes) in digital communications, the decoding methods (include Viterbi, BCJR, turbo and LDPC algorithms) are also consists of direct and iterative schemes and the similarities with what have been studied in numerical analysis is unmistakable. We feel that the similarity and hidden connections between them are interesting and worth serious further investigations.

Let's recall some results regarding inverses of band limited matrices. Stephen Demko et al have shown that the inverse of a band limited matrix (with off-diagonal exponential decay entries) is with off-diagonal exponential decay entries under certain conditions [3, 4] and have derived the following results (we cite them as lemmas here for further discussions). We will see that the extension of these results is the very foundation of our key algorithm to be introduced.

Lemma 1.1. *Let $A = (a_{i,j})$ be an $n \times n$ matrix. Assume that there is a number m such that $a_{i,j} = 0$ if $|i-j| > m$ (called m banded) and that $\|A\|_q \le 1$ and $\|A^{-1}\|_q \le \mu^{-1}$ for some $1 \le q \le \infty$ and some $\mu > 0$. Then, with $A^{-1} = (\alpha_{i,j})$, there are numbers $K > 0$ and $r \in (0,1)$ depending only on μ and m such that $|\alpha_{i,j}| \le Kr^{-|i-j|}$ for $\forall\, i,\, j$.*

Lemma 1.2. *Let A and A^{-1} be in $B(l^2(S))$. Then if A is positive definite and m banded, we have $|A^{-1}(i,j)| \le C\lambda^{-|i-j|}$ where $C = \left\|A^{-1}\right\| \left\{1, \frac{(1+\sqrt{cond(A)})^2}{2cond(A)}\right\}$ and $\lambda = \left(\frac{\sqrt{cond(A)}-1}{\sqrt{cond(A)}+1}\right)^{\frac{2}{m}}$. If A fails to be positive definite but is still m banded, bounded and bounded invertible then $|A^{-1}(i,j)| \le C_1\lambda_1^{-|i-j|}$ where $\lambda_1 = \left(\frac{\sqrt{cond(A)}-1}{\sqrt{cond(A)}+1}\right)^{\frac{1}{m}}$ and $C_1 = (m+1)\lambda_1^{-m}\left\|A^{-1}\right\| cond(A) \max\left\{1, \left[\frac{1+cond(A)}{cond(A)}\right]^2/2\right\}$.*

Lemma 1.3. *If A and A^{-1} be in $B(l^2(S))$. Then if A is positive definite and m banded set $\lambda = \left(\frac{\sqrt{cond_p(A)}-1}{\sqrt{cond_p(A)}+1}\right)^{2/m}$. For any $\gamma > \lambda$ there is a constant $C_2 = C_2(\gamma, A)$ so that $|A^{-1}(i,j)| \le C_2\gamma^{-|i-j|}$. If A fails to be positive definite but is quasi-centered, m bounded and bounded invertible set $\lambda_1 = \left(\frac{\sqrt{cond_p(A)}-1}{\sqrt{cond_p(A)}+1}\right)^{1/m}$. For any $\gamma > \lambda_1$ there is a constant $C_3 = C_3(\gamma, A)$ so that $|A^{-1}(i,j)| \le C_3\gamma^{-|i-j|}$.*

2. Matrix with Off-diagonal Exponential Decay Entries

For most practical linear systems, matrix A is with good properties. These properties are usually symmetric, positive definite, and with off-diagonal exponentially decay entries. We may also assume the matrix condition number is good. As a matter of fact, efficient numerical schemes typically come only with these stringent conditions in general.

We now look at a large class of matrices with off-diagonal exponential decay entries. It is interesting to see that the same exponential decay property (actually with the same decay ratio) is also true for their inverses as stated in the following theorem. We name it after Stephen Demko to memorize his pioneering effort three decades ago [3, 4].

Theorem 2.1. *(Demko Lemma) If matrix A has off-diagonal exponentially decay entries, $|a_{i,j}| \leq \alpha \cdot e^{-p|j-i|}$ for $\forall\, j$ and $i = 0, \dots, n-1$, $\alpha > 0$, $\rho > 0$, its inverse $B = A^{-1}$ is also with off-diagonal exponentially decay entries, $|b_{i,j}| \leq \beta \cdot e^{-\lambda|j-i|}$ for $\forall\, j$ and $i = 0, \dots, n-1$, $\beta > 0$, $\lambda > 0$, with $\beta = \frac{n^2\alpha^{n-1}}{2\det(A)(1-e^{-\rho})}$ and $\lambda = \rho$. In Particular, both A and $B = A^{-1}$ have the same exponential decay ratio.*

We first prove the following lemma on matrix determinants.

Lemma 2.2. *If matrix A has off-diagonal exponentially decay entries, $|a_{i,j}| \leq \alpha \cdot e^{-\rho|j-i|}$, we have $|\det(A)| \leq \frac{(n+1)^2}{2}\alpha^n\frac{1}{1-e^{-\rho}}$. Denote M_{ij} as the $(n-1)\times(n-1)$ matrix derived by eliminating the i-th row and j-th column from the original matrix A . We then have $|\det(M_{ij})| \leq \frac{n^2}{2}\alpha^{n-1}\frac{e^{-|i-j|\rho}}{1-e^{-\rho}}$.*

Proof. We prove the first claim by direct determinant calculation. Under the set equality sense and with $\{j_1, j_2, \dots, j_n\} = \{1, 2, \dots, n\}$, we have $\det(A) = \sum \pm a_{1j_1} \dots a_{nj_n}$. Clearly, $|\det(A)| \leq \sum |a_{1j_1} \dots a_{nj_n}|$. Define $d_{\{j_1, j_2, \dots, j_n\}} = \sum\limits_{i=1,2,\dots,n} |i - j_i|$, by the given assumptions, $|a_{1j_1} \dots a_{nj_n}| \leq \alpha^n e^{-d_{\{j_1, j_2, \dots, j_n\}}\rho}$ and $|\det(A)| \leq \sum |a_{1j_1} \dots a_{nj_n}| \leq \alpha^n \sum e^{-d_{\{j_1, j_2, \dots, j_n\}}\rho}$. Let's look at all the $n!$ terms of $\{j_1, j_2, \dots, j_n\}$. There is one term with $d_{\{j_1, j_2, \dots, j_n\}} = 0$, and son on. Each of them corresponds to permutation $\left(\begin{smallmatrix}1, 2, \dots, n\\ j_1, j_2, \dots, j_n\end{smallmatrix}\right)$, and all of the permutations form a permutation group. Any permutation besides $I = \left(\begin{smallmatrix}1, 2, \dots, n\\ 1, 2, \dots, n\end{smallmatrix}\right)$ swaps at least two positions and thus $d_{\{j_1, j_2, \dots, j_n\}} \geq 2$ if it is not zero. Actually, $d_{\{j_1, j_2, \dots, j_n\}} = 2$ only happens when two adjacent positions get exchanged. In other words, one exchange (formally named transposition) will cause $d_{\{j_1, j_2, \dots, j_n\}}$ increase at least by 2. There are at most $\lfloor \frac{n}{2} \rfloor$ pairs of positions to swap in the sequence of $\{1, 2, \dots, n\}$. To reverse the reasoning, for $d_{\{j_1, j_2, \dots, j_n\}} = K$ with K fixed, there are at most $1 + 2 + \dots + \lfloor \frac{n}{2} \rfloor \leq \frac{n(n+1)}{2} \leq \frac{(n+1)^2}{2}$ sequences to achieve it. This leads to estimate

$$
\begin{aligned}
|\det(A)| &= \sum |a_{1j_1} \dots a_{nj_n}| \leq \alpha^n \sum e^{-d_{\{j_1, j_2, \dots, j_n\}}\rho} \\
&\leq \alpha^n \sum (1 + \frac{(n+1)^2}{2} e^{-2\rho} + \dots) \leq \frac{(n+1)^2}{2}\alpha^n \frac{1}{1-e^{-\rho}},
\end{aligned}
$$

which gives the conclusion we want.

We now look at matrices M_{ij} derives by eliminating the i-th row and j-th column from the original matrix A . With the off-diagonal exponential decaying assumptions and direct determinant calculation and inequality manipulation, we can get similarly the fact that $|\det(M_{ij})| \leq \frac{n^2}{2}\alpha^{n-1}\frac{e^{-|i-j|\rho}}{1-e^{-\rho}}$. □

The proof of the previous theorem is now straightforward as follows.

Proof of Theorem 2.1: We have analyzed determinants of the related submatrices in the theoretical inverse operation. The inverse matrix of the matrix $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{1n} \end{pmatrix}$ can be express as $B = A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{1n} \end{pmatrix}$, where $A_{ij} = (-1)^{i+j}\det(M_{ij})$, M_{ij} is a $(n-1)\times(n-1)$ matrix derived by eliminating the

i-th row and j-th column from the original matrix A . From the previous lemma, we know that $|A_{ij}| \leq \frac{n^2}{2}\alpha^{n-1}\frac{e^{-|i-j|\rho}}{1-e^{-\rho}}$. We have thus $|b_{ij}| \leq \frac{n^2}{2\det(A)}\alpha^{n-1}\frac{e^{-|i-j|\rho}}{1-e^{-\rho}}$ which concludes our proof. Interestingly enough, the decay ratios for both matrices are actually the same.

Let's see what kind of band limited matrix is a off-diagonal exponentially decay matrix. Suppose $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{1n} \end{pmatrix}$ a band limited matrix with bandwidth $2M+1$, that is $a_{ij} = 0$ if $|i-j| > M$, define

$$\alpha = \min_{1 \leq i \leq n} \{|a_{ii}|\},$$

$$\rho = \min_{0<|i-j|\leq M,\, a_{ij}\neq 0} \{\frac{1}{|i-j|} \log |\frac{\alpha}{a_{ij}}|\}.$$

Conditions $\alpha > 0$ and $\rho > 0$ are required for matrix A to be off-diagonal exponentially decay, that is $|a_{ij}| \leq \alpha \cdot e^{-\rho|j-i|}$ for all i, j. These requirements simply translate into the following conditions:

$$A: \; a_{ii} \neq 0, \text{ for all } i,$$
$$B: \; |a_{ij}| \leq |a_{ii}|, \text{ if } i \neq j,$$
$$C: \; a_{ij} = 0, \text{ if } |i = j| > M.$$

These are actually very relaxed conditions (e.g. conditions A and B are even weaker than diagonal dominance). Most meaningful band limited matrices fall in this category in practice (for numerical differential equations, the previous conditions are true for most matrices derived by finite difference and finite element discretization) . As will be seen in the next section, we do demand fast decay rate for efficient numerical computations.

A subtle point commented by professor Wolfgang Hackbusch must be mentioned here for further clarification. In the standard case of partial differential equations, the off-diagonal entries of the inverse are not decaying exponentially in analysis. The inverse resembles the Green function, which for example in the 2^{nd}-order Laplace case decays like $\frac{1}{|x-y|}$ which is definitely not exponential in theory [6]. Actually, the matrix entries are in the order of $O(\frac{1}{h^2})$which is polynomial decay (let alone its inverse) in theoretical sense if finite difference is applied for discretization. Yet, we still get a band limited matrix with proper boundary conditions. By the previous conditions A-C, this matrix (in numerical analysis sense only) is with off-diagonal exponential decay entries given that the step size is fixed and the matrix is fixed. The key point, once again, is that the matrix size and entries must be fixed first. Exponential decaying properties (if so) and ratios can then be decided for numerical computations only.

We emphasize that two things need to be distinguished: the underlying partial differential equations and the derived linear system to be solved. Once the discretization scheme is fixed (include step sizes, dimensions and so on), the derived linear system is fixed. All we need to do next is how to solve this linear system with efficiency. Our view point is purely look at the numerical procedures in the second part. Even the off-diagonal entries of the inverse are not decaying exponentially with respect to step size of discretization (This is the analysis of the discretization procedure). When discretization scheme get fixed however, the derived linear system to be solved get fixed also. For most well-posed problems with proper discretization, the matrix is with good properties. Typically, it is with off-diagonal exponential decay entries in numerical sense or band-limited (include cases that the matrix can be approximated by a band limited matrix after "compression" or approximation schemes). We have also seen that as long as $a_{ii} \neq 0$, $|a_{ij}| < |a_{ii}|$, and $a_{ij} = 0$, if $|i-j| > M$, then the derived matrix $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{1n} \end{pmatrix}$ can be viewed as with off-diagonal exponentially decay entries. For a multigrid scheme with finite difference discretization, the condition number gets smaller on coarse grid (i.e. the original matrix has the largest condition number). As the step size refines, the matrix condition number becomes bigger and the matrix bandwidth grows in linear fashion. The entry decay rate on the coarse grid will not get slower though. We can see that once the discretization and level of multigrid is fixed. Every matrix get involved is with uniformly bounded condition numbers and good exponential decay properties (only in the final numerical computation sense of course).

Clearly, the previous results extended the conclusions of [3, 4] (Lemma 1.1-1.3). By great luck, it just happened this time that to analyze a general class (matrices with off-diagonal exponential decay entries) is in a sense easier than to study a special sub-class (band limited matrices).

3. An approximate algorithm to solve a single variable

We now present the following theorem, which is the main result of this paper.

Theorem 3.1. *If matrix A and its inverse $B = A^{-1}$ both have off-diagonal exponentially decay entries, $|a_{i,j}| \leq \alpha \cdot e^{-\rho|j-i|}$, $|b_{i,j}| \leq \beta \cdot e^{-\lambda|j-i|}$, for $\forall\, j$ and $i = 0, \ldots, n-1$, $\alpha > 0$, $\rho > 0$, $\beta > 0$, $\lambda > 0$, we define the following matrix and vector truncations*

$$A^{(T,i,K)} = \begin{pmatrix} a_{T(i,K,-),T(i,K,-)} & \cdots & a_{T(i,K,-),i} & \cdots & a_{T(i,K,-),T(i,K,+)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,T(i,K,-)} & \cdots & a_{i,j} & \cdots & a_{i,T(i,K,+)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{T(i,K,+),T(i,K,-)} & \cdots & a_{T(i,K,+),i} & \cdots & a_{T(i,K,+),T(i,K,+)} \end{pmatrix},$$

and

$$f^{(T,i,K)} = \begin{pmatrix} f_{T(i,K,-)} \\ \vdots \\ f_i \\ \vdots \\ f_{T(i,K,+)} \end{pmatrix},$$

where $T(i,K,-) = \max(1, i-K)$, $T(i,K,+) = \min(n, i+K)$. Suppose $x_i^{T(i,K)}$ be be the one variable in the solution vector of linear system $A^{(T,i,K)}y = f^{(T,i,K)}$ corresponds to the matrix entry $a_{i,i}$. Denote

$$\tilde{A}^{(T,i,K)} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,T(i,K,-)-1} & \cdots & a_{1,T(i,K,+)+1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ a_{T(i,K,-)-1,1} & \cdots & a_{T(i,K,-)-1,T(i,K,-)-1} & \cdots & a_{T(i,K,-)-1,T(i,K,+)+1} & \cdots & a_{T(i,K,-)-1,n} \\ & & 0 & A^{(T,i,K)} & 0 & & \\ a_{T(i,K,+)+1,1} & \cdots & a_{T(i,K,+)+1,T(i,K,-)-1} & \cdots & a_{T(i,K,+)+1,T(i,K,+)+1} & \cdots & a_{T(i,K,+)+1,n} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,T(i,K,-)-1} & \cdots & a_{n,T(i,K,+)+1} & \cdots & a_{n,n} \end{pmatrix},$$

the solution of $\tilde{A}^{(T,i,K)}y = f$ *is then* $\tilde{x}^{(T,i,K)} = \begin{pmatrix} \alpha_{[1,i-K-1]} \\ x^{(T,i,K)} \\ \beta_{[i+K+1,n]} \end{pmatrix}$ *for some* $\alpha_{[1,i-K-1]}$ *and* $\beta_{[i+K+1,n]}$. *With the notation of* $\Delta = \max_{i,K}\{\left\| \begin{matrix} \alpha_{[1,i-K-1]} \\ x^{(T,i,K)} \\ \beta_{[i+K+1,n]} \end{matrix} \right\|_\infty\}$, *we have*

$$|x_i^{T(i,K)} - x_i| \leq 4(2K+1)\Delta\alpha\beta(\sum_{j=0}^{n-1} e^{-j\rho})(\sum_{j=0}^{n-1} e^{-j\lambda})e^{-(K+1)\min\{\rho,\lambda\}}, \tag{4}$$

clearly, error $\{|x_i^{T(i,K)} - x_i|\}_{1\leq i\leq n}$ *decay to zero fast with respect to* K *uniformly.*

Proof. Given the "windowing" technique, three x_i cases need to be analyzed $1 \leq i \leq K$, $K+1 \leq i \leq n-K$ and $n-K+1 \leq i \leq n$ (they correspond to the upper-left corner, middle portion and lower-right corner of the matrix A). We present proof for the middle portion only. Other two parts can be analyzed in the same way.

Due to the fact $\tilde{A}^{(T,i,K)}\tilde{x}^{(T,i,K)} = f = Ax$, we have

$$\begin{aligned} 0 &= \tilde{A}^{(T,i,K)}\tilde{x}^{(T,i,K)} - Ax \\ &= \tilde{A}^{(T,i,K)}\tilde{x}^{(T,i,K)} - A\tilde{x}^{(T,i,K)} + A\tilde{x}^{(T,i,K)} - Ax \\ &= (\tilde{A}^{(T,i,K)} - A)\tilde{x}^{(T,i,K)} + A(\tilde{x}^{(T,i,K)} - x) \\ &= (\tilde{A}^{(T,i,K)} - A)x^{(T,i,K)} + A(\tilde{x}^{(T,i,K)} - x). \end{aligned} \tag{5}$$

The previous equation leads us to

$$\tilde{x}^{(T,i,K)} - x = A^{-1}(A - \tilde{A}^{(T,i,K)})\tilde{x}^{(T,i,K)} = B(A - \tilde{A}^{(T,i,K)})\begin{pmatrix} \alpha_{[1,i-K-1]} \\ x^{(T,i,K)} \\ \beta_{[i+K+1,n]} \end{pmatrix}. \tag{6}$$

Define

$$\begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix} = (A - \tilde{A}^{(T,i,K)})\begin{pmatrix} \alpha_{[1,i-K-1]} \\ x^{(T,i,K)} \\ \beta_{[i+K+1,n]} \end{pmatrix} = \begin{pmatrix} 0 \\ \xi_{T(i,K,-)} \\ \vdots \\ \xi_{T(i,K,+)} \\ 0 \end{pmatrix},$$

some detailed calculations based on the exponential decaying properties with absolute values gives us

$$\begin{pmatrix} 0 \\ |\xi_{T(i,K,-)}| \\ \vdots \\ |\xi_{T(i,K,+)}| \\ 0 \end{pmatrix} \leq 2\Delta\alpha(\sum_{j=0}^{n-1} e^{-j\rho})\begin{pmatrix} 0 \\ e^{-\rho} \\ \vdots \\ e^{-(K+1)\rho} \\ \vdots \\ e^{-\rho} \\ 0 \end{pmatrix}. \tag{7}$$

Please note that the previous inequality is valid with corresponding entries. For example, we have

$$(8)\qquad \begin{aligned}|\xi_{T(i,K,-)}| &\leq \Delta(\sum_{j=1}^{T(i,K,-)-1} \alpha \cdot e^{-\rho|j-T(i,K,-)|} + \sum_{j=T(i,K,+)+1}^{n} \alpha \cdot e^{-\rho|j-T(i,K,+)|}) \\ &\leq 2\Delta\alpha(\sum_{j=0}^{n-1} e^{-j\rho})e^{-\rho}\end{aligned}$$

and estimates for all other entries follow the same way.

Loot at the specific $t-$th entry of $\tilde{x}^{(T,i,K)} - x$ and apply the exponential decay properties of matrix $B = A^{-1}$, we have

$$(9)\qquad \begin{aligned}&|\tilde{x}_i^{(T,i,K)} - x_i| \\ &\leq 4\Delta\alpha\beta(\sum_{j=0}^{n-1} e^{-j\rho})(\sum_{j=0}^{n-1} e^{-j\lambda})(2\sum_{j=1}^{K} e^{-j\rho}e^{-(K-j+1)\lambda} + e^{-(K+1)\rho}) \\ &\leq 4\Delta\alpha\beta(\sum_{j=0}^{n-1} e^{-j\rho})(\sum_{j=0}^{n-1} e^{-j\lambda})(2K+1)e^{-(K+1)\min\{\rho,\lambda\}},\end{aligned}$$

and the last inequality concludes our proof. □

We must point out that the error of our approximate algorithm is also based on, besides the decaying ratios, the constant Δ. That is the window size needs to be analyzed and selected in practice. Trial and error (typically by simulation) is of course one way.

Based on the matrix analysis of the previous section, the following results are obvious.

Corollary 3.2. *If matrix A has off-diagonal exponentially decay entries, $|a_{i,j}| \leq \alpha \cdot e^{-\rho|j-i|}$, for $\forall\, j$ and $i = 0, \ldots, n-1$, $\alpha > 0$, $\rho > 0$, we have*

$$(10)\qquad |x_i^{T(i,K)} - x_i| \leq 2(2K+1)\frac{n^2\Delta\alpha^n}{\det(A)(1-e^{-\rho})}(\sum_{j=0}^{n-1} e^{-j\rho})^2 e^{-(K+1)\rho}$$

Most importantly, the previous analysis leads us to the following approximate algorithm.

Linear system decomposition algorithm: Instead of solving the original linear system $Ax = f$ directly, the sub-linear system $A^{(T,i,K)}y = f^{(T,i,K)}$ can be used to solve x_i only in an approximation sense (numerically with tolerable error). The requirement is that matrix A has off-diagonal exponentially decay entries.

As analyzed before, most band limited matrices in numerical partial differential equation cases are with off-diagonal exponentially decay entries. The other way is also true, that is off-diagonal exponential decay matrices are virtually band limited, in the sense of matrix compression. A compressed matrix $A^{(W)} = (a_{i,j}^{(W)})_{1 \leq i,j \leq n}$ can be defined as

$$(11)\qquad a_{i,j}^{(W)} = \begin{cases} a_{i,j},\ |j-i| \leq W, \\ \quad 0,\ |j-i| > W, \end{cases} \text{for } \forall\, j, \text{ and } i = 1, \ldots, n,$$

where W is called the off-diagonal band width, or $A^{(\varepsilon)} = (a_{i,j}^{(\varepsilon)})_{1\le i,j\le n}$ defined as

$$a_{i,j}^{(\varepsilon)} = \begin{cases} a_{i,j},\ |a_{i,j}| \ge \varepsilon, \\ \quad 0,\ |a_{i,j}| < \varepsilon, \end{cases} \text{ for } \forall\, j, \text{ and } i = 1, \dots, n, \tag{12}$$

where ε is called the truncation threshold. We have easily

$$\begin{aligned} &\left\| A^{(W)} - A \right\|_\infty \le a \cdot e^{-\rho(W+1}, \\ &\left\| A^{(\varepsilon)} - A \right\|_\infty \le \varepsilon. \end{aligned} \tag{13}$$

We are thus happy that our method covers practically most important matrices. The most extreme off-diagonal decay matrix is of course diagonal matrix.

Example 3.3. *Linear system with matrix*

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 1 \end{pmatrix}$$

cannot be approximately solved by the previous method. More comprehensive matrix analysis is thus needed.

Let's look at numerical stability as theorem 3.1. guarantees also the convergence. Using same notations defined in the previous proof and the standard numerical stability analysis, we can get that the "windowing" operation to solve x_i when $1 \le i \le K$ will give:

$$\frac{\left\| \delta \tilde{X}^{(T,i,K)} \right\|_p}{\left\| \tilde{X}_p^{(T,i,K)} \right\|_p} \le \frac{cond(A^{(T,i,K)})_p}{(1 - cond(A^{(T,i,K)})_p \frac{\left\| \delta A^{(T,i,K)} \right\|_p}{\left\| A^{(T,i,K)} \right\|_p})} (\frac{\left\| \delta f^{(T,i,K)} \right\|_p}{\left\| f^{(T,i,K)} \right\|_p} + \frac{\left\| \delta A^{(T,i,K)} \right\|_p}{\left\| A^{(T,i,K)} \right\|_p}),$$

with $cond(A^{(T,i,K)})_p = \left\| A^{(T,i,K)} \right\|_p \left\| A^{(T,i,K)^{-1}} \right\|_p$, and $p = 1,\ 2$ or ∞ as the corresponding matrix condition number, where condition $\left\| A^{(T,i,K)^{-1}} \right\|_p \left\| \delta A^{(T,i,K)} \right\|_p \le 1$ is hidden as usual. Therefore, a sufficient condition for stability could be the previous condition numbers are uniformly bounded.

4. Split and merge: freedom of implementation

Being able to solve one variable out a whole linear system defined by matrix with off-diagonal exponential decay entries, we can thus solve any subset or combination of variables approximately in the whole system. The practical importance of this flexibility is that we can solve only the variables we are most interested in (e.g., numerical PDE solutions at certain region or along certain curve).

A straightforward variation of the previous theorem is of course to use unequal "window" sizes as follows: using

$$A^{(T,i,K_1,K_2)} = \begin{pmatrix} a_{T(i,K_1,-),T(i,K_1,-)} & \cdots & a_{T(i,K_1,-),i} & \cdots & a_{T(i,K_1,-),T(i,K_2,+)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,T(i,K_1,-)} & \cdots & a_{ii} & \cdots & a_{i,T(i,K_2,+)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{T(i,K_2,+),T(i,K_1,-)} & \cdots & a_{T(i,K_2,+),i} & \cdots & a_{T(i,K_2,+),T(i,K_2,+)} \end{pmatrix},$$

and

$$f^{(T,i,K_1,K_2)} = \begin{pmatrix} f_{T(i,K_1,-)} \\ \vdots \\ f_i \\ \vdots \\ f_{T(i,K_1,+)} \end{pmatrix}$$

to solve for $x_i^{(T,i,K_1,K_2)}$. Following the proof of Theorem 3.1, we know that the error is decided by $\min\{K1, K2\}$ (the smaller window size) similarly.

Results of Theorem 3.1 tell us that we can solve approximately one variable in a large system. Of course, we can also solve a few variables adjacent to each other via the same "windowing" techniques (the left and right window sizes can be different). More precisely, using

$$A^{(T,[i\cdots j],K)} = \begin{pmatrix} a_{T(i,K,-),T(i,K,-)} & \cdots & \cdots & \cdots & a_{T(i,K,-),T(j,K,+)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \cdots & \begin{pmatrix} a_{i,i} & \cdots & a_{i,j} \\ \vdots & \ddots & \vdots \\ a_{j,i} & \cdots & a_{j,j} \end{pmatrix} & \cdots & \vdots \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{T(j,K,+),T(i,K,-)} & \cdots & \cdots & \cdots & a_{T(j,K,+),T(j,K,+)} \end{pmatrix},$$

and

$$f^{(T,[i\cdots j],K)} = \begin{pmatrix} f_{T(i,K,-)} \\ \vdots \\ f_i \\ \vdots \\ f_j \\ \vdots \\ f_{T(j,K,+)} \end{pmatrix}$$

to solve $x_{i\cdots j}^{(T,[i\cdots j],K)}$ (the i-th through j-th variables).

Split and Merge approach: The sub-linear system for approximate solutions $x_{i\cdots j}^{(T,[i\cdots j],K)}$ can be divided into two sub-linear systems to solve $x_{i\cdots l}^{(T,[i\cdots l],K)}$ and $x_{l\cdots j}^{(T,[\cdots j],K)}$ respectively, where $i \leq l \leq j$. Two sub-linear systems to solve $x_{i\cdots l}^{(T,[i\cdots l],K)}$ and $x_{l\cdots j}^{(T,[\cdots j],K)}$ can be merged into one sub-linear system to solve $x_{i\cdots j}^{(T,[i\cdots j],K)}$, where $i \leq l \leq j$. We call the first process "split" and the second process "merge".

Let's look at two extreme cases. Our approximate algorithm is of course the traditional direct solution for $x_{l\cdots n}^{(T,[l\cdots n],K)}$. There is no extra computation due to "window technique" processing. It has the slowest speed but with minimum amount of computation (we can take either direct or iterative approaches). The full parallel implementation of solving $\{x_i^{(T,i,K)}\}_{i=1}^n$ one by one is the fastest in speed but with maximum amount computation due to maximum possible "windowing" processing (but the computation time is independent of matrix size now). With split and merge algorithm, we can freely choose an algorithm with desired amount of computation and time of computation. These algorithms range from the slowest to the fastest in speed. It is also interesting to see the connection and harmony with the conventional approaches.

Is our approach direct or iterative? Its very essence is direct. We simply decompose the matrix into many non-overlap blocks (for the parts to be solved) and attaching two or one (if at the left or right corner) ”windows” on the edge(s) to make these blocks into over-lapping blocks for carrying out computations. For each overlapping block (simply a small linear system), we can use conventional direct or iterative methods with free choice.

5. Further discussions

Our simple intuitions have led to practical numerical algorithms. The previous results in a sense amazed us for linear algebra is a well-studied area. This short paper shows also that our understanding of finite dimensional Euclidean spaces is still limited. No numerical results are presented here due to the fact that the key techniques have been applied in Huawei HI3111 digital TV chipset with remarkable performance before 2006. In fact the first draft of the paper was originated more than ten years ago and re-polished in 2016.

Viewing from the algebraic domain decomposition point of view, what we have done is in a sense a virtual domain decomposition method in simple linear algebra form. Yet, it is very different as no iteration is involved in our approach. By analogy and just by analogy, can we say that the comparison of our approach and algebraic domain decomposition is sort of resembles the differences between the conventional direct and iterative methods? Anyway, it seems that the very ideas of decomposition can be applied and further extended for numerical procedures in approximation theory, linear programming and other problems. We will be happy to see further progresses and will try our best also to work on them.

Acknowledgement

Our gratitude goes to professors Wolfgang Hackbusch (Max-Planck-Institute, MIS), Boris Kashin (Moscow State), Gilbert Strang (MIT), Tony Chan (UCLA), Qun Lin (Chinese Academy of Sciences) and the anonymous referees for valuable suggestions that transformed our paper completely.

References

[1] Michele Benzi, Andreas Frommer, Reinhard Nabben, and Daniel B Szyld. Algebraic theory of multiplicative schwarz methods. Numerische Mathematik, 89(4):605–639, 2001.

[2] Steffen Börm, Lars Grasedyck, and Wolfgang Hackbusch. Hierarchical matrices. Lecture notes, 21:2003, 2003.

[3] Stephen Demko. Inverses of band matrices and local convergence of spline projections. SIAM Journal on Numerical Analysis, 14(4):616–619, 1977.

[4] Stephen Demko, William F Moss, and Philip W Smith. Decay rates for inverses of band matrices. Mathematics of computation, 43(168):491–499, 1984.

[5] Maksymilian Dryja and Olof B Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. New York University, Courant Institute of Mathematical Sciences, Division of Computer Science, 1989.

[6] George J Fix and W Hackbusch. Elliptic differential equations (theory and numerical treatment). Bulletin of the American Mathematical Society, 32(4):458, 1995.

[7] Junchen Du John Falkowski Jan Meyer Phong Nguyen William Smith Stephen Spence Wayne Stark Koji Tanaka Haim Teicher Qi Wang Gerhard Ammer,

Vasic Dobrica and Shuzhan Xu. Special issue on snr analysis in turbo decoding and applications. International J. of Information and System Sciences, 2(2):155–279, 2006.

[8] Gene H Golub and Charles F Van Loan. Matrix computations, volume 3. JHU Press, 2012.

[9] P. Lions. On Schwarz alternating methods. II. SIAM, 1989.

[10] Pierre-Louis Lions. On the schwarz alternating method. i. In First international symposium on domain decomposition methods for partial differential equations, pages 1–42. Paris, France, 1988.

[11] Thomas A Manteuffel and Seymour V Parter. Preconditioning and boundary conditions. SIAM Journal on Numerical Analysis, 27(3):656–694, 1990.

[12] T. Chan R. Chan and G. H. Golub. Iterative methods in scientific computing. Springer, 1997.

[13] Barry Smith, Petter Bjorstad, William D Gropp, and William Gropp. Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge university press, 2004.

[14] Gilbert Strang and George J Fix. An analysis of the finite element method, volume 212. Prentice-hall Englewood Cliffs, NJ, 1973.

[15] Wei Pai Tang. Generalized schwarz splittings. SIAM Journal on Scientific and Statistical Computing, 13(2):573–595, 1992.

[16] James William Thomas. Numerical partial differential equations: finite difference methods, volume 22. Springer Science & Business Media, 2013.

[17] Ulrich Trottenberg, Cornelius W Oosterlee, and Anton Schuller. Multigrid. Academic press, 2000.

[18] Raf Vandebril, Marc Van Barel, and Nicola Mastronardi. Matrix computations and semiseparable matrices: linear systems, volume 1. JHU Press, 2007.

[19] Richard S Varga. Matrix iterative analysis, volume 27. Springer Science & Business Media, 2009.

[20] Cai Xiaocun. An additive schwarz algorithms for parabolic convection diffusion equations. Numer Math, 60:41–61, 1991.

School of Mathematics and Statistics, Jiangsu Normal University, Xuzhou China and Institute of Applied Mathematics, The Chinese academy of Sciences, Beijing, China

E-mail: qschang@amss.ac.cn

Department of Applied Mathematics, Hong Kong Polytechnic University, Kowloon, Hong Kong, China

E-mail: yanping.lin@polyu.edu.hk and yanlin@ualberta.ca

Shangrilantis Limited, Hong Kong, China

E-mail: xushuzhan1965@163.com