The arrowhead decomposition method for a block-tridiagonal system of linear equations

P A Belov¹, E R Nugumanov¹,² and S L Yakovlev¹
¹ Department of Computational Physics, St. Petersburg State University, Ulyanovskaya 1, 198504 St. Petersburg, Russia
E-mail: pavelbelov@gmail.com

Abstract. The arrowhead decomposition method (ADM) for the parallel solution of a block-tridiagonal system of linear equations is presented. The method consists in rearranging the initial linear system into an equivalent one with the “arrowhead” structure of the matrix. It is shown that such a structure provides a good opportunity for parallel solving. The computational speedup of ADM with respect to the sequential matrix Thomas algorithm is analytically estimated based on the number of elementary multiplicative operations for the parallel and serial parts of the methods. A number of parallel processors required to reach the maximum computational speedup is found. A good agreement of the analytical estimations of the computational speedup and practically obtained results is observed.

1. Introduction
Many boundary value problems arising in physics and applied mathematics after discretization lead to a linear system with a block-tridiagonal matrix. As a simple example, one can consider a finite-difference approximation of the two-dimensional Poisson equation on the equidistant grids over each variable. In this case, each block of the matrix becomes large and sparse, but the block-tridiagonal structure is kept. A combination of the finite-difference method over one coordinate and the basis set method [1] over another coordinate generally produces block-tridiagonal matrix with dense blocks. The intermediate case of the basis set with a local support, namely splines [2], results in the band blocks [3, 4] with the bandwidth comparable with that one from the finite-difference method.

The direct method for solving the linear systems is the Gauss elimination. An adaptation of the Gauss elimination to the block-tridiagonal systems is known as the matrix Thomas algorithm [5] or as the matrix sweeping algorithm [6]. In this algorithm, the idea of Gauss elimination is applied to the blocks themselves. The algorithm is well defined and robust for matrices with the diagonal dominance, but its sequential nature makes it difficult to be applied for parallel calculations.

The present paper is focused on the developed arrowhead decomposition method (ADM) for efficient parallel solving the block-tridiagonal linear systems. In this method, the initial matrix is logically reduced to the “arrowhead” form, namely some new independent on-diagonal blocks, sparse off-diagonal blocks and a coupling matrix of much smaller size. The method originates from the domain decomposition [7-11] and nested dissection [12, 13], where the idea

² Present address: BashNIPIneft LLC, 86/1, Lenin Street, 450006 Ufa, Russia
to divide a large problem into small ones which can, to some extent, be solved independently was introduced. In Ref. [14] this idea is illustrated by the tridiagonal matrix obtained from the finite-difference discretization of the one-dimensional Laplace operator. It is shown that this method is faster than the Wang algorithm [15, 16] and following ones. In this paper, the computational speedup [17] of ADM with respect to the sequential matrix Thomas algorithm is analytically estimated based on the number of elementary multiplicative operations for the parallel and serial parts of the methods. A number of parallel processors required to achieve the maximum computational speedup is obtained. The analytical results are compared with the results of practical calculations. For the test linear systems, we use the discretized boundary value problems for the integro-differential Faddeev equations [18-20, 4, 21, 22].

2. Arrowhead decomposition method

The block-tridiagonal linear system under consideration is described by the equation

\[ A_i X_{i-1} + C_i X_i + B_i X_{i+1} = F_i, \quad A_1 = B_N = 0, \]

where \( A_i, B_i, C_i, i = 1, \ldots, N \), are the blocks of the matrix and \( F_i \) are the blocks of the right-hand side (RHS) supervector \( F \). The unknown supervector \( X \) is composed of blocks \( X_i \). The size of each block of the matrix is \( n \times n \), whereas the size of each block of the supervectors is \( n \times l \), where \( l \geq 1 \) is the number of columns.

The idea of the ADM is presented in Figure 1. The initial block-tridiagonal linear system (1) is rearranged into the equivalent “arrowhead” form which allows the parallel solving. The rearrangement is performed by interchanging the block-rows and block-columns. The interchange of block-columns also leads to the change of the elements of the unknown vector and the RHS vector. The new structure of the matrix can be represented by the \( 2 \times 2 \) block-matrix

\[
\begin{pmatrix}
S & W_R \\
W_L & H
\end{pmatrix}
\begin{pmatrix}
s \\
h
\end{pmatrix} =
\begin{pmatrix}
F_s \\
F_h
\end{pmatrix}.
\]

(2)

Here, the unknown solution \( h \) corresponds to the moved part of the full solution. The notation is shown in Figure 1. The square superblock \( S \) consists of the new independent blocks \( S^k \), \( k = 1, \ldots, M \), on the diagonal. The matrix element \( H \) is the bottom right coupling superblock. Other lateral superblocks \( W_R, W_L \) present additional blocks of the matrix. The solution of the system (2) is given by the relations

\[
\begin{cases}
s = S^{-1} F_s - S^{-1} W_R h \\
h = (H - W_L S^{-1} W_R)^{-1} (F_h - W_L S^{-1} F_s)
\end{cases}
\]

(3)

These relations contain matrix products and inverses which can, to some extent, be done in parallel. The independence of blocks \( S^k \) allows the parallel calculation of the products \( S^{-1} F_s \) and \( S^{-1} W_R \). Actually, instead of inverses we solve in parallel over \( k \) the linear systems for \( S^k \).

The sparse structure of blocks \( W_L \) and \( W_R \) drastically reduces the number of matrix operations in Eq. (3). Once a part \( h = (h^1, \ldots, h^{M-1})^T \) of the total solution \( X = (s, h)^T \) is obtained from the second relation of Eq. (3), the remaining part is calculated in parallel over \( k \) by the formula

\[ s^k = z^k - Z^k h^{k-1} - Z^k h^k, \]

where formally \( z^k = (S^k)^{-1} F_s, Z^k = (S^k)^{-1} W_R \), and \( h^0 = h^M = 0 \).

To solve the independent linear systems with \( S^k \) and the equation for \( h \), one can apply any appropriate technique. Although the ADM can be used recursively (nestedly) for solving these linear systems [23], in the present paper, we employ the matrix Thomas algorithm.
Figure 1. A graphical scheme for rearrangement of the initial block-tridiagonal linear system into the equivalent form. The nonzero blocks and vectors are denoted by thick lines. New independent superblocks and corresponding vectors at each panel are denoted by thin lines. Top panel: the initial linear system with the marked interchanged blocks. Central panel: the obtained rearranged system with the “arrowhead” matrix. Bottom panel: the notation of the matrix elements for the rearranged system.
3. Computational speedup
The computational speedup of the ADM with respect to the sequential matrix Thomas algorithm can be estimated as the ratio of the computation time by the matrix Thomas algorithm to that one of the ADM [17]. The computation time is directly related to the number of serial operations, namely products and divisions, of each algorithm. We calculated a number of serial multiplicative operations for both algorithms and, as a consequence, analytically estimated the computational speedup. The additive operations were not taken into account assuming that they require much less computational time. The time for the memory management is also considered to be negligible.

According to Ref. [24], we hold that for multiplication and inverse of a block of size \( n \) one needs exactly \( n^3 \) multiplicative operations. These estimations are in agreement with the conventional realizations of these algorithms, namely LINPACK and LAPACK [25, 26]. Then, the number of multiplicative operations for the sequential matrix Thomas algorithm is [27]

\[
O_{TA} = (3N - 2)(n^3 + n^2l).
\]

For the ADM, the total number of multiplicative operations was found to be given by the formula

\[
O_{ADM} = \left( \frac{4N_1 - 1}{M} \right)(7n^3 + 5n^2l) + (3M - 5)(n^3 + n^2l),
\]

where \( N_k, k = 1, \ldots, M \) is the number of blocks on the diagonal of \( S^k \). In general, \( N_k \) may be different for different \( k \), but should satisfy the relation:

\[
\sum_{k=1}^{M} N_k = N - (M - 1).
\]

The maximum performance is achieved if the computation time for solving the linear system with the block \( S^k \) is equal for each \( k \). In this case, the quantities \( N_k \) are equal for \( k = 2, \ldots, M - 1 \), and \( N_1, N_M \) are larger than the others. The profit in the performance of such choice of \( N_k \) with respect to the case when all \( N_k, k = 1, \ldots, M \) are equal is relatively small. Therefore, for the sake of simplicity, we consider the case when all \( N_k \) are the same. In such a case, if the number of parallel processors equals to the number of blocks \( M \) on the diagonal, then the number of serial operations of the ADM will be

\[
O_{ADM} = \left( \frac{4N_1 - M + 1}{M} \right)(7n^3 + 5n^2l) + (3M - 5)(n^3 + n^2l).
\]

The computational speedup, as the ratio \( S = O_{TA}/O_{ADM} \), is easily obtained:

\[
S = \frac{3N - 2}{3M - 5 + \left(5 + \frac{2}{l/n}\right) \left(\frac{N-M+1}{M}\right)}.
\]

One can estimate that the dependence of the computational speedup on the ratio \( l/n \) is weak for \( l \ll n \). Figure 2 shows the computational speedup (5) for \( N = 3071, n = 100, 400, l = 1 \). As a function of number of parallel processors, \( M \), the speedup is linear \( S = 3M/7 \) for \( M \ll N \). As \( M \) increases further, the speedup flattens out, reaches its maximum and, then, decreases. Based
on the obtained speedup (5), the parallel efficiency of the ADM can be defined as
\[ E = \frac{S}{M} \] [28].
A number of parallel processors required to achieve the maximum speedup is given as
\[ M^* = \sqrt{\frac{(N + 1)}{3} \left( 5 + \frac{2}{1 + l/n} \right)} \] (6).

As a result, for using the ADM one can choose all independent blocks \( S^k \) of the same size. The optimal number of parallel processors \( M^* \) for the fixed number of blocks \( N \) on the diagonal is given by Eq. (6).

![Figure 2.](image-url)

**Figure 2.** Left panel: the analytically estimated computational speedup (solid curve) of the ADM with respect to the matrix Thomas algorithm as well as the speedup obtained in practical calculations (empty squares and circles) as a function of the number of parallel processors, \( M \geq 4 \). Right panel: the parallel efficiency of the ADM for the same series of numerical experiments. The size of each block of the block-tridiagonal matrix is \( n = 100, 400 \), a number of blocks in the diagonal is \( N = 3071 \), only one RHS vector is used.

4. Practical results

The analytical estimations, given in the previous section, were validated in practical calculations. We used the Symmetric Multiprocessor (SMP) system with 64 processors (Intel Xeon CPU X7560 2.27GHz) and 2TB of shared operative memory. The ADM was implemented as an independent program written in C/C++. For the linear algebra operations, the corresponding LAPACK 3.5.0 subroutines [26] were called. The parallelization was done using the OpenMP 3.0 [29].

In validation studies of the computational speedup (5) only the solution time was taken into account. This time includes time needed for computation itself as well as time required for possible memory management during the solution process. The time for generation of the initial blocks was ignored.

The initial block-tridiagonal matrix was obtained from the well-known discretization of the two-dimensional Laplace operator of the Faddeev equations in polar coordinates [18, 19, 4]. The block-tridiagonal structure arose from an expansion of the unknown solution of these equations on a basis of the Hermite splines [3, 30] over the \( \theta \)-grid and the finite-difference approximation of the second derivative over the equidistant \( \rho \)-grid [31]. Additionally, the zero elements of the blocks were filled by relatively small random nonzero values to guarantee the integrity of our studies.
In Figure 2, we show the computational speedup as well as the parallel efficiency of the ADM measured in a series of experiments with various numbers $M$ of parallel processors and sizes of blocks taken to be $n = 100, 400$. The general trend of the analytical result (5) is clearly confirmed by the practical calculations. Since for a small number of processors, $4 \leq M \leq 24$, the computational speedup is linear with respect to the number of parallel processors $S = 3M/7$, the efficiency is bounded by the coefficient $3/7$. For larger $M$, the linear behaviour is distorted. The practical results are a little bit smaller than the analytical ones. This difference may be attributed to the nonideality of the memory management of the computational system or to additional system processes running simultaneously and affecting the performance of the studied task. These issues are especially noticeable for the case when the number of parallel processors is $M = 48$ and $M = 64$. For these cases, the practically achieved computational speedup is apparently smaller than the analytical one.

5. Conclusions
In summary, we presented the arrowhead decomposition method for the efficient parallel solution of the block-tridiagonal systems of linear equations. The method consists in rearranging the initial linear system into the equivalent one with the “arrowhead” structure of the matrix. The computational speedup of the arrowhead decomposition method with respect to the sequential matrix Thomas algorithm was analytically estimated based on the number of elementary multiplicative operations for the parallel and serial parts of the methods. We showed that the maximum of the computational speedup for a given number of blocks on the diagonal is achieved for a finite number of parallel processors. A number of parallel processors required to reach the maximum computational speedup was obtained. A good agreement of analytical estimations of the computational speedup and practically measured results was observed.

Acknowledgments
Financial support from SPbU (grant No. 11.38.241.2015) and RFBR (grant No. 16-32-00047) is acknowledged. The calculations were carried out using the facilities of the SPbU Resource Center “Computational Center of SPbU”.

References
[1] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 Numerical recipes: the art of scientific computing (New York: Cambridge University Press)
[2] Ahlberg J H, Nilson E N and Walsh J L 1967 Theory of splines and their applications (New York: Academic Press)
[3] Prenter P M 1975 Splines and variational methods (New York: Wiley)
[4] Belov P A and Yakovlev S I 2013 Phys. Atom. Nucl. 76 126
[5] Thomas L H 1949 Elliptic problems in linear difference equations over a network Watson Sci. Comput. Lab. Rept. (New York: Columbia University)
[6] Samarskii A A and Nikolaev E S 1978 Methods for solving finite-difference equations (Moscow: Nauka)
[7] Bjerstad P E and Widlund O B 1986 SIAM J. Numer. Anal. 23 1097
[8] Il'in V P, Litvinenko S A and Sveshnikov V M 2003 Parallelization of alternating direction implicit methods for three-dimensional domains Lect. Not. in Comp. Sci. 2763 89
[9] Korneev V G 2013 Comp. Meth. in Appl. Math. 13 161
[10] Korneev V G and Langer U 2015 Dirichlet-dirichlet domain decomposition methods for elliptic problems (Singapore: World Scientific)
[11] Gurieva Y L, Il'in V P and Perevozkin D V 2016 Vychisl. Metody Programm. 17 132
[12] Duff I S, Erisman A M and Reid J K 1986 Direct methods for sparse matrices (Oxford: Clarendon Press)
[13] Solovev S A 2014 Vychisl. Metody Programm. 15 441
[14] Ortega J M 1988 Introduction to parallel and vector solution of linear systems (New York: Springer)
[15] Wang H H 1981 ACM Trans. Math. Soft. 7 170
[16] Volokhova A V, Zemlyanaya E V and Rikhvitskiy V S 2015 Vychisl. Metody Programm. 16 281
[17] Gallopoulos E, Philippe B and Sameh A H 2016 Parallelism in matrix computations (Springer)
[18] Merkuriev S P, Gignoux C and Laverne A 1976 Ann. Phys. 99 30
[19] Faddeev L D and Merkuriev S P 1993 Quantum scattering theory for several particle systems (Dordrecht: Kluwer)
[20] Belov P A and Yakovlev S L 2012 Bull. Russ. Acad. Sci. Phys. 76 913
[21] Belov P A and Yakovlev S L 2014 Phys. Atom. Nucl. 77 344
[22] Belov P A and Yakovlev S L 2017 Few-Body Syst. 58 24
[23] Belov P A, Nugumanov E R and Yakovlev S L 2015 Decomposition method for block-tridiagonal matrix systems E-print arXiv:1505.06864
[24] Samarskii A A and Gulin A V 1989 Numerical methods (Moscow: Nauka)
[25] Dongarra J J et al 1979 LINPACK Users’ guide (Philadelphia: SIAM)
[26] Anderson E et al 1999 LAPACK Users’ guide (Philadelphia: SIAM)
[27] Isaacson E and Keller H B 1966 Analysis of numerical methods (New York: Wiley)
[28] Douglas C C, Haase D and Langer U 2003 A tutorial on elliptic PDE solvers and their parallelization (Philadelphia: SIAM)
[29] Adeniyi-Jones C et al 2008 OpenMP Application Program Interface. Http://www.openmp.org/mp-documents/spec30.pdf
[30] Pupyshev V V 2004 Phys. Part. Nucl. 35 257
[31] Khramtsov E S, Belov P A, Grigoryev P S et al. 2016 J. Appl. Phys. 119 184301