DOI 10.1007/s10958-016-2939-2
Journal of Mathematical Sciences, Vol. 216, No. 6, August, 2016

THE ADDITIVE PEACEMAN–RACHFORD METHOD

N. I. Gorbenko* **and V. P. Il'in*** UDC 519.6

A new version of the parallel Alternating Direction Implicit (ADI) method by Peaceman and Rachford for solving systems of linear algebraic equations with positive-definite coefficient matrices represented as sums of two commuting terms is suggested. The algorithms considered are suited for solving two-dimensional grid boundary-value problems with separable variables, as well as the Sylvester and Lyapunov matrix equations. The approach to rising parallel efficiency proposed in the paper is based on representing rational functions as sums of partial fractions. An additive version of the factorized ADI method for solving Sylvester's equation is described. Estimates of the speedup resulting from increasing the number of computer units are presented. These estimates demonstrate a potential advantage of using the additive algorithms when implemented on a supercomputer with large number of processors or cores. Bibliography: 5 *titles.*

1. Introduction

The Alternating Direction Implicit (ADI) method of Peaceman–Rachford has been known for about sixty years and has been investigated by many authors, in particular, from the standpoint of its suitability for implementation on multiprocessor computer systems, see, e.g., [1–4] and the references therein. There is quite a number of ADI type methods, and they are essentially equivalent with respect to the convergence rate. For this reason, we limit our considerations to the Peaceman–Rachford method, which is denoted by ADI for shortness. In the last years, such algorithms have attracted increased interest in connection with their application to solving Lyapunov and Sylvester matrix equations [5].

In application to solving systems of linear algebraic equations of the form

$$Au = f, \;\; A = A_1 + A_2, \;\; A \in \mathcal{R}^{N,N}, \tag{1}$$

the method under consideration is described by the formulas

$$\begin{aligned} (I + \tau_n A_1)u^{n-1/2} &= (I - \tau_n A_2)u^{n-1} + \tau_n f, \\ (I + \tau_n A_2)u^{n} &= (I - \tau_n A_1)u^{n-1/2} + \tau_n f, \end{aligned} \tag{2}$$

where n, τ_n, and I are the iteration number, iteration parameter, and identity matrix, respectively. Observe that other alternating direction implicit methods, having essentially the same convergence rate, are known. In the sequel, such methods will also be denoted by ADI. Here and below, we assume that the matrices A_1 and A_2 are real and positive semidefinite; in addition, it is assumed that one of them is positive definite, i.e.,

$$(A_k v, v) \geq \delta_k (v, v), \;\; \delta_k \geq 0, \quad k = 1, 2; \quad \delta_1 + \delta_2 > 0, \;\; v \in \mathcal{R}^n. \tag{3}$$

ADI methods possess the highest convergence rate in the case where the original coefficient matrix of a linear system is represented as a sum of two commuting terms with known eigenvalue bounds. Such matrices arise, for instance, in finite-volume or finite-element approximation of two-dimensional boundary-value problems with separable variables. In this case, if the eigenvalues of one of the matrices A_k, $k = 1, 2$, are known, then the ADI methods, with an appropriate choice of the parameters τ_n, yield the exact solution of the system in N_k iterations (provided that no round-off errors are made), where N_k is the number of grid nodes

*Institute of Computational Mathematics and Mathematical Geophysics, SB RAS, Novosibirsk, Russia; Novosibirsk State University, Novosibirsk, Russia, e-mail: ilin@sscc.ru.

Translated from *Zapiski Nauchnykh Seminarov POMI*, Vol. 439, 2015, pp. 47–58. Original article submitted November 23, 2015.

in one of the coordinate directions, $N = N_1 N_2$. More exactly, the number of iterations equals $\bar{N}_k \le N_k$, where $\bar{N}_k$ is the number of distinct eigenvalues of the matrix A_k. Note that from the standpoint of achieving a given precision ε in the sense that the Euclidean norm of the residual satisfies the stopping criterion

$$||r^n|| \equiv ||f - Au^n|| \le \varepsilon ||f||, \;\; \varepsilon \ll 1, \tag{4}$$

in a minimal number of iterations $n(\varepsilon)$, the optimal values of the parameters τ_n are determined by solving Zolotarev's problem [1], and, in this case,

$$n(\varepsilon) \le C \, \log \varepsilon^{-1} \log(\mathrm{cond}(A)). \tag{5}$$

Here, C is a constant, and $\mathrm{cond}(A)$ is an upper bound for the condition number of A. Strictly speaking, if the matrices A_1 and A_2 have different spectral bounds, then the optimal values of the iteration parameters in (2) have different values, $\tau_n^{(1)}$ and $\tau_n^{(2)}$, for the first and second equations, respectively. However, in the case where $\mathrm{cond}(A_1)$ and $\mathrm{cond}(A_2)$ are of the same order in N, Eqs. (2) allow one to obtain an "almost optimal" (or an optimal in the order) sequence of the parameters τ_n. For simplicity, we limit our considerations to this case.

Since, for instance, for grid approximations of elliptic second-order equations (for brevity, they will be referred to as the model systems of linear equations), $\mathrm{cond}(A) \sim O(N)$, the corresponding number of iterations proves to be proportional to $\log N$, and the ADI methods are competitive with the Fast Fourier Transform (FFT) algorithm, and a specific interrelation of their performances on a multiprocessor computer system depends on the properties of a problem to be solved, as well as on the technologies of program parallel implementation used.

2. Some properties of the ADI method

Upon elimination of the auxiliary vector $u^{n-1/2}$ from (2), the iterative process considered can be written as

$$(s_n I + A_1)(s_n I + A_2)u^n = (s_n I - A_1)(s_n I - A_2)u^{n-1} + 2s_n f, \tag{6}$$

where the new parameter $s_n = \tau_n^{-1}$ plays the role of a shift for the eigenvalues of the matrices A_1 and A_2. Define the preconditioning matrix

$$B_n = \frac{1}{2s_n}(s_n I + A_1)(s_n I + A_2). \tag{7}$$

Then relation (6) can be brought to the form

$$u^n = u^{n-1} + B_n^{-1} r^{n-1}, \tag{8}$$

and we obtain the following relations for the residual vectors:

$$\begin{aligned} r^n &= r^{n-1} - AB_n^{-1} r^{n-1} = (B_n - A)B_n^{-1} r^{n-1} \\ &= (s_n I - A_1)(s_n I - A_2)(s_n I + A_2)^{-1}(s_n I + A_1)^{-1} r^{n-1}. \end{aligned} \tag{9}$$

Assuming that the matrices A_1 and A_2 commute, $A_1 A_2 = A_2 A_1$ (this property has not been exploited until now), from (8) we derive

$$r^n = R_n(A_1) R_n(A_2) r^{n-1} \equiv T_n r^{n-1}, \tag{10}$$

where we denote

$$R_n(A_k) = P_n(A_k) Q_n^{-1}(A_k), \;\; P_n(A_k) = s_n I - A_k, \;\; Q_n(A_k = (s_n I + A_k), \quad k = 1, 2, \tag{11}$$

and T_n is the iteration matrix. From (9) and (10) it follows that

$$\begin{aligned} r^n &= \bar{R}_n(A_1)\bar{R}_n(A_2)r^0, \\ \bar{R}_n(A_k) &= \bar{P}_n(A_k)\bar{Q}_n^{-1}(A_k) = \prod_{l=1}^{n} P_l(A_k)\prod_{l=1}^{n} Q_l^{-1}(A_k), \end{aligned} \tag{12}$$

where, in the scalar case, $\bar{R}_n(s)$ is a rational function of the form

$$\bar{R}_n(s) = \frac{\bar{P}_n(s)}{\bar{Q}_n(s)} \equiv \frac{(s_1 - s)\cdots(s_n - s)}{(s_1 + s)\cdots(s_n + s)}. \tag{13}$$

Observe that if the residual vectors $r^0, r^1, \ldots, r^{n-1}$ are known, then the approximate solutions can be computed with the use of the formula

$$u^n = u^0 + B_1^{-1}r^0 + \cdots + B_n^{-1}r^{n-1}, \tag{14}$$

following from (8).

Instead of (14), one can also use relations not involving the residual vectors. To this end, we write

$$u^n = T_n u^{n-1} + g_n, \quad g_n = 2s_n Q_n(A_1)Q_n(A_2)f, \tag{15}$$

implying that

$$u^n = T_{n,1}u^0 + T_{n,2}g_1 + \cdots + T_{n,n}g_{n-1} + g_n. \tag{16}$$

Here, we use the notation

$$T_{n,l} = T_{n,l}^{(1)}T_{n,l}^{(2)}, \quad T_{n,l}^{(k)} = \prod_{m=l}^{n}(s_m I + A_k)^{-1}(s_m I - A_k). \tag{17}$$

3. The additive Peaceman–Rachford method

As is seen from the previous section, the ADI method can be realized with the use of rational matrix functions of the form (12). From the standpoint of parallelization, it is reasonable to perform operations with such functions based on their partial fraction decompositions, which will be looked for in the following form:

$$\bar{R}_n(A) \equiv \prod_{l=l}^{n}(s_l I - A)(s_l I + A)^{-1} = z_0 I + \sum_{l=1}^{n} z_l R_l(A). \tag{18}$$

Here, $s_1, \ldots, s_n$ are some given numbers; $z_1, \ldots, z_n$ are the desired coefficients of the decomposition, and the subscript k is omitted for simplicity. Multiplying both sides of (18) by the matrix polynomial $\bar{Q}_n(A)$, we arrive at the relations

$$\begin{aligned} \bar{P}_n(A) &= (s_1 I - A)\cdots(s_n I - A) = z_0\bar{Q}_n(A) + \sum_{l=1}^{n} z_l \bar{S}_l(A), \\ \bar{S}_l(A) &= \prod_{k=l}^{n}(t_{k,l}I + A), \end{aligned} \tag{19}$$

where $t_{k,l} = -s_l$ if $k = l$ and $t_{k,l} = s_l$ otherwise. By equating the coefficients at the same powers of A in (19), we obtain a system of n linear algebraic equations with respect to the unknowns z_l, $l = 1, \ldots, n$.

Represent the matrix polynomials $\bar{P}_n(A)$, $\bar{Q}_n(A)$, and $\bar{S}_k(A)$ of order n as follows:

$$\begin{aligned}\bar{P}_n(A) &= b_0 I - b_1 A + \cdots + (-1)^n b_n A^n,\\ \bar{Q}_n(A) &= b_0 I + b_1 A + \cdots + b_n A^n,\\ \bar{S}_l(A) &= b_{0,l} I + b_{1,l} A + \cdots + b_{n,l} A^n.\end{aligned} \tag{20}$$

The coefficients occurring in (20) are given by the formulas

$$b_n = 1, \qquad b_{n-1} = \sum_{k=1}^{n} s_k, \qquad b_{n-k} = \sum_{1\le i_1<i_k\le n}\ \prod_{j=l}^{k} s_{i_j}, \qquad b_0 = s_1 \dots s_n,$$

$$b_{n,l} = 1, \qquad b_{n-1,l} = \sum_{i=1}^{n} t_{l,i}, \qquad b_{n-k,l} = \sum_{1\le i_1<\dots i_k\le n}\ \prod_{j=l}^{k} t_{l,i_j}, \qquad b_{0,l} = t_{1,l} \dots t_{n,l}.$$

From the definition of $t_{l,k}$ it follows that for all $l = 1, \dots, n$, we have $b_{0,l} = -b_0$. Furthermore, from (19) we obtain

$$\begin{aligned} b_0 z_0 - \sum_{l=1}^{n} b_{0,l} z_l &= b_0,\\ b_k z_0 - \sum_{l=1}^{n} b_{k,l} z_l &= (-1)^k b_k, \quad k = 1, \dots, n.\end{aligned} \tag{21}$$

It is convenient to eliminate z_0 from Eqs. (21), which yields

$$z_0 = 1 - \sum_{l=1}^{n} z_l,$$

$$\sum_{l=1}^{n} \Big(b_k + \sum_{l=1}^{n} b_{k,l} \Big) z_l = \big[(-1)^k - 1\big] b_k, \quad k = 1, \dots, n.$$

Thus, for the unknown vector $z = \{z_k\}$ we have the following system of n linear algebraic equations:

$$C\, z = g, \quad C = \{c_{k,l} = 1 + b_{k,l}/b_k\}, \quad g = \{g_k = (-1)^k - 1\}. \tag{22}$$

If, in accordance with formulas (18)–(22), we find the coefficients $z_0^{(k)}, z_1^{(k)}, \dots, z_n^{(k)}$, $k = 1, 2$, for the partial fraction decompositions of the rational functions $\bar{R}_n(A_1)$ and $\bar{R}_n(A_2)$, then, for the residual vector r^n from (12), we obtain the expression

$$r^n = \Big[z_0^{(1)} I + \sum_{l=1}^{n} z_l^{(1)} R_l(A_1) \Big] \Big[z_0^{(2)} I + \sum_{l=1}^{n} z_l^{(2)} R_l(A_2) \Big] r^0. \tag{23}$$

One can also find another representation of a rational function in terms of partial fractions. To this end, instead of (18), we use the relation

$$n \bar{R}_n(A) = \sum_{l=1}^{n} (\widehat{z}_i I - A)(s_i I + A)^{-1}, \tag{24}$$

where $\widehat{z}_i$ are the new coefficients to be determined.

In this case, instead of (22), for the coefficients $\widehat{z}_i$ from (24) we obtain the system of linear algebraic equations

$$\widehat{C}\widehat{z} = \widehat{g}, \quad \widehat{z} = \{\widehat{z}_i\}, \tag{25}$$

where the entries of the coefficient matrix $\widehat{C}$ and the components of the right-hand-side vector $\widehat{g}$ are determined by the relations

$$\widehat{C} = \{\widehat{C}_{i,j} = \sum_{k=1}^{n-1} b_{n-1,k}(\widehat{t}_{1,i}, \ldots, \widehat{t}_{j,i}, \ldots, \widehat{t}_{n-1,i})\},$$

$$\widehat{g} = \{\widehat{g}_i = (-1)^i n b_{n,i}(s_1, \ldots, s_n) - \sum_{j=1}^{n} b_{n,j}(\check{t}_{1,i}, \ldots, \check{t}_{j,i}, \ldots, \check{t}_{n,i})\}.$$

Here, we use the notation

$$\widehat{t}_{j,i} = \begin{cases} s_i, & i = j, \\ s_{i+1}, & i \neq j, \end{cases} \qquad \check{t}_{j,i} = \begin{cases} 0, & i = j, \\ s_j, & i \neq j. \end{cases}$$

Formulas (18)–(25), considered in the present section, assume that the iteration parameters $s_1, \ldots, s_n$ are prescribed. We do not dwell on the problem of choosing them because it is widely covered in the literature. It should only be emphasized that the shifts s_k and the iteration number $n(\varepsilon)$ are assumed to be determined by the stopping criterion ε and the spectral bounds for the matrix A, $0 < m \leq \lambda(A) \leq M < \infty$, in accordance with (4) and (5), where one can set $\mathrm{cond}(A) = M/m$.

Once the residual vectors are found, the approximate solution u^n can be computed by using (14) or (16). We emphasize that preliminarily one must determine the vectors $r^0, \ldots, r^{n-1}$ in the first case or $g_1, \ldots, g_n$ in the second one.

In the ADI methods under consideration, we do not use iterative processes in Krylov's subspaces because this will not essentially change the interrelation between the parallel performances of the multiplicative algorithm (2) and the corresponding additive versions, an analysis of which is the main object of the present paper.

Note also that for small n, the coefficient matrices C and $\widehat{C}$ of systems (22) and (25) for determining the coefficients of the partial fraction decompositions prove to be nonsingular, provided that the shift parameters s_k are distinct. However, in the general case, the solvability and numerical stability of systems (22) and (25) must be specially investigated.

4. Estimates of the parallel efficiency of the ADI methods

An analysis of the parallel efficiency of different implementations of the Peaceman–Rachford method will be done, for the sake of simplicity, on the model problem resulting from the five-point difference approximation of the two-dimensional Dirichlet problem for the Poisson equation on a square domain discretized using the square grid with step size h, see [1–4] for more details. In this case, the matrices A_1 and A_2 are positive definite and block tridiagonal (if the unknowns are ordered naturally), and the order of the linear system is equal to $N = N_1 N_2 = h^{-2}$, where N_1 and N_2 are the numbers of nodes in the two coordinate directions, $N_1 = N_2 = h^{-1}$.

For simplicity, the performances of different algorithms will be compared on an abstract computer system with huge shared memory, sufficient for solving the problems considered by multi-thread technology without data exchanges and conflicts in data access. We may assume, for instance, that the algorithm is implemented on a Central Processing Unit (CPU) or a Graphics Processing Unit (GPU) with a sufficient number of cores.

The time, T_m, of performing computations by the multiplicative ADI method in accordance with formulas (2) (which will be measured in computer cycles, the time of a cycle being determined by the clock period of the multiprocessor computer system) can be estimated by

the expression

$$T_m = 2(T_s + T_r)\, n(\varepsilon). \tag{26}$$

Here, T_s is the number of cycles needed for solving a system of linear algebraic equations with a coefficient matrix of the form $I + \tau A_k$, and T_r is the time required to compute the right-hand sides in each of the equations in (2).

The quantities T_s and T_r can be evaluated using the formulas

$$T_s = C_s/P, \quad T_r = C_r/P, \tag{27}$$

where P is the number of computing threads (it is assumed that the number of grid nodes along one direction is proportional to P, i.e., $N^{1/2}/P$ is an integer), whereas C_s and C_r are certain numbers determined by a specific algorithm being applied. For instance, linear systems with tridiagonal coefficient matrices can be solved either by the "standard" forward and backward substitutions or by a spectral algorithm. The quantities C_s and C_r are assumed to be the same for both the multiplicative and additive ADI methods; therefore, from the standpoint of their comparison, specific values of C_s and C_r are of no importance.

In deriving formulas (27), it is taken into account that the solution of block tridiagonal linear systems actually reduces to parallel solution of decoupled tridiagonal systems corresponding to grid lines, first, along the first direction and then along the second one. At each of these iterative half-stages, parallelization is performed in a natural (and practically optimal) way, so that the total number of cycles for solving a linear systems using (2) is equal to

$$T_m = 2(C_s + C_r)\, n(\varepsilon)/P, \tag{28}$$

i.e., for $P \le N^{1/2}$, the speedup is linear with respect to P.

It is also worth mentioning that if the number of threads P is greater than the number of grid nodes in one direction and proportional, say, to $N^{1/2}$, then the values C_s and C_r can be decreased owing to "inner" parallelization of arithmetic operations for each of the grid lines (but this can again be done in the same fashion for both the multiplicative and additive ADI methods, whence the result of comparison of their parallel efficiency will not be altered).

The additive variant of the ADI method allows one to achieve a higher degree of parallelization (provided that the number of cores is sufficiently large) owing to computing the matrix-vector products occurring in (23) and (14) in parallel. Here, we take into consideration the fact that the coefficients z_k of the partial fraction decomposition of a rational matrix function are computed only once, prior to iterations, and the relevant arithmetic costs are independent of h and can be neglected.

The solution of a linear system by the additive ADI method can be split into two stages: computation of the residual vectors by formula (23) and computation of the approximate solutions u^n from (14). The most expensive operation of the first stage is the computation of a vector-matrix sum of the form

$$w = \sum_{l=1}^{n} z_l (s_l I - A_k)(s_l I + A_k)^{-1} v,$$

repeated twice.

Thus, the time for computing r^n in accordance with (23) is estimated as

$$T_a^{(1)} \cong\ 2[T_s + T_1 + C_1 \log_2 n(\varepsilon)], \tag{29}$$

where T_s is the same as in (26); T_1 is the time for multiplying a vector by $z_l(s_l I - A_k)$, and the last term is equal to the total summation costs by the binary reduction method on n cores. Note that for a fast computation of the approximate solutions u^n, all the vectors r^k,

$k = 0, 1, \ldots, n$, must be available. The latter vectors can indeed be computed simultaneously in the time $T_a^{(1)}$, but then a larger number of threads is required, i.e., $P \geq N^{1/2} n(\varepsilon)$.

The time for computing u^n in accordance with (14) is estimated as

$$T_a^{(2)} \cong T_s + T_1 + C_1 \log_2 n(\varepsilon). \tag{30}$$

Since the total solution time of the additive method is equal to $T_a = T_a^{(1)} + T_a^{(2)}$, we conclude that the speedup resulting from applying the additive ADI method is proportional to the ratio $n/\log_2 n$. Obviously, it is unreasonable to apply such an algorithm with one or a few threads because the required arithmetic and memory costs will increase considerably. Above, in (29), we have used the first of the two variants of partial fraction decomposition of a rational function. However, it is clear that for the second variant of the partial fraction decomposition, computed in accordance with (24)–(25), the result will be qualitatively the same.

5. Conclusion

In this paper, an approach to rising efficiency of program implementation of parallel ADI methods is suggested. We assume that the number of computing threads with shared memory can be increased. This approach is based on using the partial fraction decomposition of a rational matrix function. It is worth mentioning that the iteration parameters τ_n (or $\tau_n^{(1)}$ and $\tau_n^{(2)}$), as well as the convergence rate in the case of exact arithmetic, remain the same as for the classical Peaceman–Rachford algorithm.

For model grid problems, for which $n(\varepsilon) \sim \log h$, the speedup factor corresponding to passing from the multiplicative variant to the additive one equals $\log h^{-1}/\log(\log h^{-1})$ if the iteration parameters are optimal, and it increases if the parameters are not optimal.

Note that if the components of the vectors u and f in (1) are the values of the grid functions at the nodes of a regular rectangular grid, i.e.,

$$u = \{u_{i,j}\}, \quad f = \{f_{i,j}\}, \quad i = 1, \ldots, N_1, \quad j = 1, \ldots, N_2,$$

and the matrices A_1 and A_2 correspond to the model linear algebraic system and can be represented as

$$A_1 = \text{blockdiag}\{\bar{A}_1\}, \quad A_2 = \text{blockdiag}\{\bar{A}_2\},$$

where the blocks $\bar{A}_1$ and $\bar{A}_2$ are tridiagonal matrices of orders N_1 and N_2, then the original linear algebraic system can be written in the form of the following Sylvester matrix equation:

$$\bar{A}_1 U + U \bar{A}_2 = F, \qquad \bar{A}_1 \in \mathcal{R}^{N_1, N_1}, \qquad \bar{A}_2 \in \mathcal{R}^{N_2, N_2}, \qquad U, F \in \mathcal{R}^{N_1, N_2}. \tag{31}$$

As was shown in [5] and the references therein, if the matrix on the right-hand side of (31) is factorized as

$$F = F_1 F_2^T, \quad F_k \in \mathcal{R}^{N_k, m}, \quad m \ll N_k, \quad k = 1, 2,$$

then Eq. (31) can be solved by applying the following very efficient "Factorized" ADI method (FADI):

$$\begin{aligned} U^n &= U_1^n D^n (U_2^n)^T, \quad U_1^n = (U_1^1, \ldots, U_1^n) \in \mathcal{R}^{N_1, mn}, \\ D &= \text{diag}((s_1^{(1)} + s_1^{(2)}) I_m, \ldots, (s_n^{(1)} + s_n^{(2)}) I_m) \in \mathcal{R}^{mn, mn}, \\ U_2^n &= (U_2^1 \ldots U_2^n) \in \mathcal{R}^{N_2, mn}. \end{aligned} \tag{32}$$

Here, the subscript T denotes transposition; $s_n^{(1)}$ and $s_n^{(2)}$ are prescribed iteration parameters (the shifts of the spectra of $\bar{A}_1$ and $\bar{A}_2$); I_m is the identity matrix of order m, and approximations of the matrix factors $U_1^{(k)} \in \mathcal{R}^{N_1, m}$ and $U_2^{(k)} \in \mathcal{R}^{N_2 m}$, $k = 1, \ldots, n$, are computed via

the following recursive relations:

$$U_1^{(0)} = (A_1 + s_0^{(1)} I)^{-1} F_1, \quad U_1^{(n)} = \prod_{k=1}^{n} \widehat{R}_k(A_1) U_1^{(0)},$$
$$U_2^{(0)} = (F_2)^T (A_2 + s_0^{(2)} I), \quad U_2^{(n)} = (U_2^{(0)})^T \prod_{k=1}^{n} \check{R}_k(A_2), \tag{33}$$
$$\widehat{R}_k(z) = (z - s_{k-1}^{(1)})/(z + s_k^{(2)}), \quad \check{R}_k(z) = (z - s_{k-1}^{(1)})/(z + s_k^{(2)}).$$

Since, in the above formulas, the products of the rational matrix functions $\widehat{R}_k(A_1)$ and $\check{R}_k(A_2)$ can be represented as sums of partial fractions, it is nondifficult to construct a parallelizable additive variant of the FADI method. Note also that the quantities $U_1^{(n)}$ and $U_2^{(n)}$ in (33) can be computed independently and simultaneously.

This work was supported by the Russian Science Foundation (project No. 14-11-00485) and the Russian Foundation for Basic Research (project No. 14-07-00128).

Translated by L. Yu. Kolotilina.

REFERENCES

1. A. A. Samarskiĭ and E. S. Nikolaev, *Methods for Solving Grid Equations* [in Russian], Nauka, Moscow (1978).
2. V. P. Il'in, *Finite-Difference and Finite-Volume Methods for Elliptic Equations* [in Russian], IVM SO RAN, Novosibirsk (2000).
3. V. P. Il'in, *Methods and Tehnologies of Finite Elements* [in Russian], IVM SO RAN, Novosibirsk (2007).
4. V. P. Il'in and D. V. Knysh, "Parallel algorithms for solving separable boundary value problems," in: Int. Sci. Conf. "Parallel Computer Tehnologies," SPbPGU, St.Petersburg (2008), pp. 107–118.
5. P. Benner, Ren-Cang Li, and Nin Truhar, "On the ADI method for Sylvester equations," *J. Comput. Appl. Math.*, **233**, 1035–1045 (2009).