Acceleration of multiple solution of linear systems for analyses of microstrip structures

T.R. Gazizov, S.P. Kuksenko, and R.R. Akhunov

Abstract**—**Complexity of algorithms of LU-decomposition, BiCGStab and CGS methods, using arithmetic complexity and *O*-notation, has been estimated. Algorithms for multiple solution of linear systems by an iterative method with preconditioner recomputation have been developed; these algorithms are based on the analysis of complexity and arithmetic mean time of linear systems solutions. It has been revealed that the order of solution and the choice of matrixes for computing a preconditioner, significantly affect the resulting acceleration. A number of experiments on computation of 100 capacitive matrixes of two strip structures have been carried out.

Keywords**—**multiple solution, linear systems, iterative method, preconditioning, reformation, capacitive matrix, microstrip line, modal filter.

I. Introduction

The problem of the acceleration of multiple solution of linear systems with variable matrix obtained in various fields of scientific research and engineering applications, including recursive least squares computations, wave scattering problems, numerical methods for integral equations, image restorations and others, is being thoroughly investigated [1, 2]. Methods of multiple solution are also investigated, but with a constant matrix and with many different right hand side vectors [3–6]. Thus, it is currently important to improve methods of multiple solution of linear systems.

In practice we often need to calculate a number of capacitive matrices of strip structures when their parameters are variable, but it takes a lot of time [7]. In general terms, implementation of the method of moments can reduce this problem to the multiple solution of m linear systems of $\mathbf{A}_k\mathbf{x}_k=\mathbf{b}$ type with $k=1\div m$ and $\mathbf{A}_k$ being square and dense matrix of order N. Block LU-decomposition is suggested (and it has been proved effective) for cases with varied dielectric permittivity, for example [8]. However, if changes of structure parameters are petit, changes of matrix entries can also be insignificant, but they can be arbitrary located, in this case iterative methods are preferable to achieve acceleratation, for example, BiCGStab [9]. To accelerate the iterative process, papers [10, 11] suggest the following ways: an initial guess vector is equal to the solution vector of the previous linear system (a unit vector was used for solution of the first linear system); implicit preconditioning using matrix M derived from the first linear system by means of LU-decomposition.

However, there is a decline in the efficiency of the preconditioner, if the difference between the first and the latter matrices increases. This problem can be solved by the preconditioner adjustment. However, adjustment of the matrices **L** and **U** is rather complicated and almost always ineffective for implicit preconditioning [2], thus, it has not been considered in this paper. Preconditioner recomputation is the second approach. For example, [11] proposes to recompute the preconditioner, if the number of iterations is above a predetermined threshold, and an optimal threshold is determined (for which solution time of all linear systems is minimal). If the predetermined threshold is not optimal, the approach becomes inefficient. Therefore, it is important to determine the optimal threshold value, however it is impossible, until the solution is finished. It limits practical implementation of this approach and facilitates search for a condition determining the recomputation moment, which would not have this disadvantage. In addition, in order to obtain acceleration close to the maximum, the condition of the recomputation should adaptively take into account changes in the matrices. Also, it has not been researched yet how acceleration depends on a specific order (sequence) of solutions.

The aim of the present paper is to find approaches to improve multiple solution of linear systems in order to accelerate computation of a number of capacitive matrices of strip structures. To this end, it is necessary to develop algorithms for multiple solution of linear systems with variable matrix, to determine whether it is possible to obtain acceleration via setting an order of linear systems solution, to perform a computational experiment evaluating the effectiveness of the developed algorithms.

II. Multiple solution algorithms

The overall complexity of the solution (without recomputation of a preconditioner) of m linear systems (F_Σ)

Computations of the capacitance by the method of moments was carried out in TUSUR and supported by the Russian Science Foundation (project No. 14-19-01232); the development of accelerated algorithms for solving linear algebraic systems was supported by the Russian Ministry of Education and Science (contract No. 8.1802.2014/K) and the Russian Foundation for Basic Research (projects Nos. 14-07-31267 and 14-29-09254).

Authors are with Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia, e-mail: talgat@tu.tusur.ru, ksergp@mail.ru, arr1982@sibmail.com.

can be expressed in terms of arithmetic mean complexity of one system ($\overline{F}$): $F_\Sigma = m\overline{F}$. Then, the minimum complexity of the solution of m linear systems can be achieved by reducing $\overline{F}$. This can be achieved by controlling the current value of the arithmetic mean of the complexity of solution of k linear systems: $\overline{F_k} = F_{\Sigma k}/k$, where $F_{\Sigma k}$ – the complexity of the solution of k linear systems.

To determine the complexity, we propose to use the arithmetic complexity (Q) and O-notation. Also, the overall complexity of the solution is directly proportional to the solution time of all linear systems. Therefore, the minimum complexity of solution of m linear systems can be achieved by controlling the arithmetic mean time of the solution of k linear systems: $\overline{T}_k = T_{\Sigma k}/k$.

These approaches may be used with any iterative method. To confirm this, we shall consider two iterative methods: BiCGStab and CGS [12]. LU-decomposition was used to derive a preconditioner matrix **M**, as in [10]. To make it easier to understand, the algorithms of BiCGStab and CGS methods are given below.

Algorithm 1: BiCGStab

1. Choose some initial guess $\mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\,\mathbf{x}_0$
2. Choose $\tilde{\mathbf{r}}$, (for example, $\tilde{\mathbf{r}} = \mathbf{r}_0$)
3. For i from 1 to N_{it}^{max}
4. $\rho_{i-1} = (\tilde{\mathbf{r}}, \mathbf{r}_{i-1})$
5. If $\rho_{i-1} = 0$
6. method fails
7. If $i = 1$
8. $\mathbf{p}_i = \mathbf{r}_{i-1}$
9. Else
10. $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})\,(\alpha_{i-1}/\omega_{i-1})$
11. $\mathbf{p}_i = \mathbf{r}_{i-1} + \beta_{i-1}(\mathbf{p}_{i-1} - \omega_{i-1}\,\mathbf{v}_{i-1})$
12. Find the vector $\tilde{\mathbf{p}}$ from the equation $\mathbf{M}\tilde{\mathbf{p}} = \mathbf{p}_i$
13. $\mathbf{v}_i = \mathbf{A}\tilde{\mathbf{p}}$
14. $\alpha_i = \rho_{i-1}/(\tilde{\mathbf{r}}, \mathbf{v}_i)$
15. $\mathbf{s} = \mathbf{r}_{i-1} - \alpha_i\,\mathbf{v}_i$
16. If $\|\mathbf{s}\|_2 / \|\mathbf{r}_0\|_2 \le Tol$
17. then STOP
    ($\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\tilde{\mathbf{p}}$ – vector of solution)
18. Find the vector $\tilde{\mathbf{s}}$ from the equation $\mathbf{M}\tilde{\mathbf{s}} = \mathbf{s}$
19. $\mathbf{t} = \mathbf{A}\tilde{\mathbf{s}}$
20. $\omega_i = (\mathbf{t}, \mathbf{s}) / (\mathbf{t}, \mathbf{t})$
21. $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\tilde{\mathbf{p}} + \omega_i\tilde{\mathbf{s}}$
22. $\mathbf{r}_i = \mathbf{s} - \omega_i\,t$
23. If $\|\mathbf{r}\|_2/\|\mathbf{r}_0\|_2 \le Tol$
24. then STOP
    ($\mathbf{x}_i$ – vector of solution)
25. Set $i = i+1$

Algorithm 2: CGS

1 Choose some initial guess $\mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
2 Choose $\tilde{\mathbf{r}}$, (for example, $\tilde{\mathbf{r}} = \mathbf{r}_0$)
3 For i from 1 to N_{it}^{max}
4 $\rho_{i-1} = (\tilde{\mathbf{r}}, \mathbf{r}_{i-1})$
5 If $\rho_{i-1} = 0$
6 method fails
7 If $i = 1$
8 $\mathbf{u}_1 = \mathbf{r}_0$
9 $\mathbf{p}_1 = \mathbf{u}_1$
10 Else
11 $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})$
12 $\mathbf{u}_i = \mathbf{r}_i + \beta_{i-1}\mathbf{q}_{i-1}$
13 $\mathbf{p}_i = \mathbf{u}_i + \beta_{i-1}(\mathbf{q}_{i-1} + \beta_{i-1}\mathbf{p}_{i-1})$
14 Find the vector $\tilde{\mathbf{p}}$ from the equation $\mathbf{M}\tilde{\mathbf{p}} = \mathbf{p}_i$
15 $\tilde{\mathbf{v}} = \mathbf{A}\tilde{\mathbf{p}}$
16 $\alpha_i = \rho_{i-1}/(\tilde{\mathbf{r}}, \tilde{\mathbf{v}})$
17 $\mathbf{q}_i = \mathbf{u}_i - \alpha_i\tilde{\mathbf{v}}$
18 Find the vector $\tilde{\mathbf{u}}$ from the equation $\mathbf{M}\tilde{\mathbf{u}} = \mathbf{u}_i + \mathbf{q}_i$
19 $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\tilde{\mathbf{u}}$
20 $\tilde{\mathbf{q}} = \mathbf{A}\tilde{\mathbf{u}}$
21 $\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i\tilde{\mathbf{q}}$
22 If $\|\mathbf{r}\|_2/\|\mathbf{r}_0\|_2 \le Tol$
23 then STOP ($\mathbf{x}_i$ – vector of solution)
24 Set $i = i+1$

Two possible conditions of exit from the iteration process (lines 17 and 24) were taken into account when computing complexity of the algorithm BiCGStab. CGS algorithm has only one check of the iteration hault. The analytic estimations of complexity for LU-decomposition (used to compute the preconditioner), BiCGStab and CGS, taking into account specifics of software implementation, as recommended in [13], are shown in Table I. (Derivation of expressions is omitted.)

TABLE I. ESTIMATIONS OF ALGORITHMS COMPLEXITY

Algorithm	Arithmetic complexity (Q)	O-notation
LU-decomposition	$f_{LU}(N) = (20N^3 - 6N^2 + 32N - 36)/12$	$f_{LU}(N) = N^3/6$
BiCGStab, condition 1 (exit in line 17)	$f(N, N_{it}) = 5N^2 + 13N + 3 + N_{it}(10N^2 + 31N + 16) + (N_{it} - 1)(10N^2 + 33N + 11)$	$f(N, N_{it}) = 4N^2 + 6N + (N_{it} - 1)(4N^2 + 8N)$
BiCGStab, condition 2 (exit in line 24)	$f(N, N_{it}) = 5N^2 + 8N + 2 + N_{it}(20N^2 + 64N + 37)$	$f(N, N_{it}) = 2N^2 + 2N + N_{it}(4N^2 + 8N)$
CGS	$f(N, N_{it}) = 10N^2 + 16N + 1 + N_{it}(20N^2 + 43N + 15) + (N_{it} - 1)(10N + 4)$	$f(N, N_{it}) = 2N^2 + N_{it}(8N^2 + 5N)$

Then, the algorithms for multiple solution of linear systems with recomputation of a preconditioner, performed with accordance to the estimates from Table I and on the basis of the arithmetic mean time of solution of k linear systems ($\overline{T}_k$) are given.

Algorithm 3: Multiple solution of *m* linear systems with recomputation of a preconditioner, when the average complexity of solution of *k* linear systems increases

1 Compute matrix **M** from matrix $\mathbf{A}_1$ using LU-decomposition
2 $F_\Sigma = f_{LU}(N)$
3 Find $\mathbf{x}_1$ from equation $\mathbf{M}\mathbf{A}_1\mathbf{x}_1 = \mathbf{M}\mathbf{b}$ with given accuracy Tol

4 Save number of iterations into N_{it}
5 $F_k = f(N, N_{it})$
6 $F_\sum = F_\sum + F_k$
7 For k from 2 to m
8 Find $\mathbf{x}_k$ from equation $\mathbf{MA}_k\mathbf{x}_k = \mathbf{Mb}$ with given accuracy *Tol*
9 Save number of iterations into N_{it}
10 $F_k = f(N, N_{it})$
11 If $F_\sum / (k-1) < (F_\sum + F_k)/ k$
12 Compute matrix **M** from matrix $\mathbf{A}_k$ using LU-decomposition
13 $F_\sum = F_\sum + f_{LU}(N)$
14 $F_\sum = F_\sum + F_k$
15 Set k=k+1

Algorithm 4: Multiple solution of *m* linear systems with recomputation of a preconditioner, when the average mean time of solution of *k* linear systems increases

1 Compute preconditioning matrix **M** from matrix $\mathbf{A}_1$
2 Save computational time into T_{PR}
3 Find $\mathbf{x}_1$ from equation $\mathbf{MA}_1\mathbf{x}_1 = \mathbf{Mb}$ with given accuracy *Tol*
4 Save computational time into $\mathbf{T}_1$
5 $T_\sum = T_{PR} + T_1$
6 For k from 2 to m
7 Find $\mathbf{x}_k$ from equation $\mathbf{MA}_k\mathbf{x}_k = \mathbf{Mb}$ with given accuracy *Tol*
8 Save computational time into T_k
9 If $T_\sum /(k-1) < (T_\sum + T_k)/k$
10 Compute preconditioning matrix **M** from matrix $\mathbf{A}_k$
11 Save computational time into T_{PR}
12 $T_\sum = T_{PR}$
13 Else
14 $T_\sum = T_\sum + T_k$
15 Set k=k+1

III. CHOICE OF SOLUTIONS ORDER

Certain order of solutions of linear systems can be a way to accelerate solution. Indeed, this sequence is usually determined by the given change in the structure parameter. But if the total time of all linear systems solution depends on which linear system will be the first, the second and so on, then there is an optimal sequence providing the minimum time of solution. The fact of such dependence follows from the very essence of multiple solution of linear systems by iterative method with preconditioning. Obviously, it is determined by two factors: choice of a matrix to calculate the preconditioner (the matrix affects the number of iterations required); use of the solutions of the previous linear system as an initial guess of the current linear system (the closer to the solution the initial guess is, the less iterations are required) [10].

Multivariant analysis uses a few basic types of parameter changes: linear, logarithmic, with user-defined values. During optimization a change may be random, in any direction. Let us consider the most simple, but widely used linear change. It implies simple order of solutions, i.e. in order of parameter ascending (direct order) or descending (reverse order).

We shall note that linear variation of parameter does not guarantee monotonic change of matrix elements of linear system or its norm, but it is frequently used in practice. In any case, it is useful to analyze particular structures. To assess changes in the matrices, norms were used: $\|\Delta\mathbf{A}_{i,j}\|_1$ and $\|\Delta\mathbf{A}_{i,j}\|_\infty$, where $\Delta\mathbf{A}_{i,j}$ – variation matrix ($\Delta\mathbf{A}_{i,j}=\mathbf{A}_i-\mathbf{A}_j$), i and j – sequence numbers of the compared matrices (i, j=1÷m, m – total number of linear systems). The analyzed matrices were obtained in the TALGAT software [14] using mathematical models based on the method of moments [15]. The structure (Fig. 1 (b)), being a symmetric modal filter with front coupling has been used [16]. The number of segments on each boundary of the structure has not changed in order to provide constant N. Matrices with N=2001 are derived by changing the gaps (s) in the range. Changes were made in the direct (100, 101, ..., 199 μm) and reverse (199, 198, ..., 100 μm) orders.

Dependences of the relative norms of the variation matrix for the direct and reverse orders on the number of the linear system being solved are shown in Fig. II. As can be seen, the nature of the changes is the same for both norms, but a monotonic rise at a decreasing rate can be seen for the direct order (better demonstrated at the beginning and less - at the end of the range), and there is opposite nature of the dependencies for reverse order.

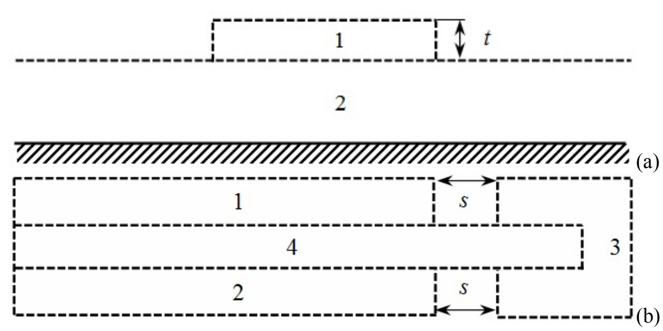

Fig. I Cross-section of examined structures: 1 – conductor (1) on a dielectric substrate (2) over a perfectly conducting plane (a); 2 – modal filter (conductors 1, 2 and 3, dielectric 4 is placed between them) (b)

As mentioned, there is a dependence of the preconditioner efficiency on the matrices changes. Changes in the reverse order for almost all i are less than for the direct order. Therefore, in the reverse order the preconditioner efficiency increases due to the smaller and more gradual changes in the linear system matrix. Thus, to accelerate the solutions, it is better firstly to solve linear systems with smaller changes between matrices and then - with large. To do this, for each matrix we should define a close matrix with minimal changes, i.e. to find $\min_j \|\Delta A_{i,j}\|_1$ or $\min_j \|\Delta A_{i,j}\|_\infty$ and then sort in the order of ascending changes. However, this search is time-consuming, as it requires to find a variation matrix m^2 times. Therefore, this approach is not suitable for practice.

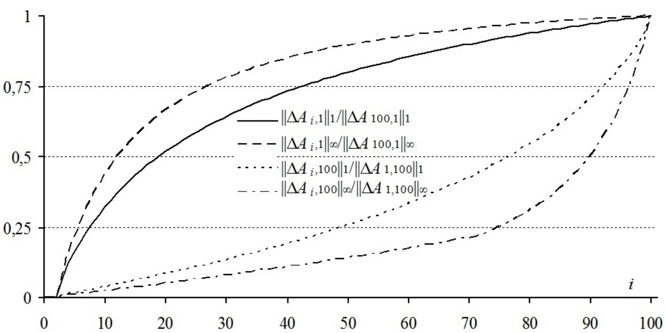

Fig. II Dependences of the relative norms of variation matrix on number of linear system being solved for direct and reverse orders of solution

However, in this case there is a linear variation of the parameter that allows, changing only the order of the solution (direct or reverse), to identify the optimum. On the basis of the approach, proposed to determine the optimal order of the multiple solution of linear systems, a general algorithm has been proposed:

1 Set order of linear systems solution
2 Compute preconditioner matrix **M** from matrix $\mathbf{A}_1$
3 For i from 1 to m
4 Find $\mathbf{x}_i$ from equation $\mathbf{MA}_i\mathbf{x}_i = \mathbf{Mb}$ with given accuracy *Tol*
5 Set $i = i+1$

IV. Computational Experiment

We used a personal computer (parallelization was not exploited, i.e., one core of the processor was busy) with the following parameters: platform – Intel(R) Core (TM) i7; processor frequency – 2.80 GHz, memory – 12 Gb; number of cores – 8; operating system – Windows 8x64. We formed 100 matrices for each of the cases. Two structures were considered. For structure 1 (Fig. I (a)) matrices of order 1600 are obtained by varying the height of the conductor (t) in the range 6, 7, …, 105 µm. For structure 2 (Fig. I (b)) matrices of order 2001 are obtained by varying gaps (s) in the range 100, 101, …, 199 µm.

Obtained accelerations of solutions of 100 linear systems via proposed algorithms for structures 1 and 2 with respect to the algorithm without recomputation of a preconditioner (for each iteration method separately) are summarized in Table II. The results obtained while recomputating a preconditioner by setting an optimum threshold for iterations number (N_{it}^{Opt}) are also given [11].

Table II Acceleration of solution of 100 linear systems by BiCGStab and CGS methods with recomputation of a preconditioner

Number of strucrure		**1**		**2**	
N		**1600**	**3200**	**2001**	**3001**
Optimum threshold of iterations number (N_{it}^{Opt})	BiCGStab	1.42 (8)	1.16 (8)	1.62 (10)	1.58 (10)
	CGS	1.34 (7)	1.14 (7)	1.44 (8)	1.31 (9)
O-notation	BiCGStab	1.20	0.92	1.60	1.55
	CGS	1.52	1.27	0.94	0.87
Arithmetic complexity	BiCGStab	1.12	0.86	1.52	1.44
	CGS	1.40	1.01	1.05	0.95
Arithmetic mean time	BiCGStab	1.32	1.00	1.60	1.46
	CGS	1.09	1.19	1.41	1.23

An algorithm with recomputation of a preconditioner by setting the optimal threshold of iterations number gives maximum acceleration (except for the structure 1 and the use of CGS). Acceleration of multiple solution of linear systems, obtained by proposed algorithms, is not constant. For example, BiCGStab method for structure 1 showed deviation of acceleration values up to 26%, while for the structure 2 – not more than 9%. For structure 1 CGS showed deviation of acceleration values up to 19%, and for structure 2 – up to 35%. Such variations are explained by the fact that the recomputation of a preconditioner occurs at different moments. Thus, during the computational experiment, it was observed that the speed of the linear systems solution is strongly influenced by recomputation, i.e. for which k_p it occurs. Table III gives recomputation moments for each test. We see that for a later recomputation of a preconditioner the acceleration decreases. It was also observed that the closer is the moment of recomputation, the smaller acceleration is obtained.

Table III Moment of recomputation (K_P) of a preconditioner for given algorithms

Number of strucrure		**1**		**2**	
N		**1600**	**3200**	**2001**	**3001**
Optimum threshold of iterations number (N_{it}^{Opt})	BiCGStab	37	40	33	37
	CGS	23	27	32	31
O-notation	BiCGStab	37	54	37	37
	CGS	19, 53	27	18, 45, 71	31, 74
Arithmetic complexity	BiCGStab	60, 99	65	49	48
	CGS	53	81	56	57
Arithmetic mean time	BiCGStab	54	67	37	48
	CGS	54	55	42	57

Further the impact of the order of solutions made on acceleration of multiple solution of linear systems was evaluated. Table IV summarizes accelerations obtained by using reverse order of linear systems solution relatively to the direct order (without recomputation of a preconditioner).

Table IV Acceleration of solution of 100 linear systems by BiCGStab and CGS methods with optimal order of linear systems solution

Number of strucrure	**1**		**2**	
N	**1600**	**3200**	**2001**	**3001**
BiCGStab	1.76	1.63	1.71	1.84
CGS	1.73	1.66	1.58	1.53

It is seen that acceleration is obtained for all structures using the reverse order of linear systems solutions (up to 1.84 for structure 2 for N = 3001, BiCGStab method), and it is obtained for all N and for both iterative methods.

Acceleration is obtained due to the difference in the number of iterations required for solution of linear systems in the direct (N^{+}_{it}) and reverse (N^{-}_{it}) orders, expressed in square of figure bounded by the curves of the number of iterations (Fig III). For both structures the solution of linear systems in direct order requires more iterations than reverse. The main reason for this is that linear systems, from which a preconditioner is obtained, are different: in the direct order a preconditioner is obtained from the first linear system, and in the reverse - from the 100th. A different degree of changes in the linear systems matrix at the beginning (strong) and the end (weak) of the range also affected the required number of iterations. The results confirm the hypothesis about the impact of the order of solution of linear systems on the efficiency of the solution; it shows that in order to achieve greater acceleration, we should choose a matrix for the calculation of a preconditioner from the middle of the range. To this end, we propose a general algorithm:

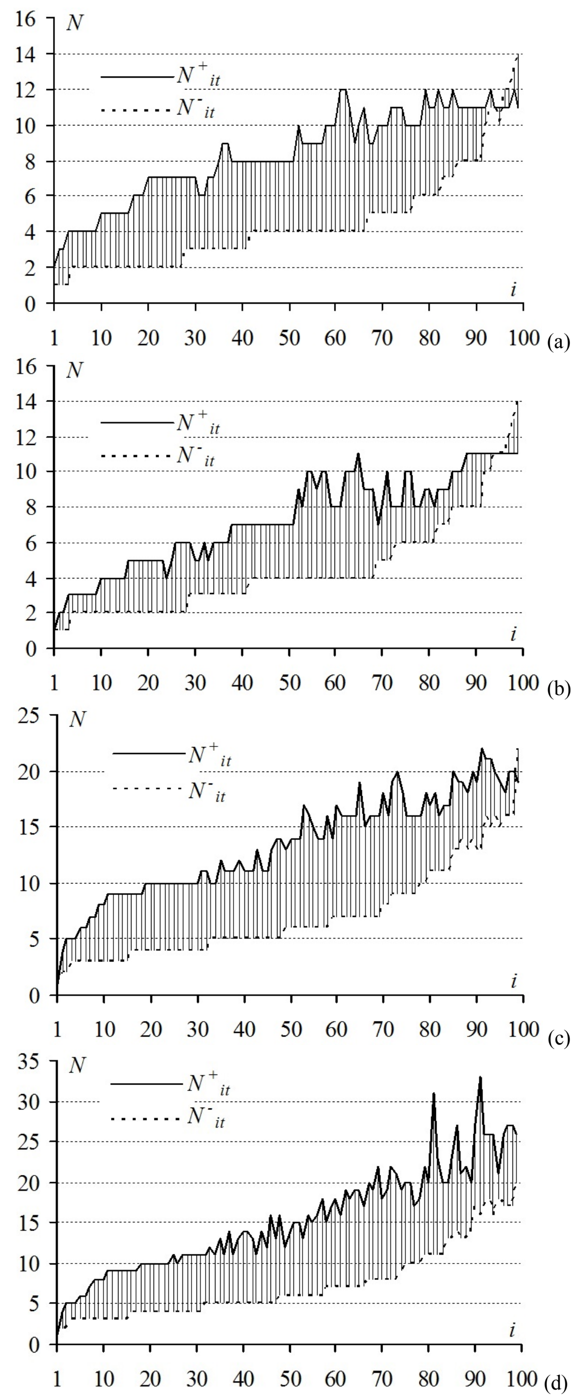

Fig. III The number of iterations for multiple solution by iterative BiCGStab method in direct (N^{+}_{it}) and reverse (N^{-}_{it}) orders for structures: 1 – for N=1600 (a), 3200 (b); 2 – for N=2001 (c), 3001 (d)

1 Choose matrix $\mathbf{A}_{op}$
2 Set solution order
3 Compute preconditioning matrix $\mathbf{M}$ from matrix $\mathbf{A}_{op}$
4 For i from 1 to m
5 Find x_i from equation $\mathbf{MA}_i\mathbf{x}_i = \mathbf{Mb}$ with given accuracy Tol
6 Set $i = i+1$

Calculation results (acceleration) using the BiCGStab method, when the 50th matrix was chosen (k=50) for forming of a preconditioner for both structures are summarized in Table V. It can be seen that in this case, the speed up is about two times comparing to the solution without a recomputation. Thus, this method is most advantageous. Obviously, the optimal choice of the matrix to calculate a preconditioner can give additional acceleration, but it is difficult to make it. It is worth noting that in this case, as the first study, we chose one of the possible orders of solutions: after the calculation of the preconditioning matrix, solution of systems starts in turn from 1 to 100 (direct order). In the future, it is necessary to investigate the remaining options, as well as to develop an algorithm to select the optimal matrix for the calculation of a preconditioner.

TABLE V ACCELERATION OF SOLUTION OF 100 LINEAR SYSTEMS BY BICGSTAB METHOD CHOOSING THE 50TH MATRIX TO CALCULATE A PRECONDITIONER

Number of strucrure	1		2	
N	1600	3200	2001	3001
Acceleration	2.14	1.94	2.07	2.21

V. Conclusion

The complexity of LU-decomposition algorithms and BiCGStab and CGS methods has been evaluated using O-notation and arithmetic complexity accounting for specifics of software. We propose the algorithms for multiple solution of linear systems by iterative method in which the recomputation of the preconditioning matrix occurs simultaneously with the increase of arithmetic mean complexity (based on estimates of the LU-decomposition and the iterative method complexity) and the arithmetic mean time for linear systems solution. The advantage of the algorithms is the possibility of their use in different iterative methods of Krylov type. We have investigated solutions of the linear systems with linear change of one of the dimensions of the analyzed structures in direct and reverse orders, and revealed that the reverse order is more preferable for solutions. The approbation of the proposed algorithms with a significant change of linear system matrix entries was performed on the example of computation of capacitive matrices of two strip structures. The obtained accelerations, compared to the solution without recomputation, in most cases are close to the optimal. The advantage of these algorithms is their adaptability to changes in the matrix, which allows to determine the time of preconditioner recomputation without user intervention. It has been shown possible to obtain additional acceleration by selecting from the middle of a range a matrix for calculating a preconditioner, which is used in multiple solutions.

References

[1] S. Karimi, "A new iterative solution method for solving multiple linear systems", *Advances in Linear Algebra and Matrix Theory*, vol. 2 no. 3, pp. 25-30, 2012.

[2] C. Calgaro, J.P. Chehab, Y. Saad, "Incremental incomplete LU factorizations with applications to time-dependent PDEs", *Numer. Lin. Algebra with Appl*., no. 17(5), pp. 811–837, 2010.

[3] G. Golub and C. Loan, "Matrix computations", 2nd Edition, Johns Hopkins Press, Bultimore, 1989.

[4] S. Karimi and F. Toutounian, "The block least squares method for solving nonsymmetric linear systems with multiple right-hand sides", *Applied Mathematics and Computation*, vol. 177, no. 2, pp. 852-862, 2006.

[5] A. El Guennouni, K. Jbilou, H. Sadok, "A block version of Bicgstab for linear systems with multiple right-hand sides", *Electronic Transactions on Numerical Analysis*, vol. 16, pp. 129-142, 2003.

[6] V. Simoncini, E. Gallopoulos, "An iterative method for nonsymmetric systems with multiple right-hand sides", *SIAM J. Sci. Comput.*, no. 16(4), pp. 917-933, 1995.

[7] A. Devi, M. Gandhi, K. Varghese, D. Gope, "Hardware accelerator for 3D method of moments based parasitic extraction", The 2013 IEEE Electrical Design of Advanced Packaging & Systems (EDAPS) symposium, December 2013, Nara, Japan. pp. 100-103.

[8] R.S. Surovtsev, S.P. Kuksenko., T.R. Gazizov, Analytic evaluation of the computational costs for solving systems of inear algebraic equations in multiple computing of the capacitance matrix in a range of the dielectric permittivity of dielectrics, *J. of mathematical sciences*, vol. 207, no. 5, pp. 795-802, 2015.

[9] H. Van der Vorst, "Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for solution of nonsymmetric linear systems", *SIAM J. Scientific and Statistical Computing*, no. 13, pp. 631–644, 1992.

[10] R.R. Akhunov, S.P. Kuksenko, T.R. Gazizov, "Acceleration of multiple iterative solution of linear algebraic systems in computing the capacitance of a microstrip line in wide ranges of its sizes", *J. of mathematical sciences*, vol. 207, no. 5, pp. 686-692, 2015.

[11] R.R. Akhunov, S.P. Kuksenko, T.R. Gazizov, "Multiple solution of systems of linear algebraic equations by an iterative method with recomputed preconditioners", *J. of mathematical sciences*, vol. 207, no. 5, pp. 693-697, 2015.

[12] P. Sonneveld, "CGS, a fast Lanczos-type solver for nonsymmetric linear systems", *SIAM J. Sci. Statist. Comput.*, no. 10, pp. 36-52, 1989.

[13] J.J. McConnell, "Analysis of algorithms: an active learning approach", Jones & Bartlett Learning, 2001.

[14] S.P. Kuksenko, A.M. Zabolotsky, A.O. Melkozerov, T.R. Gazizov, "New features of electromagnetic compatibility in TALGAT simulation software", *Dokl. Tomsk. Univ. Sist. Upr.Radioelektr*, no. 2(36), pp. 45-50, 2015.

[15] T.R. Gazizov, "Analytic expressions for Mom calculation of capacitance matrix of two dimensional system of conductors and dielectrics having arbitrary oriented boundaries", Proc. of the 2001 IEEE EMC Sympo., Montreal, Canada, 2001. vol. 1. pp. 151-155.

[16] A.M. Zabolotsky, T.R. Gazizov, A.O. Melkozerov, P.E. Orlov, E.S. Dolganov, "Improved design of modal filter for electronics protection", Proc. of 31-th Int. conf. on lightning protection. Vienna, Austria. September 2–7, 2012, pp. 1–4.