

PAPER • OPEN ACCESS

Choosing order of operations to accelerate strip structure analysis in parameter range

To cite this article: S P Kuksenko *et al* 2018 *J. Phys.: Conf. Ser.* **1015** 032076

View the [article online](#) for updates and enhancements.

Related content

- [On the canonical strip](#)
Vasilii V Golyshev
- [Chalcogenide waveguide structure for dispersion in mid-infrared wavelength](#)
Nandam Ashok, Yeung Lak Lee and WooJin Shin
- [Spin-fluctuation-mediated pairing symmetry on the metallic kagome lattice](#)
Jing Kang, Shun-Li Yu, Zi-Jian Yao et al.

Choosing order of operations to accelerate strip structure analysis in parameter range

S P Kuksenko, R R Akhunov, T R Gazizov

Tomsk State University of Control Systems and Radioelectronics, 40, Lenin Ave., Tomsk, 634050, Russia

E-mail: ksergp@mail.ru

Abstract. The paper considers the issue of using iteration methods in solving the sequence of linear algebraic systems obtained in quasistatic analysis of strip structures with the method of moments. Using the analysis of 4 strip structures, the authors have proved that additional acceleration (up to 2.21 times) of the iterative process can be obtained during the process of solving linear systems repeatedly by means of choosing a proper order of operations and a preconditioner. The obtained results can be used to accelerate the process of computer-aided design of various strip structures. The choice of the order of operations to accelerate the process is quite simple, universal and could be used not only for strip structure analysis but also for a wide range of computational problems.

1. Introduction

Distributed circuits based on various strip structures are widely used in radioelectronic equipment, both as transmission lines that maintain proper characteristics for desired signals for a long time and as a basis for new protective devices. A strip structure consists of signal and ground conductors and a dielectric substrate. The separation between conductors, their thickness, other geometrical parameters and dielectric permittivity of a substrate can be repeatedly changed during simulation and optimization of elements and devices. These processes significantly increase computational costs. Generally, hardware accelerators (multicore workstations, clusters, graphical processing units) are used to decrease computational costs, while algorithmic methods are often ignored. A quasistatic approach, which is based on calculating electric capacitance with the method of moments [1], is widely used to decrease computational costs of strip structure analysis in contrast to the electrodynamic approach. The method of moments implies solving linear systems with dense matrix. This paper presents the research into application of iterative methods for solving dense linear systems repeatedly during multivariant analysis of strip structures using the method of moments with the optimal choice of order of operations.

2. Approaches to solving the sequence of linear systems

Some tasks are time-consuming and require linear systems to be repeatedly solved. If all components of a linear system are changed, the task is reduced to solving the equation:

$$\mathbf{A}_k \mathbf{x}_k = \mathbf{b}_k, k = 1, 2, \dots, m \quad (1)$$



where \mathbf{A}_k are general nonsingular matrices, \mathbf{b}_k are corresponding right-hand sides, k is a sequence number of linear systems, m is a total number of linear systems. It is required to solve equations of type (1) in various applications, for example, in methods that use recursive computation of least squares, in image restoration, applied statistics, optimization etc.

There are several approaches to solve the sequence (1) by means of an iterative method when the system has sparse matrix. The first approach is based on recomputing a preconditioner from scratch for each matrix of a sequence. The second approach is based on computing a preconditioner from the first matrix of a sequence and its further use in solving other systems (frozen preconditioner) [2]. The third approach consists in updating a preconditioner obtained from the matrix of one of the systems (seed preconditioner), and in repeating its updates when necessary [3, 4]. The fourth approach is based on periodic recomputation of a preconditioner. The fifth approach is equivalent to the previous one, however, if, during some period, the number of iterations required to solve the current system is bigger than the sum of iterations required to solve the first system in this period and a predetermined threshold, then the preconditioner is updated [5]. The last approach is based on the adaptive use of information about Krylov subspaces obtained in the previous steps [6].

The tasks with some fixed components of linear systems represent a special case of solving linear systems repeatedly [7]. In this case, the task is simplified and can be solved by means of block versions of Krylov-type iterative methods. To solve sparse matrices, there are the following methods: block BiCGStab, block GMRES, GL-LSQR, MHGMRES(m), MEGCR, and others [8]. Typically, if an effective preconditioner is used, block methods are preferred rather than solving linear systems sequentially with different right-hand sides. However, if all right-hand sides are unavailable at the same time, these methods are not applicable.

There are tasks that require solving linear algebraic systems of type $\mathbf{A}_k \mathbf{x}_k = \mathbf{b}$ for $k = 1, 2, \dots, m$ multiple times. An example of such problem is the time-consuming computation of the capacitance matrix for strip structures with varied parameters (\mathbf{A}_k is a square and dense matrix of N order) [9] with boundary element methods [10], also called the method of moments in electromagnetics [11], and the panel method in fluid dynamics. The linear system matrix in the method of moments has a much smaller order in comparison with the time-domain finite-difference, finite element, finite integration, and transmission line matrix methods, which are also used in electromagnetic tasks. The main drawback of the numerical methods in general and particularly of the methods mentioned above is that computational costs and especially solution time increase substantially when a structure under analysis becomes more complicated. Therefore, it is important to develop new approaches to speed up such calculations, both via hardware [12] and algorithms.

Traditionally, linear systems with a dense matrix are solved by means of Gaussian elimination or its compact model, based on the LU-decomposition. For example, the block LU decomposition is used to calculate the capacitance matrix derived from the same structure when the permittivity value of dielectrics is changed. However, in general case, changing the sizes of a structure results in changing the irregularly located matrix entries, so it is preferable to use iterative methods to accelerate the solving process. For example, the iterative method BiCGStab showed a significant speed-up in comparison with Gaussian elimination. To accelerate the iterative process, two methods were used: in the first one, the computed solution of the previous system is used as the initial guess of the current linear algebraic system; in the second method, implicit preconditioning matrix \mathbf{M} , computed from the matrix of the first linear system, is used to solve the current system. However, the effectiveness of the preconditioner decreases, as the difference between the first and the current matrix increases. To solve this problem, it was suggested to recompute matrix \mathbf{M} when the convergence rate of solving the current linear system is too slow [13]. Existence of optimal threshold value was shown. However, it is *a priori* impossible to determine when to recompute matrix \mathbf{M} . Thus, the search for an *a priori* condition of the recomputation is relevant. Paper [13] is devoted to one of the conditions. Other conditions proposed by the authors have been investigated in paper [14], so they are omitted in this paper. Summarizing all mentioned above, the search for the optimal order of operations to solve linear

systems is currently important. In this paper, let us make an attempt to fill this gap and continue previous research [14] generalizing the recent results and presenting some new ones.

3. Choosing the optimal order of operations to solve linear systems

A specific order of operations to solve linear systems can accelerate the total system-solving time. An order is usually predetermined by the desired change of structure parameters. If the total system-solving time depends on the order of operations, then there is an optimal order that provides the minimum system-solving time. The very principle of solving linear systems repeatedly using the iterative method with a preconditioner proves the existence of such order. The order of operations is determined by two factors: the choice of a matrix for recomputing a preconditioner, and the use of a previous solution vector as an initial guess for the current solution. Multivariate analysis may include the following types of parameter variation: linear, logarithmic, and user-defined. Variation of the parameters during optimization can be random in any direction.

Let us consider the simplest but widely used linear variation. In such case, there are two options of system-solving order: with the increase (straight) and a decrease (reverse) of the parameter. It is worth noting that linear parameter variation does not always provide monotonous change of linear system matrix entries. Anyway, it is useful to analyse some particular structures. Let us consider linear parameter variation in order to obtain the optimal solution by means of straight or reverse order of operations. The first results obtained for two structures [14] showed the promise of this approach.

In order to test the proposed approach, the authors formed 100 matrices for 4 structures (Figure 1). For structure 1 (Figure 1a) the matrices of orders $N = 1600$ were obtained by varying the thickness of the conductor (t) in the range of 6, 7 ... 106 μm . The matrices of linear systems change more significantly with changing t than with changing the substrate thickness (h) or the conductor width (w); therefore further the authors only considered different t . The number of segments on the structure boundaries was not changed, and all the systems were of the same order 1600. Then matrices of order 3200 were obtained by means of a denser segmentation. For structure 2 (Figure 1b), which is a modal filter (MF) with broad-side coupling, the matrices of orders 2001 and 3001 were obtained by varying the gaps (s) in the range of 100, 101 ... 200 μm . For structure 3 (Figure 1c), being a mirror MF [15], matrices with $N = 1709$ and 3109 were obtained by means of changing s in the range of 7.1, 7.2, ..., 16.9 μm . For structure 4 (Figure 1d), being an asymmetrical MF [16], matrices with $N = 1901$ and 2851 were obtained by means of changing s in the range of 5, 10, ... 500 μm . Comparison of the matrices revealed that the matrix of structure 1 has much lower (by three orders of magnitude) condition number, and structure 2 yields matrices of the worst conditioning. If the order of a matrix is doubled, then the condition number is doubled as well.

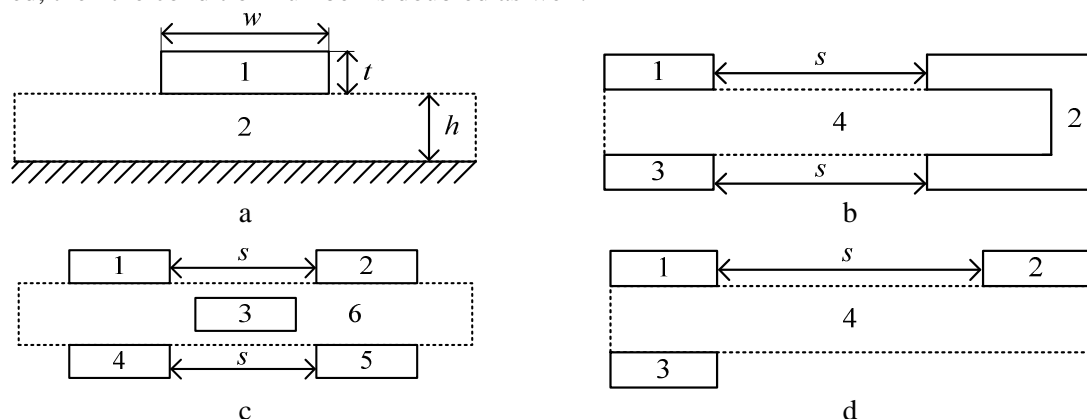


Figure 1. Cross-sections of the examined structures: (a) 1 – microstrip line (1 – conductor, 2 – dielectric); (b) 2 – symmetrical MF with broadside coupling (1–3 – conductors, 4 – dielectric); (c) 3 – mirror symmetrical MF (1–5 – conductor, 6 – dielectric); (d) 4 – asymmetrical MF with broadside coupling (1–3 – conductors, 4 – dielectric).

Table 1 gives accelerations (columns of BiCGStab and CGS methods with a reverse order of operations relatively to the straight order) obtained for the reverse order of operations to solve linear systems and compared to the straight order during the analysis of structures 1–4. Hereinafter, the solution to the previous system is used as an initial guess (unit vector is used for the first system); iterations are carried out till a residual norm is more than 10^{-8} . The solution of all structures is accelerated during the reverse order of operations, plus for all N and both iterative methods. The acceleration is explained by a different number of iterations required to solve the systems in the straight (N_{it}^+) and reverse (N_{it}^-) orders [14]. The process of solving linear systems in the straight order of operations requires more iterations than in the reverse order for all types of structures. It is explained by different linear system matrices that are used for computing a preconditioner; in the straight order of operations, a preconditioner is obtained from the 1st linear system, while in the reverse order it is obtained from the 100th linear system. Different degrees of matrix change in the beginning of a range (strong) and in the end (weak) it also influences the number of iterations. The obtained results confirm that the order of operations to solve linear systems affects the total system-solving time, moreover it has a significant effect: the authors have obtained average acceleration by a factor of 1.5 and the maximum of 1.84.

Table 1. Acceleration of solving 100 linear systems

Structure	N	BiCGStab and CGS with the reverse order of operations relatively to the straight order		BiCGStab with 50 th matrix used for preconditioning			
		BiCGStab	CGS	Variant of order			
				I	II	III	IV
1	1600	1.76	1.73	2.16	2.16	2.16	2.14
	3200	1.63	1.66	1.95	1.94	1.96	1.94
2	2001	1.71	1.58	2.05	2.07	2.07	2.07
	3001	1.84	1.53	2.19	2.21	2.20	2.21
3	1709	1.82	1.59	1.81	1.80	1.80	1.81
	3109	1.83	1.32	1.85	1.86	1.86	1.86
4	1901	1.72	1.61	1.85	1.83	1.84	1.82
	2851	1.76	1.63	1.91	1.90	1.91	1.89

A preconditioner influences the total time required to solve linear systems repeatedly. As a rule, the structure parameter is changed within a predetermined range, so to compute a preconditioner one can choose the proper parameter value. The easiest way is to choose a mean value of a parameter. To evaluate how the choice of a preconditioner matrix and the optimal order of operations (straight or reverse) influenced the process of system-solving, the authors have carried out a computational experiment using the BiCGStab method. The 50th matrix was chosen to compute a preconditioner, and 4 different variants given in Table 2 were used to determine the optimal order of operations.

Table 2. Orders to solve 100 linear systems with the 50th matrix used for preconditioning

Variant	Order to solve linear systems	
	1–49	50–100
I	←	←
II	→	←
III	←	→
IV	→	→

Table 1 (columns of the BiCGStab method for different variants of order with 50th matrix used for the preconditioner) contains the obtained accelerations compared to the 1st matrix for computing the preconditioner. It is clear that system-solving process is accelerated for all structures and all values of N (average acceleration by a factor of 2.0 and maximum of 2.21). Figure 2 shows the number of iterations for structures 2 and 3 while solving the k^{th} linear system in the cases when preconditioners are obtained from the first (\mathbf{M}_1) and the 50th (\mathbf{M}_{50}) matrices. It is evident that the total number of iterations for the case when the preconditioner is computed from the 50th matrix is less than the one from the 1st matrix. This difference explains the obtained acceleration. One can see that the

acceleration is independent from the variant of order of operations and is constant for all structures and orders of the linear system matrix (small differences can be explained by a computational error of time measurement).

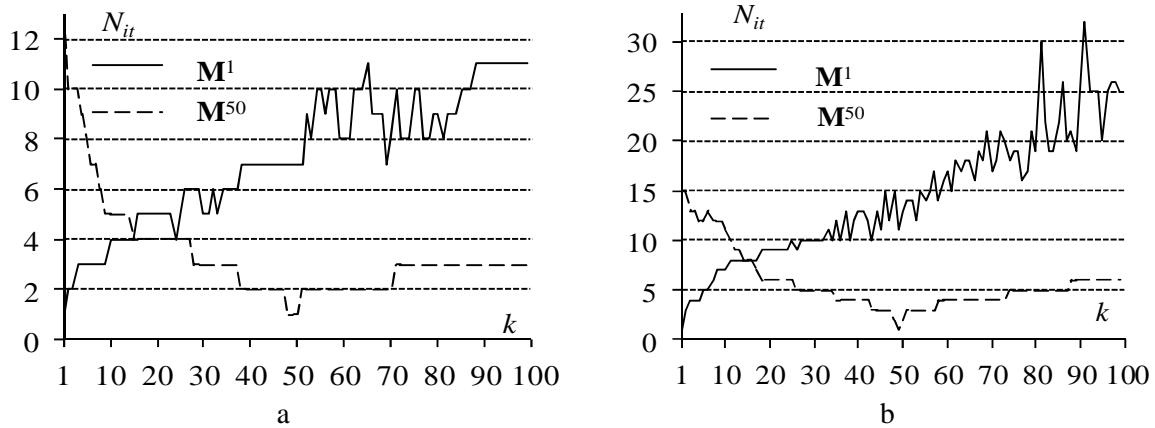


Figure 2. Number of iterations depending on k , with preconditioner obtained from the first (\mathbf{M}_1) and the 50th (\mathbf{M}_{50}) matrices of linear systems for structures: 1 – $N = 3001$ (a); 2 – $N = 3001$ (b).

Then the authors carried out a computational experiment to choose the f -th matrix (\mathbf{M}_f) for computing the preconditioner used in solving the whole sequence of linear systems. The investigation started with structure 2 that has the worst matrix conditioning. Figure 3 shows the obtained accelerations compared to the case of the 1st matrix. It is clear that acceleration is almost constant in the range of f from 30 to 60. The fastest acceleration is 2.21 for $f = 50$ and $N = 3001$. For structure 4 and $N = 1901$, the fastest acceleration of 1.95 is obtained for $f = 67$. Consequently, let us make a conclusion that the 1st matrix is the worst option for a preconditioner, as all other matrices provide acceleration, for example, by a factor of 2 for the middle of the range.

Figure 4 shows dependencies of per-unit-length capacitance derived by means of Gaussian method (C^{GE}) and capacitance error derived by means of the BiCGStab (C^t) method on t for structure 1 for $N = 1600$. One can see that the maximum error is less than a thousandth of a per cent. Thus, computations are quite accurate.

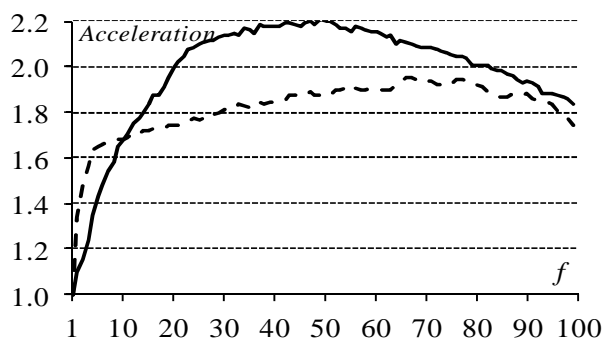


Figure 3. Acceleration of solving 100 linear systems obtained by the BiCGStab method with the \mathbf{M}_f matrix for computing a preconditioner (with respect to the 1st matrix) for structure 2 for $N = 3001$ (—) and structure 4 for $N = 1901$ (---).

4. Conclusion

Using the analysis of 4 strip structures, the authors have proved that additional acceleration of iterative process can be obtained when solving linear systems multiple times by choosing a proper order of operations and computing the optimal preconditioner. The first matrix of a sequence is ineffective for computing a preconditioner while a simple choice from the middle of the range (the 50th matrix) can give acceleration by a factor of 2. Various orders of operations in computing a preconditioner from the middle of the range insignificantly influence total acceleration. So, there is no need either to

recompute the preconditioner or to update it.

Thus the obtained results can be useful for accelerating computer-aided design of various strip structures. Similar investigations for logarithmic and random variations of a parameter and simultaneous variations of several parameters in analysing more complex elements of the equipment should be carried out in the future. It is worth noting that the proposed idea for acceleration is very simple and easy to implement in software. It does not require additional computer memory or user-defined parameters for computations. Moreover, it could be used for any iterative method with preconditioning. Therefore, choosing the order of operations to accelerate the process of solving systems is quite universal and could be used not only for strip structure analysis but also for a wide range of computational problems.

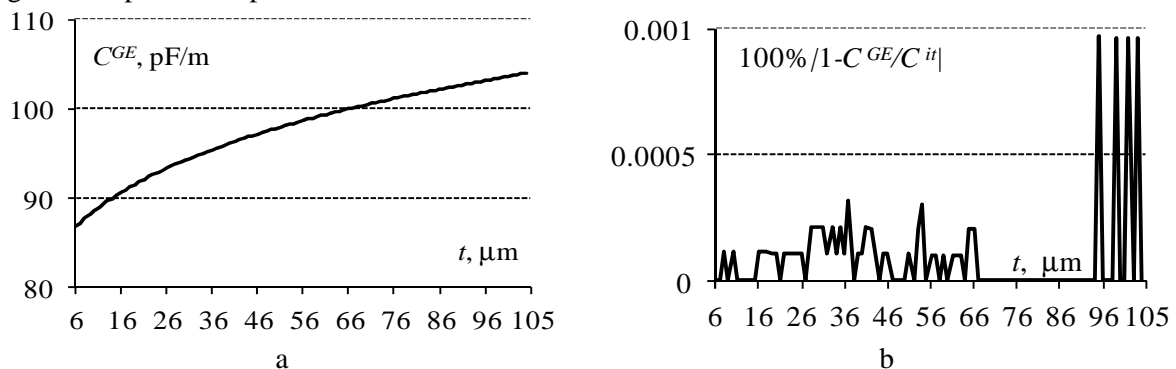


Figure 4. Dependencies of per-unit-length capacitance (a) and error of BiCGStab method C^{it} with respect to Gaussian method C^{GE} (b) on t for structure 1 for $N = 1600$.

5. Acknowledgments

This research was supported by the Ministry of Education and Science of the Russian Federation (RFMEFI57417X0172).

References

- [1] Wei C, Harrington R F, Mautz J R, and Sarkar T K 1984 *IEEE Trans. Microw. Theory Tech.* **32(4)** 439–450
- [2] Tebbens J D, Tuma M 2010 *Numer. Linear Algebra Appl.* **17(6)** 997–1019
- [3] Bertaccini D. *Elect. Trans. Numer. Math.* 2004 **18** 49–64
- [4] Bellavia S, Morini B, Porcelli M *Opt. Meth. and Soft.* 2014 **29(2)** 321–340
- [5] Birken P, Tebbens J D, Meister A, Tuma M 2008 *Appl. Numer. Math.* **58(11)** 1628–1641
- [6] Parks M L, De Sturler E, Mackey G, Johnson D D, Maiti S 2006 **28(5)** *SIAM J. Sci. Comp.* 1651–1674
- [7] Simoncini V, Gallopoulos E 1996 *Comp. and Appl. Math.* **66(1–2)** 457–469
- [8] Puzyrev V, Cela J M 2015 *Geophys. J. Int.* **202** 1241–1252
- [9] Devi A, Gandhi M, Varghese K, Gope D 2013 *IEEE Electrical Design of Advanced Packaging and Systems Symposium* 100–103
- [10] Edelman A 1993 *Int. J. of High Perf. Comp. Appl.* **7(2)** 113–128
- [11] Harrington R 1990 *IEEE Ant. and Prop. Mag.* **32(3)** 31–35
- [12] Livesey M, Stack J F, Costen F, Nanri T, Nakashima N, Fujino S 2012 *IEEE Ant. and Prop. Mag.* **54** 186–195
- [13] Akhunov R R, Gazizov T R, Kuksenko S P 2016 *Computational Mathematics and Mathematical Physics* 56(8) 1382–1387
- [14] Gazizov T R, Kuksenko S P, Akhunov R R 2015 *Int. J. Math. Models and Meth. in Appl. Sci.* **9** 721–726.
- [15] Zabolotsky A M 2015 *Dokladi TUSUR* **2(36)** 41–44 (in Russian)
- [16] Zabolotsky A M, Gazizov A T 2015 *Int. J. of Circ. Syst. and Sig. Proc.* **9** 68–74.