New Developments for Improved Simulation of Interconnects Based on Method of Moments

Sergei Kuksenko, Talgat Gazizov, Alexander Zabolotsky, Roman Ahunov, Roman Surovtsev, Vasiliy Salov and Egor Lezhnin

634034 Lenin Ave., 40, Tomsk State University of Control Systems and Radioelectronics
Tomsk, Russia

***Abstract*—The paper gives arguments for the topicality of electromagnetic compatibility modeling. Trends of electromagnetic simulation software development are stated. General simulation process based on method of moments is described. On examples of interconnects the improvement of simulation is shown, obtained by means of decrease of computational costs on segmentation and solution of linear algebraic systems. Time response calculation is described and results of waveform animation are presented. Recently designed testing tools are briefly described.**

Keywords-electromagnetic compatibility; interconnects; method of moments; adaptive iterative choice of the optimal segmentation; linear systems; software

I. INTRODUCTION[1]

Electronics is widely used in all areas of the infrastructure of modern society. Increase of electronics working in confined space, leads to the increase of its density. The steady increase in efficiency of electronics is ensured mostly by the growth of the upper frequency of its signal frequency band. Now, these trends more frequently lead to malfunctions of electronics due to electromagnetic interference, thus, it is necessary to ensure electromagnetic compatibility (EMC). For this aim lengthy and expensive tests are conducted. Elimination of defects breaks the working process and leads to additional costs. Costs and delays that would be necessary in the case of ignoring the EMC issues can be minimized by means of early and regular accounting of the EMC in the product design. Particularly, signal and power integrity problem due to distortions of a signal propagating along electrical intecconnects is one of the important EMC problems [1]. Therefore, it is necessary to solve this problem at the stage of electronics design; it can be done through simulation via specialized software. In recent years there are two trends in the development of such software.

The first trend consists in expanding the functionality of the software by improving primary (that historically appeared first, and on the basis of which the software was created) numerical method, the integration of other numerical methods and their hybrids, as well as integration of the segmentation, methods of linear systems solving, and import tools. This approach allows expanding the capabilities of the software mostly by enhancing its versatility. If the main method is not suitable for the desired type of analysis, a user can always use another method or hybrid. For example, there are such software as FEKO and EMC Studio. Initially they used the method of moments, and now allow the use of finite element and/or finite differences method.

The second trend is the takeover of some software by the others. If we speak about expanding of the functionality, this approach has a lot in common with the first one, but it is more versatile. For example, the CST company has taken over the Flomerics company, which has been developing products for the electromagnetic simulation, such as MicroStripes and FLO/EMC. After this takeover, CST expanded its variety of computing modules by means of well-established developments of Flomerics, based on the method of transmission line matrix. We saw the similar situation when ANSYS has taken over ANSOFT. Besides, nowadays companies are working toward improvement of the user interface, implementation of computing using graphics processors, and adding of new modules (for example, CST EMC Studio).

Now the Finite Difference Time Domain method and Method of Moments (MoM) are most often used to solve electromagnetic problems. Various software is developed for this aim, proposing new features for designers. However, complexity of electronics dictates new software to be developed. This paper considers use of MoM in TALGAT software on example of interconnects simulation.

II. BRIVE OVERVIEW OF TALGAT SOFTWARE

Clients of TALGAT software are the interface modules. The core of the software is TLCORE. Computational modules: MOM2D – capacitance matrix computation for two-dimensional structures; MOM3D – capacitance matrix computation for three-dimensional structures; RESPONSE – quasi-static analysis of multiconductor transmission lines response; MOMW – electromagnetic analysis of wire structures. Optimization modules: ES – evolution strategy; GA – genetic algorithms. Utility modules – assistant modules (matrix processes, operations with diagrams etc.). An interpreted script language TALGAT_Script, using postfix notation of expressions, provides "dialogue" with the software.

III. METHOD OF MOMENTS FOR CAPACITANCE MATRIX COMPUTATION

Effectiveness of MoM application to obtain results of desirable accuracy is defined, finally, by computational costs. Not considering the synthesis problem, in analysis problem for one set of parameters, we have MoM solution consisting of the following steps: derivation of integral equations for a considered structure from Maxwell equations; discretization of structure (segmentation on N subdomains, in each of which the sought-for function is approximated by basis functions); calculation of entries of square matrix of order N for linear algebraic system; calculation of entries of excitation vector **v** of size N; solution of the linear algebraic system; calculation of desirable characteristics (for example, capacitance matrix) from the solution.

Order of discretization is the following [2]. First we are segmenting the conductor-dielectric boundaries (subdividing them into intervals further called subintervals) and assign numbers from 1 to N_C to these subintervals. (The infinite ground plane is not segmented. If there is a second infinite plane, it is bounded at a point far removed from the conductors, segmented as a conventional conductor, and considered to be at zero potential. If there are other conductors that are always at zero potential, they all are conventionally segmented.) Then we are segmenting the dielectric-dielectric boundaries and assign to obtained N_D subintervals numbers from N_C+1 to N, so that N=N_C+N_D.

For clarity, consider the calculation for two-dimensional problem (formulas for three-dimensional problem [3] are similar). Used notations are ε_n – the permittivity of the nth conductor-dielectric subinterval; ε_n^+ и ε_n^- – (further used only with m index) the permittivities on the positive (to which the vector $\boldsymbol{n}_n$ is pointed) and negative sides (from which the vector $\boldsymbol{n}_n$ is emanated) of the nth dielectric-dielectric subinterval, where n_n is vector with unit length, pointing out orthogonally from center of nth subinterval. Generally, $\underline{r}_n$ vector is according to center of nth subinterval. Particularly, $\underline{r}'_n$ vector is according to subinterval of integration, while the $\underline{\hat{r}}'_n$ vector is according to the image of the subinterval relatively to infinite ground plane. For the matrix **S** rows with numbers m=1, …, N_c, according to conductor-dielectric subintervals we have for s_{mn} entries

$$s_{mn}=\frac{1}{2\pi\varepsilon_0}\left(\text{iflg}\cdot\hat{\text{I}}_{mn}-\text{I}_{mn}\right),\begin{cases}m=1,\ldots,N_\text{c}\\ n=1,\ldots,N\end{cases},$$

where iflg=1 if infinite ground plane is present, else iflg=0;

$$\text{I}_{mn}=\int_{\lambda_n}\ln\left|\underline{r}_m-\underline{r}'_n\right|d\lambda'\ ;\ \hat{\text{I}}_{mn}=\int_{\lambda_n}\ln\left|\underline{r}_m-\underline{\hat{r}}'_n\right|d\lambda'\ .$$

For the matrix rows with numbers m=(N_c + 1), …, N, according to dielectric-dielectric subintervals we have for s_{mn} entries

$$s_{mn}=\frac{1}{2\pi\varepsilon_0}\left(\text{I}_{mn}-\text{iflg}\cdot\hat{\text{I}}_{mn}\right),\begin{cases}m=(N_c+1),\ldots,N\\ n=1,\ldots,N\end{cases},m\neq n;$$

$$s_{mm}=\frac{1}{2\pi\varepsilon_0}\left(I_{mm}-\text{iflg}\cdot\hat{I}_{mm}\right)+\frac{1}{2\varepsilon_0}\frac{\varepsilon_m^++\varepsilon_m^-}{\varepsilon_m^+-\varepsilon_m^-},m=(N_c+1),\ldots,N,$$

where

$$I_{mn}=\int_{\lambda_n}\frac{\underline{r}_m-\underline{r}'_n}{\left|\underline{r}_m-\underline{r}'_n\right|^2}\underline{n}_n d\lambda'\ ,\ \hat{I}_{mn}=\int_{\lambda_n}\frac{\underline{r}_m-\underline{\hat{r}}'_n}{\left|\underline{r}_m-\underline{\hat{r}}'_n\right|^2}\underline{n}_n d\lambda'\ .$$

As a result, we obtain the matrix **S** of structure shown in Figure I.

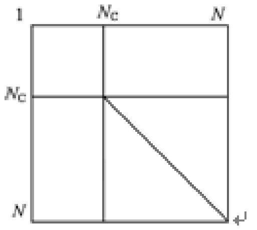

FIGURE I. STRUCTURE OF MATRIX S

Thus, the problem of capacitive matrix calculation for arbitrary system of conductors and dielectrics is reduced to linear algebraic equation **S**$\boldsymbol{\sigma}$=**v**, with square and dense matrix of order N, relating charge densities (composing vector $\boldsymbol{\sigma}$) on discretization subdomains of boundaries of conductors and dielectrics, with assigned potentials of these subdomains (composing vector **v**). This equation is solved N_{COND} times (N_{COND} is number of conductors without the reference), where for ith solution, the ith conductor potential v_i, i=1, …, N_{COND}, is 1 V, while all other entries of vector **v** are 0 V.

IV. SEGMENTATION OF BOUNDARIES

One of effective ways to decrease computational costs is proper segmentation. It makes sense to start with the rough segmentation, as following computation requires minimal costs, and rather acceptable results can be obtained even with the rough segmentation. The need for more accurate results naturally leads to the iterative improvement of segmentation, i.e. the repeat of calculations with improved segmentation until solutions with a given accuracy is obtained. It is natural to use the results of the previous iteration, for example, the calculated charge densities at the boundaries – for the nonhomogeneous segmentation (more frequent in places with a big changes in the density). For this purpose, the software provides an adaptive iterative choice of the optimal segmentation (AICOS) for boundaries of conductors and dielectrics, it allows to obtain the desired response with controlled accuracy, and saves time and memory [4]. Simple version of AICOS has the following steps:

AICOS algorithm

```
1   Assign parameters of structure and accuracy of calculation ($tol$).
2   Set $i$=0 and the segment length equal to width of conductor $w$.
3   Calculate value ($K_i$) of characteristic under investigation.
4   Set $i$=$i$+1 and the half the segment length.
5   Calculate value ($K_i$) of characteristic under investigation.
6   If $(K_i – K_{i–1})/K_{i–1} > tol$, go to step 4.
```

To demonstrate the AICOS effectiveness the computational experiment has been conducted for two-dimensional structure of cross section shown in Figure II. All sizes are taken from real printed circuit board: conductor width w = 890 μm; separations s_1 = 500 μm, s_2 = 1890 μm; thickness of all conductors t = 35 μm, thickness of prepreg layers h_1 = h_3 = 144 μm, substrate thickness h_2 = 220 μm. Thickness of solder mask h_M = 30 μm. Width of side grounded conductors is 5w. To decrease random uncertainty the time of 100 computations was measured.

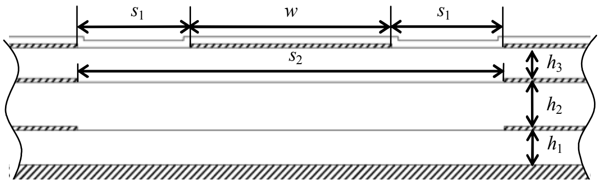

FIGURE II. FRAGMENT OF CROSS SECTION OF CONSIDERED STRUCTURE

For each iteration number (i) and length of segment (l_i) the order (N_i) of solved linear algebraic system, the time of calculation (T_i), the value of calculated capacitance per unit of length (C_i), are given in Table I. The number of required iterations for tol = 0.01 is 4. For equidistant segmentation of 10 μm, defined by user (taken from condition of subdividing an edge of conductor by 3 segments) the computational time is 1798.02 s, N = 7135, C = 129.7 pF/m. The total time of iterations (the sum of T_i in rows 1–4 of Table I) is 13.47 s, that is by 133 times less than time for the user defined segmentation.

TABLE I. RESULS OF COMPUTATIONAL EXPERIMENT FOR C, $C0$ AND Z

i	l_i, μm	N_i for C	T_i, s for C	C_i, pF/m	N_i for $C0$	T_i, s for $C0$	$C0_i$, pF/m	T_i, s for Z	Z_i, Ω
1	890	105	0.47	121.4	76	0.46	38.09	1.054	48.78
2	445	182	0.96	124.4	138	0.93	39.12	2.286	47.66
3	222.5	340	2.70	125.7	262	2.50	39.62	6.302	47.15
4	111.25	659	9.34	125.8	510	8.19	39.85	21.962	46.94

Similar results for capacitance per unit of length, calculated without dielectrics ($C0_i$), are also given in Table I. For similar user defined segmentation the computational time is 1309.12 s, N = 5560, $C0$ = 40.11 pF/m. The total time is 12.08 s that is by 108 times less than time for the user defined segmentation.

At last, similar results for wave impedance (Z_i) are also given in Table I. For similar user defined segmentation the computational time is 3127.25 s, while the total time is 30,604 s that is by 142 times less.

Thus, use of AICOS, even of simple algorithm option, can reduce considerably and automatically (without a user) the computational costs for calculation of transmission line parameters. Therefore, usage and further improvement of AICOS are perspective.

V. TAKING ACCOUNT OF FREQUENCY DEPENDENCE OF MATERIAL PERMITTIVITY

To demonstrate importance of taking account of frequency dependence of material permittivity we consider again the simple structure of coupled microstrip line. (Case of two conductors permits to estimate effect of considered factors on even and odd modes of the line signal separately.) For dielectric layer we take commonly used FR-4, which permittivity dependence on frequency $\varepsilon_r(f)$ is well known [5] and implemented in TALGAT. When simulating with ε_r and tgδ being independent on frequency their values for 1 GHz are taken. As excitation the trapezoidal pulse with rise and fall times of 100 ps and flat top time of 10 ps is considered. Length of the line is 1 m. To mitigate reflections all ends of the line are terminated by resistance of geometric average of even and odd mode impedances. Four options are simulated (Figure III): ε_r does not depend on frequency (ε_r= 4.54), lossless dielectric (tgδ= 0); ε_r depends on frequency, tgδ = 0; ε_r does not depend on frequency (ε_r= 4.54), but dielectric losses are taken into account as frequency independent (tgδ = 0.0153); 4) ε_r and tgδ depend on frequency.

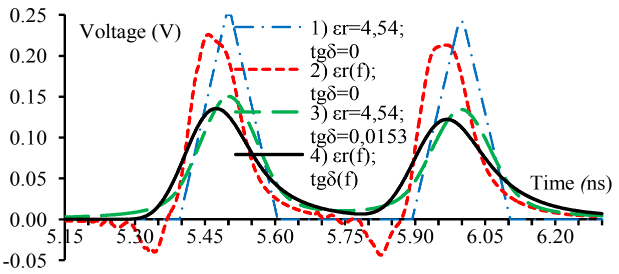

FIGURE III. WAVEFORMS AT EXCITED CONDUCTOR END OF COUPLED MICROSTRIP LINE FOR 4 OPTIONS

Option 1 shows that input pulse is decomposed on two pulses (of even and odd modes) of close magnitudes. In option 2 the noncausality appeared in form of negative undershoots before arrival of main part of both pulses. Option 3 shows that magnitudes of first and second pulses are reduced considerably due to losses of energy in dielectric. Previous noncausality of option 2 disappeared, but early and slow rise of pulses is observed that is also noncausal. Note also that the noncausal rise of second pulse increased fall time of first pulse. At last, option 4 shows most real response. Thus, simulation with complete taking account of frequency dependence of complex permittivity permits to obtain more correct results, than the simulation without such accounting.

However, comparison of options 3, 4 shows that changing the values of ε_r and tgδ in option 3 we can obtain maximum coincidence of waveforms of options 3, 4. The obtained values can give acceptable accuracy of simulation in a range of parameters or during optimization. Thus, it is possible to use option 3 with single calculation of capacitive matrix and to avoid use of option 4 that is time-consuming due to need of multiple calculation of capacitive matrix for each frequency point of signal spectrum. Additional observation shows that despite of change of even and odd mode delays their difference is not changed. This fact again permits to use option 3 instead of option 4, at least for preliminary simulation.

VI. Multiple compmuting of the capacitance matrix

To calculate the capacitive matrix ($\underline{\mathbf{C}}$) of a structure consisting of conductors and dielectrics, it is necessary to solve a linear system of $\mathbf{S}\boldsymbol{\sigma}=\mathbf{v}$ with a square and dense matrix $\mathbf{S}$ of order N. The number of subdomains on the boundaries of conductor-dielectric (N_C) and a dielectric-dielectric (N_D) form the order of a matrix, and the elements of the matrix are calculated from the parameters of these subdomains. Vector $\mathbf{v}$ consists of the assigned potentials of these subdomains, and the desired vector $\boldsymbol{\sigma}$ gives the distribution of the charge density on them. In practice, to obtain accurate and causal results we should take into account the frequency dependence of relative dielectric permittivity ε_r of dielectrics. Then linear systems solution is performed for each frequency point of the band, which increases the total computational time, proportionally to the number of frequency points. However, when ε_r is changed (if other parameters of the original structure are constant), we can see changes only in elements with the index greater than N_C, situated on the main diagonal of the linear system matrix, corresponding to dielectric-dielectric subdomains. This resource was implemented in the software to reduce the total computational time [6].

The original algorithm for computing M capacitance matrices $\underline{\mathbf{C}}$ in TALGAT software as follows.

Algorithm 1

```
Set $\underline{\mathbf{C}}$=0
For $k$ from 1 to $M$
   Compute the entries of the matrix $\mathbf{S}_k$
   Perform the LU factorization of the matrix $\mathbf{S}_k$
   For $i$ from 1 to $N_{COND}$
      Compute the elements of the excitation vector $\mathbf{v}_i$
      Find the solution vector $\boldsymbol{\sigma}_{ik}$ from the equation $\mathbf{S}_k\boldsymbol{\sigma}_{ik}=\mathbf{v}_i$
      Compute the entries of the ith column of the matrix $\underline{\mathbf{C}}$ based on $\boldsymbol{\sigma}_{ik}$
   Set $i=i+1$
Set $k=k+1$
```

In order to evaluate the arithmetic complexity of solving a linear system by the improved algorithm for computing the capacitance matrix [6], consider this algorithm in more details.

Given a certain ε_r, represent the corresponding coefficient matrix S and the related matrices $\mathbf{L}$ and $\mathbf{U}$ in block form as follows:

$$\mathbf{S}=\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\Rightarrow\mathbf{L}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{C} & \mathbf{I}\end{bmatrix},\mathbf{U}=\begin{bmatrix}\mathbf{A} & \mathbf{A}^{-1}\mathbf{B}\\ \mathbf{0} & \mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B}\end{bmatrix},$$

where $\mathbf{I}$ is the identity matrix. This block representation differs from the conventional one but provides for the maximum acceleration when implemented. As ε_r changes, only the diagonal entries of the block $\mathbf{D}$ vary. Represent the block $\mathbf{D}$ as the sum of a matrix $\underline{\mathbf{D}}$ with zero diagonal and a diagonal matrix $\mathbf{Diag}$, which contains the diagonal entries of $\mathbf{D}$, and take into account that we need to invert the matrix $\mathbf{A}$, then:

$$\mathbf{S}=\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \underline{\mathbf{D}}\end{bmatrix}+\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{Diag}\end{bmatrix}\Rightarrow\mathbf{L}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{C} & \mathbf{I}\end{bmatrix},$$

$$\mathbf{U}=\begin{bmatrix}\mathbf{A}^{-1} & \mathbf{A}^{-1}\mathbf{B}\\ \mathbf{0} & \underline{\mathbf{D}}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B}+\mathbf{Diag}\end{bmatrix}.$$

From the above Algorithm 1 it is seen that formally the inner loop over i can be avoided if the vectors $\boldsymbol{\sigma}_i$ and $\mathbf{v}_i$ (i=1, 2, ..., N_{COND}) are replaced by the corresponding $N\times N_{COND}$ matrices $\sum$ and $\mathbf{V}$, and by using an auxiliary $N\times N_{COND}$ matrix $\mathbf{X}$ (these matrices are represented in block form; the blocks $\sum_0$, $\mathbf{V}_0$, and $\mathbf{X}_0$ are $N_C\times N_{COND}$ matrices, whereas the blocks $\sum_1$, $\mathbf{V}_1$, and $\mathbf{X}_1$ are $N_D\times N_{COND}$ matrices). Now the improved algorithm for computing the capacitance matrix can be described as follows.

Algorithm 2

```
1   Set $\underline{\mathbf{C}}$=0
2   For $k$ from 1 to $M$
3      If $k$=1
4         then compute the entries of the matrix $\mathbf{S}_1$, and write the diagonal
          entries of the block $\mathbf{D}$ to the matrix $\mathbf{Diag}_1$
5         $\mathbf{A}_1 = \mathbf{A}_1^{-1}$
6         $\mathbf{B}_1 = \mathbf{A}_1\mathbf{B}_1$
7         $\mathbf{D}_1 = \underline{\mathbf{D}} - \mathbf{C}_1\mathbf{B}_1$
8         Compute the entries of the excitation matrix $\mathbf{V}$
9         $\mathbf{X}_0 = \mathbf{A}_k\mathbf{V}_0$ ($\mathbf{X}_0$ and $\mathbf{V}_0$ are $N_C\times N_{COND}$ matrices)
10        $\mathbf{X}_1 = \mathbf{V}_1 - \mathbf{C}_k\mathbf{X}_0$ ($\mathbf{X}_1$ and $\mathbf{V}_1$ are $N_D\times N_{COND}$ matrices)
11     Else
12        $\mathbf{S}_k = \mathbf{S}_1$
13     $\mathbf{D}_k = \mathbf{D}_k + \mathbf{Diag}_k$
14     $\boldsymbol{\sigma}_{1k} = \mathbf{D}_k^{-1}\mathbf{X}_1$
15     $\boldsymbol{\sigma}_{0k} = \mathbf{X}_0 - \mathbf{B}_k\boldsymbol{\sigma}_{1k}$
16     Compute the entries of the capacitance matrix $\underline{\mathbf{C}}$
17  Compute the entries of the matrix $\mathbf{Diag}_{k+1}$
18  Set $k=k+1$
```

Table II presents the expressions from Algorithm 2 that are used in solving linear systems and their complexities (without taking into account writing matrices and vectors to the memory), which are computed by the above formulas (with account for the relation $N=N_C+N_D$).

TABLE II. Arithmetic complexity of solving a linear system by Algorithm 2

Number of step	Expression	Arithmetic complexity (Q)	Number of computations
5	$\mathbf{A} = \mathbf{A}^{-1}$	$Q_5=\frac{8}{3}N_C^3-\frac{3}{2}N_C^2-\frac{1}{6}N_C$	1
6	$\mathbf{B} = \mathbf{AB}$	$Q_6=N_DN_C\left(2N_C-1\right)$	1
7	$\mathbf{D} = \underline{\mathbf{D}} - \mathbf{CB}$	$Q_7=2N_CN_D^2$	1
9	$\mathbf{X}_0 = \mathbf{AV}_0$	$Q_9=N_CN_{COND}\left(2N_C-1\right)$	1
10	$\mathbf{X}_1 = \mathbf{V}_1 - \mathbf{CX}_0$	$Q_{10}=2N_CN_DN_{COND}$	1
14	$\boldsymbol{\sigma}_1 = \mathbf{D}^{-1}\mathbf{X}_1$	$Q_{14}=\frac{8}{3}N_D^3+N_D^2\left(2N_{COND}-\frac{3}{2}\right)+$ $N_D\left(N_{COND}-\frac{1}{6}\right)$	M
15	$\boldsymbol{\sigma}_0=\mathbf{X}_0 - \mathbf{B}\boldsymbol{\sigma}_1$	$Q_{15}=2N_CN_DN_{COND}$	M

The ultimate analytic expression for the acceleration of solving M linear algebraic systems is the ratio (β_Q) of the number of operations of Algorithm 1 to the number of operations of Algorithm 2:

$$\beta_Q = \frac{Q_{LUSolve}}{Q_{BLUSolve}} = \frac{MQ_{LU}}{\left(Q_5 + Q_6 + Q_7 + Q_9 + Q_{10}\right) + M\left(Q_{14} + Q_{15}\right)} = \frac{MQ_{LU}}{Q_F^* + MQ_M^*},$$

here

$$Q_{LU} = \frac{\left(N_C + N_D\right)}{6}\left\{\begin{matrix} 4\left(N_C + N_D\right)^2 - 3\left(N_C + N_D\right) + \\ 6N_{COND}\left[2\left(N_C + N_D\right) - 1\right] - 1 \end{matrix}\right\}$$

is the number of operations of Algorithm 1;

$$Q_F^* = \frac{N_C}{6}\left\{\begin{matrix} 12\left(N_C + N_D\right)^2 - 6\left(N_C + N_D\right) + 4N_C\left(N_C - 3N_D\right) - \\ 3N_C + 6N_{COND}\left[2\left(N_C + N_D\right) - 1\right] - 1 \end{matrix}\right\}$$

is the number of operations for computing the matrix decomposition and the auxiliary matrices when computing the matrix $\underline{\mathbf{C}}$ for the first time;

$$Q_M^* = \frac{N_D}{6}\left\{16N_D^2 - 9N_D + 6N_{COND}\left[2\left(N_C + N_D\right) - 1\right] - 1\right\}$$

is the number of operations for solving linear systems starting from the second one.

Numerical values of the acceleration for N=1000 computed for N_C=900 and M=1, 100, 500, 1000, are presented in Table III.

TABLE III. ACCELERATION PROVIDED BY THE IMPROVED ALGORIHM IN THE CASE OF COMPUTING M CAPACITANCE MATRICES FOR N=1000, N_C=900

N_{COND}	M			
	1	100	500	1000
1	0.31	27.73	94.1	134.27
10	0.32	26.33	76.80	101.0

Obviously, if the analysis is required when changing the geometrical parameters of the structure, the matrix entries will vary in arbitrary places, and therefore the above-mentioned approach is not applicable. To overcome this problem a use of iterative methods was proposed.

The algorithm for multiple iterative solution of the linear system with partially changing matrix was presented in [7]. In this algorithm, the preconditioner matrix M is formed from the first linear system. Further, this matrix (without recalculation) is used for solving the following linear systems, thereby reducing the total solution time with acceptable accuracy. Finally, it was supposed that a similar algorithm can be applied when changing the sizes of the structure being analyzed. As a first step in this direction the reduction of the residual norm was investigated for solving the 10 linear systems, obtained by small changes for several parameters of a structure [8]. Example of these calculations for dielectric height (h) of microstrip line is presented in Figure IV.

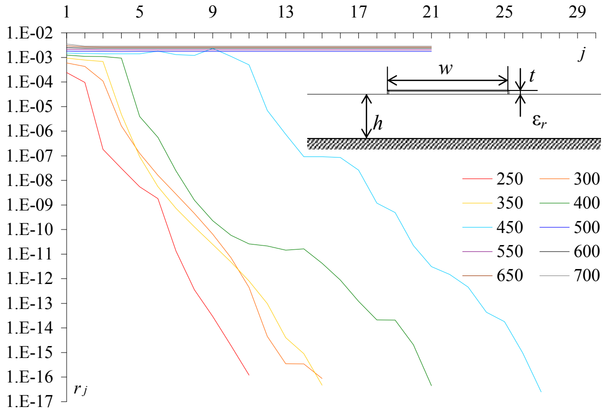

FIGURE IV. DEPENDENCE OF RELATIVE RESIDUAL NORM ON THE NUMBER OF ITERATIONS FOR H=250, 300, ..., 700 μM WITH CROSS SECTION OF CONSIDERED STRUCTURE

It was shown that the use of compressed sparse row format (CSR) for storing a preconditioner matrix is effective to reduce the computational cost. Acceleration of iterative solution of linear systems with dense matrices using sparse matrix storage formats has been considered in details [9]. Formulas for comparing the sparse matrix storage formats have been derived. An iterative algorithm for solving linear algebraic systems using sparse row format for storing prefiltered preconditioners has been designed. A modification of the sparse row format leading to 1.14–1.23-times speed-up for matrices of order 1000 has been suggested. It has been demonstrated that as opposed to the usual storage format, the sparse row format provides for 1.5–1.6-times speed-up in solving the linear systems of orders 4800, 6000, and 8000. The use of the obtained results allows one to reduce both memory and time requirements in solving large-scale problems with dense matrices.

Then, improvements to the ILU(0) factorization algorithm for preconditioning linear algebraic systems with dense matrices have been suggested [10]. (The preconditioner is stored in compressed sparse row format.) On the example of the problem of computing the electrical capacity of two stripes, it has been demonstrated that the modifications proposed provide for a significant reduction of the time for computing the ILU(0) preconditioner (up to 4 times) and for solving the preconditioned linear system (up to 2.5 times). On real PCB structure problems, a new investigation has been performed in order to reveal the optimal value of the main parameter (drop tolerance) of the iterative solution of linear systems. The algorithm for calculation of capacitance matrices has been improved for case of multiple calculations. The improved algorithm works up to 4 times faster than the initial one.

The possibility of multiple iterative solution of linear systems was further investigated for computing the capacity of microstrip line in the wide ranges of its sizes. To accelerate the iterative process two ways were considered. The first one is a use of a previous linear system solution (vector $\mathbf{x}_{i-1}$) as an initial guess for a following solution (vector $\mathbf{x}^0{}_i$), i.e. $\mathbf{x}^0{}_i$=$\mathbf{x}_{i-1}$ (for the first system a unit vector are used). The second one is the use of preconditioning matrix M, obtained by solving the first linear algebraic equation, i.e. $\mathbf{M}_i$=$\mathbf{M}_0$. In computational experiments four variants were used: in variant 1 acceleration was not used. In variants 2 and 3, these ways were used separately, and in variant 4 these ways were used together.

The previous structure (Figure IV) was investigated. The aim of the experiment was to evaluate the time expenses required for the calculation of 100 capacitive matrices obtained by changing one of the dimensions of the structure: dielectric height h (in the range of 12–112 μm or 933%); conductor width w (in the range 18–118 μm, or 656%); conductor height t (in the range of 6–106 μm, or 1767%). The number of segments on each boundary of structure has not changed, which allows for constant order N=1600 of the linear system matrices for correct comparison. As iterative method the BiCGStab method was chosen. Iterations were continued until the relative norm of the residual vector was more than 10^{-8}. Gaussian elimination was used for comparison. Ratios of the total (for 100 linear systems) solution times by Gauss elimination and by iterative method are shown in Table IV for all variants. Calculations have demonstrated the effectiveness of the proposed acceleration ways. The number of iterations when using the variant 4 is minimal, that reduces the total time of 100 linear systems solution by factor about 5–12 and proves the effectiveness of the combined usage of acceleration ways.

TABLE IV. ACCELERATION OF SOLVING 100 LINEAR SYSTEMS

Varied parameter	Variant 1	Variant 2	Variant 3	Variant 4
h	0.48	1.32	6.49	11.77
w	0.31	1.15	5.87	10.98
t	0.37	1.28	2.87	4.92

In the cases where h and w are changed, the number of iterations is less than in the case where t changes. This effect can be explained by considerably larger changes in the matrix entries. In order to evaluate these changes, the ratios of the matrix norms $\|\Delta\mathbf{A}_i\|_1/\|\Delta\mathbf{A}_1\|_1$ and $\|\Delta A_i\|_\infty/\|\Delta A_1\|_\infty$ as functions of the number of linear systems being solved were computed. The corresponding results, presented in Figure V, show that the changes in the matrix norms resulting from changing t are greater than those resulting from changing h and w.

Efficiency of the method considered decreases as the difference between the initial and current matrices grows. This results in that the required number of iterations increases, and the iterative solution slows down. In order to alleviate this inefficiency, it has been suggested to recompute the preconditioner **M** whenever the rate of convergence in solving the current linear system is too slow.

We suggest to change the preconditioning matrix if the current iteration number N_{it} exceeds a prescribed threshold value N_{it}^{MAX}. The corresponding algorithm for solving M linear algebraic systems, in which the matrix **M** is changed when N_{it}^{MAX} is exceeded, can be written (Algorithm 3).

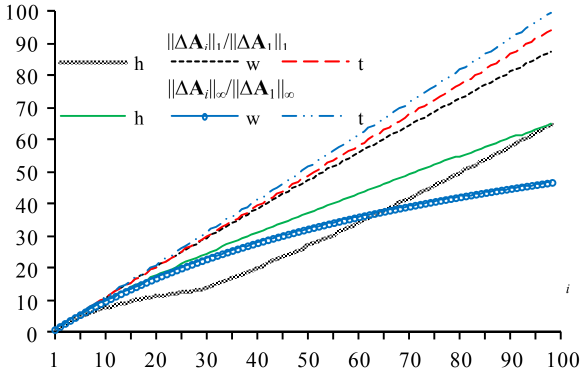

FIGURE V. RATIOS OF THE MATRIX NORMS AS FUNCTIONS OF THE NUMBER OF LINEAR ALGEBRAIC SYSTEMS FOR VARYING H, W, AND T

Algorithm 3

```
1  Compute M as the ILU(0) factorization of $\mathbf{A}_1$
2  $N_{it} = 0$
3  For $i = 1, \ldots, M$
4    if $N_{it} > N_{it}^{MAX}$ and $i > 1$, then
5      compute M as the ILU(0) factorization of $\mathbf{A}_i$
6    Find $\mathbf{x}_i$ by solving $\mathbf{MA}_i\mathbf{x}_i = \mathbf{Mb}$ with a prescribed accuracy
7  $i = i + 1$
```

In the above algorithm, the classical ILU(0) factorization is used, and no prefiltering is carried out because, for a large number of linear systems, this algorithm is efficient only if the zero drop tolerance value is used [10]. The initial preconditioner M is computed at Step 1. If the current iteration number exceeds N_{it}^{MAX}, then the matrix M is recomputed (Step 5) and used in solving the subsequent linear systems. Consider the influence of the threshold parameter N_{it}^{MAX} on the solution process. If it is too small, then the preconditioner will be recomputed too many times, which will result in an unacceptable solution time. In the case where N_{it}^{MAX} is too large, the preconditioner will be recomputed not many times but the total costs will be large because too many iterations will be performed for solving poorly preconditioned systems. These arguments suggest that one should look for an optimal value of the threshold parameter, for which the total solution time will be minimized. This problem was investigated numerically. In Table V, detailed data characterizing the solution of 100 linear systems for different values of N_{it}^{MAX} are provided.

Considering the data presented in Table V, we arrive at the following conclusions. For the smallest value of the threshold parameter, the preconditioner is recomputed quite many times. This results in that the total solution time increases because the time for computing a preconditioner is quite considerable (about 3500 ms). The average number of iterations per one linear system is very small (about 2), because it is limited by N_{it}^{MAX}.

TABLE V. CHARACTERISTICS OF SOLVING 100 LINEAR SYSTEMS FOR DIFFERENT VALUES OF N_{it}^{MAX}

N_{it}^{MAX}	**The number of recomputing M**	**Total number of iterations for all linear systems**	**Total solution time, ms**	**Acceleration w.r.t. the algorithm with the initial precondioner**
2	12	196	53673	0.77
3	5	249	30701	1.35
4	2	307	24039	1.72
5	2	415	29504	1.40
6	1	416	26040	1.59
7	1	387	24571	1.68
8	1	389	24978	1.66
9	1	487	28903	1.43
10	1	487	29018	1.43
11	1	496	29303	1.41
12	1	666	37122	1.11
13	0	825	41364	1.00

Thus, in the case considered, the total solution time is mainly determined by the time spent on computing the preconditioners, whereas the contribution of iterations is small. It should be emphasized that the so frequent recomputation of the preconditioners is actually not needed because it increases the total solution time (the speed up equals 0.77) as compared with the case where the initial preconditioner is never modified. For the largest value of N_{it}^{MAX}, the preconditioner is never recomputed (the speed up equals 1), and the total solution time is mostly determined by the iterations. However, the solution time is considerable despite the fact that iterations are not time consuming (about 50 ms) because the number of iteration increases about 4 times.

VII. ANIMATED RESPONSE

For visual display of changes in signal propagating along a multiconductor transmission line, animation of calculated plots is implemented. Simple example of two coupled lines connected as turn of meander line is shown in Figure VI. For calculation of the response on pulse signal for circuit diagram the theoretical base described in [11] is used. The calculation is reduced to the solution of linear algebraic system. The equation in the frequency-domain for a network containing *n* multiconductor transmission lines and lumped circuits at the ends of the lines can be written as

$$\mathbf{V}(s) = \left(s\mathbf{W} + \mathbf{H} + \sum_{k=1}^{n} \mathbf{D}_k \mathbf{Y}(s)_k \mathbf{D}_k^t\right)^{-1} \mathbf{E}(s), \qquad (1)$$

where **W**, **H** $\in \Re^{N\times N}$ are constant matrices describing the lumped memory and memoryless elements of network, respectively, and $\Re^N$ is the node-space of network; **E**(*s*) $\in \Re^N$ is a constant vector with entries determined by the independent voltage and current sources; $\mathbf{D}_k$=[i, j] with elements $l_{i,j} \in \{0,1\}$ where $i \in \{1,...,N\}$, $j \in \{1,...,2N_{COND}\}$ with a maximum of one nonzero in each row or column, is the selector matrix that maps the terminal currents of the distributed subnetwork to the nodal space of the linear network and N_{COND} is number of conductors of a transmission line; **V**(*s*) $\in \Re^N$ is the vector of node voltage waveforms appended by independent voltage source currents, linear inductor current waveforms of linear network; $\mathbf{Y}(s)_k \in \Re^{2N_{COND}\times 2N_{COND}}$ is the admittance matrix having complex dependency on frequency, which are described in terms of transmission line parameters.

However, the equation (1) allows calculating the signal only at the nodes of the circuit. Therefore, to calculate the signal propagating along a transmission line for a network we used equations:

$$\mathbf{V}(x)=\mathbf{S}_V(\mathbf{E}(x)\mathbf{C1}+\mathbf{E}(x)^{-1}\mathbf{C2}), \qquad (2)$$

$$\mathbf{I}(x)=\mathbf{S}_I(\mathbf{E}(x)\mathbf{C1}-\mathbf{E}(x)^{-1}\mathbf{C2}), \qquad (3)$$

where $\mathbf{S}_V$ is the matrix of modal voltages; $\mathbf{S}_I$ is the matrix of modal currents; **E**(*x*) is the diagonal matrix {exp(–$\gamma_1 x$), exp(–$\gamma_2 x$),…, exp(–$\gamma_{Ncond} x$)} and γ is the propagation constant, *x* is the coordinate along the transmission line; **C1**, **C2** are constant vectors.

To find **V**(*x*) and **I**(*x*) for each transmission line of network we must calculate **C1** and **C2** from equation

$$\begin{bmatrix}\mathbf{C1}\\ \mathbf{C2}\end{bmatrix} = \begin{bmatrix}\mathbf{S}_V & \mathbf{S}_V\\ \mathbf{S}_V\mathbf{E}(l) & \mathbf{S}_V[\mathbf{E}(l)]^{-1}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{V}(0)\\ \mathbf{V}(l)\end{bmatrix}, \qquad (4)$$

where **E**(*l*) is the diagonal matrix {exp(–$\gamma_1 l$), exp(–$\gamma_2 l$),…, exp(–$\gamma_{Ncond} l$)} and γ is the propagation constant of the transmission line, *l* is the length of the transmission line; **V**(0) and **V**(*l*) are constant vectors describing the voltage on the near and far ends of the transmission line, determined after the solution of equation (1).

For visual display of changes in signal propagating along a transmission line it is necessary to indicate the initial node A and the end node B on the circuit diagram (Figure VI). After assigning the nodes, the possible pathway of the pulse is determined automatically. To make the list of available nodes, we use the GoThrough algorithm, based on the depth-first search algorithm. The conductor, in which the signal propagates, is painted the color of the animated diagram of this signal waveform. Figure VII presents (in static) the voltage waveforms in the time domain for the circuit shown in Figure VI. One can see considerable changes of magnitude and distortions during the propagation of the pulse along the turn. The animated visualization is useful to locate the points of possible interference penetration for a circuit.

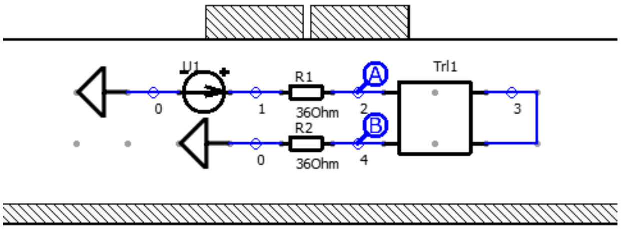

FIGURE VI. CROSS SECTION OF MEANDER LINE TURN AND CIRCUIT DIAGRAM OF THE TURN

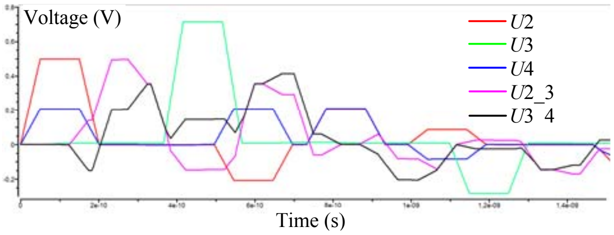

FIGURE VII. ANIMATION OF RESPONSE ON PULSE SIGNAL: U2, U3 AND U4 – RESPONSES IN NODES 2, 3 AND 4; U2_3(5) AND U3_4(5) – RESPONSES IN HALF-TURN CENTERS

VIII. Testing Tools

To speed up the software improvement process, development and testing should be performed at the same time and be inseparable. Engineers involved in testing should be consulted at after each release of the software and alert developers of any errors, checking the quality of the developed software more thoroughly. In addition, there is a need for the regression tests for the detection of errors in previously tested parts of the initial code of the software. For this purpose we developed the code that uses the script files as initial data. After the test, one can know which scripts had errors (results did not coincide with the reference). This approach can effectively detect and fix regression bugs, which reduces the time required to develop and debug new functionality.

Unfortunately, not all errors can be detected in the testing phase, in this regard, there is a need for automatic error reporting on user workstations, and these reports should be sent to the developers. For this reason, a library of open source - Crashrpt [12] was built in the software, allowing to create an archive with bug reports, screen shots and information about the operating software of a user. If an error leading to the closure of the software occurs, a user is offered to send the error report, with the ability to preview the contents of an archive. A report is sent using a POST request to the http-server of the developer. To process archive in the language php, received from Crashrpt, we implemented a processer that extracts information from the archive and adds it to Redmine software of management of projects and developer`s goals [13]. Reports contain complete information for effective determination of the causes of errors and user actions that led to errors.

IX. Conclusions

The main trends in the development of specialized software for the simulation of electromagnetic compatibility are given. New features of electromagnetic compatibility simulation TALGAT software that follow these trends are briefly described. Results of simulation for various interconnect structures are given. Improvement of simulation is described, concerning the main stages of simulation based on method of moments. Necessity of taking account for dependence of complex permittivity of dielectric substrate is shown. New algorithms are presented to accelerate multiple solutions of linear algebraic systems by use of particular changes of matrix entries. The obtained speed-up shows that the proposed approaches are effective. At last, recently designed testing tools are briefly described. Thus, the presented results systematically describe some new approaches for improvement of interconnects simulation based on method of moments.

Acknowledgment

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).

References

[1] J. Kim, E. Li,. “Special Issue on PCB Level Signal Integrity, Power Integrity, and EMC”, IEEE Transactions on Electromagnetic Compatibility, May 2010, vol. 52, no. 2, pp. 246–248.

[2] 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 Symposium, Montreal, Canada, August 13–17, 2001, vol. 1. pp. 151–155.

[3] T.R. Gazizov, “Calculation of capacitance matrix of three dimensional multiconductor system in multiple dielectric media”, Record of International Symposium on Electromagnetic Compatibility. Magdeburg, Germany, October 5–7, 1999, pp. 31–36.

[4] R.I. Ashirbakiev and V.K. Salov, "Adaptive iterative selection of the optimal segmentation of a conductor and dielectric boundaries in electrostatic problems", Doklady TUSUR, 2013, no. 3 (29), pp. 159-161.

[5] A.R. Djordjevich, R.M. Biljic, V.D. Likar-Smiljanic, T.K. Sarkar, “Wideband frequency-domain characterization of FR-4 and time-domain causality”, IEEE Transactions on Electromagnetic Compatibility, 2001, vol. 43, no. 4 pp. 662–666.

[6] S.P. Kuksenko and T.R. Gazizov, “Improvement of an algorithm for computing the capacitance matrices of structures of conductors and dielectrics by the method of moments in a range of values of the dielectric permeability,” Elektromagn. Volny Elektron. Sist., 10, 13‑21 (2012).

[7] R.R. Akhunov, S.P. Kuksenko, V.K. Salov, T.R. Gazizov, “Multiple iterative solution of linear algebraic systems with a partially varying matrix,” Journal of Mathematical Sciences, vol. 199, no. 4, June, 2014, pp. 381–385.

[8] V.K. Salov, T.R. Gazizov, O.A. Nikitina, “Convergence of multiple iterative solution of linear algebraic systems with a fully varying matrix using a single calculated initial preconditioner,” Innovative Information Technologies: Materials of the International scientific-practical conference. Part 2. / Ed. Uvaysov S.U.–M.: HSE, 2014. April 21–25, 2014, Prague, Czech – pp. 452–457.

[9] R.R. Akhunov, S.P. Kuksenko, V.K. Salov, T.R. Gazizov, “Sparse matrix storage formats and acceleration of iterative solution of linear algebraic systems with dense matrices,” Journal of Mathematical Sciences, vol. 191, May, 2013, pp. 19–27.

[10] R.R. Akhunov, S.P. Kuksenko, V.K. Salov, T.R. Gazizov, “Optimization of the ILU(0) factorization algorithm with the use of compressed sparse row format,” Journal of Mathematical Sciences, vol. 191, May, 2013, pp. 19–27.

[11] R. Achar and M.S. Nakhla, “Simulation of high-speed interconnects”, Proceedings of the IEEE, 2001, vol. 89, No. 5, pp. 693–728.

[12] http://code.google.com/p/crashrpt.

[13] http://www.redmine.org.