The Periodic Steady-State Analysis Based on Single-Step High Order Integration Methods

M.M. Gourary¹, S.G. Rusakov¹, S.L. Ulyanov^{1,2}, M.M. Zharov¹

¹Institute for Design Problems in Microelectronics of Russian Academy of Sciences, ippm@ippm.ru

²National Research University Higher School of Economics, Moscow Institute of Electronics and Mathematics

Abstract — In this paper the method of periodic steady-state analysis in analog integrated circuits is proposed. The periodic boundary-value problem is solved using shooting-Newton method, in which to solve the initial value problem the single step integration methods based on the Obreshkov formula are suggested to apply. The formulation of methods of order 1 up to 4 are obtained for charge oriented circuit equations. The formulas for computing sensitivity matrix are presented. The numerical examples of steady-state analysis are given which demonstrate the numerical accuracy and efficiency of the proposed method.

Keywords — circuit simulation, periodic steady-state, boundary value problem, shooting method, integration methods.

I. INTRODUCTION

The design of modern integrated circuits heavily requires efficient and accurate circuit simulation tools. One of the most important types of analysis that support the design of RF and high-speed circuits is determination of periodic steady state. The application of conventional transient analysis to find the periodic steady-state solution often results in a long simulation time and hence special purpose means are needed [1]. The periodic steady-state response of a nonlinear circuit corresponds to the steady-state solution of differential equations which can be found by solving a boundary-value problem with periodic boundary constraint [1, 2]. The standard approach to solve this problem is to apply finite-difference methods or shooting methods.

The finite-difference methods approximate the system of differential equations with a set of difference equations. The system of difference equations of large size is solved with respect to all time points simultaneously. If the Newton method is applied then operations with Jacobian matrix require a large amount of memory and time. This limits the use of these methods for simulation of medium or large sized circuits [1].

The shooting methods solve the boundary-value problem by solving the equivalent nonlinear algebraic problem with respect to the state vector at the beginning of one period. In fact the shooting methods transform the solution of a periodic boundary-value problem to the solution of sequence of initial value problems at one period

of input signal. The initial value problem is solved using algorithms of transient analysis. In comparison with finite-difference methods shooting methods require less memory because there is no need to store all time points.

However if the Newton method is applied to solve the nonlinear algebraic problem the computation of sensitivity matrix and the solution of linear system with dense Jacobian matrix are required to perform. The efficient technique for computation of sensitivity matrix was proposed in [3, 4]. The reduction of computational cost of the solution of linear system with dense Jacobian matrix can be obtained by iterative methods [5, 6].

Another factor that determines the computational cost of the shooting method is a numerical technique used to integrate differential equations on the period of input signal. To perform numerical integration of ordinary differential or differential-algebraic equations the comprehensive variable order and variable time step integration algorithms are used. The most commonly used methods are backward Euler, trapezoidal and backward-differentiation formulas (BDF) [7-9]. The first order backward Euler and the second order trapezoidal methods are A-stable and do not restrict time step [7, 10]. The common drawback of BDF methods is the lack of A-stability for order higher than 2. As a result high order methods are not implemented in modern simulators. However the usage of high order and A-stable methods can improve accuracy and speed up time-domain transient analysis.

This paper presents the shooting-Newton method, in which the single step integration methods based on the Obreshkov formula [11] are applied to solve the initial value problem. The main practical difficulties of implementation are connected with computation of high-order derivatives of modified nodal analysis (MNA) variables and partial derivatives needed to form the Jacobian matrix. In the case of linear circuits this task can be relatively easily solved leading to a system of algebraic equations with respect to MNA unknowns and their time derivatives [12]. However, the application to nonlinear circuits requires the computation of high-order derivatives of nonlinear terms in MNA formulation which represent the contributions from nonlinear device models. Hence besides conventional data commonly used in the transient

analysis, device models must provide the computation of high-order derivatives. With modern device models, which include nonlinear dependencies in several variables as well as different model behavior in different regions of device operation, this is extremely difficult, if not impossible.

In this paper the single step integration methods of order 1 up to 4 obtained by the Obreshkov formula are used [13]. In contrast to the previous approaches these methods do not require the computation of high-order derivatives and hence the analysis is based on standard device models existing in a circuit simulator.

II. SHOOTING-NEWTON METHOD

Assuming the charge oriented model and the modified nodal analysis the nonlinear circuit can be described by a system of differential equations [1,2]:

$$\dot{q}(v(t)) + i(v(t)) + u(t) = 0, \quad (1)$$

where $\dot{q}(v(t)) = \frac{dq(v(t))}{dt}$, q is the vector of charges and

fluxes, i is a vector of the conductive contributions to the circuit equations, and u is the vector of input currents. Each node contributes on equation arising from Kirchhoff's current law. In addition, there are branch equations for the inductors and voltage sources. i and q are functions of v , which apart from the node voltages, can also consist of inductor and independent voltage source currents [7].

If a nonlinear circuit is driven with periodic input signal $u(t) = u(t + T)$, where T is a period, the circuit will typically have a steady-state response satisfying the condition

$$v(0) = v(T). \quad (2)$$

The ODE system (1) with the condition (2) constitutes a boundary-value problem. The shooting method is used for solving boundary-value problems and hence for finding the periodic steady-state solution of (1).

The shooting method reduces the solution of a boundary-value problem (1), (2) to the solution of a system of algebraic equations with respect to the state vector at initial time $v(0)$. Introducing the state-transition function $\varphi(v(t_0), t_0, t)$ and assuming that $t_0 = 0$ one can obtain $v(T) = \varphi(v(0), 0, T)$. Then the condition (2) can be rewritten in the form of a system of algebraic equations

$$\varphi(v(0), 0, T) - v(0) = 0. \quad (3)$$

The shooting method can be thought of as iterative method applied to solving the nonlinear algebraic system (3), while the vector $v(T) = \varphi(v(0), 0, T)$ is determined from solution of initial value problem with appropriate initial values at every iteration [1,2].

Applying Newton-Raphson to the system (3) results in the linear system to be solved at every iteration

$$[\Phi(T) - E]\Delta v^j(0) = -(\varphi(v^j(0), 0, T) - v^j(0)), \quad (4)$$

where j is the iteration number, $\Delta v^j(0) = v^{j+1}(0) - v^j(0)$ is the Newton's update, $\Phi(T) = \frac{\partial \varphi(v(0), 0, T)}{\partial v(0)} = \frac{\partial v(T)}{\partial v(0)}$ is

a fundamental matrix of the equations

$$\frac{d}{dt}[C(v)\Phi(t)] + G(v)\Phi(t) = 0, \quad (5)$$

where $C(v) = \frac{\partial q(v(t))}{\partial v}$, $G(v) = \frac{\partial i(v(t))}{\partial v}$.

The state-transition function is computed by integrating (1) numerically. The matrix $\Phi(t)$ can be computed simultaneously. For example discretization (1) using implicit Euler method yields [1, 2]

$$r(v_{n+1}) = \frac{1}{h}[q(v_{n+1}) - q(v_n)] - i(v_{n+1}) + u_{n+1} = 0, \quad (6)$$

Here v_n , v_{n+1} are vectors at $t=t_n$ и $t=t_{n+1}=t_n+h$, h is time step. Applying Newton-Raphson method to the system (6) results in the linear system with respect to Newton's update

$$[G(v_{n+1}^j) + \frac{C(v_{n+1}^j)}{h}](v_{n+1}^{j+1} - v_{n+1}^j) = -r(v_{n+1}^j), \quad (7)$$

where $G(v_{n+1}^j) = \frac{\partial i(v_{n+1}^j)}{\partial v}$, $C(v_{n+1}^j) = \frac{\partial q(v_{n+1}^j)}{\partial v}$.

Differentiating (6) with respect to $v_0 = v(0)$ one can obtain

$$[G(v_{n+1}) + \frac{C(v_{n+1})}{h}] \frac{\partial v_{n+1}}{\partial v_0} = \frac{C(v_n)}{h} \frac{\partial v_n}{\partial v_0}, \quad (8)$$

Expression (8) provides an efficient computation of $\Phi(t)$ because the required matrices are already computed and factored during the solution of (6).

III. OBRESHKOV FORMULA

The initial value problem (IVP) to be solved can be written in the following form [10]

$$\begin{aligned} \dot{x}(t) &= f(t, x(t)) \\ x(t_0) &= x_0 \end{aligned} \quad (9)$$

Here $x(t)$, $\dot{x}(t)x$ are the vector of unknowns and of their time derivatives, f is a nonlinear vector function, t is time, x_0 is the initial value at $t=t_0$.

The conditions to guarantee the existence and uniqueness of the solution are assumed to be satisfied. It is assumed that $f(t, x)$ is continuous in its first variable and satisfies a Lipschitz condition in its second variable. Then for any x_0 , there exists a unique solution $x=x(t, x_0)$.

Numerical methods seek an approximation of the exact solution $x(t)$ at discrete time points $t_0, t_1, t_2, \dots$. Having obtained an approximation x_n at $t=t_n$ one-step methods

find an approximation x_{n+1} at $t=t_{n+1}=t_n+h$, where h is the time step.

The Obreshkov methods [11] are based on using the Hermite interpolation that provides linear relationship between x_{n+1} and time derivatives $x_{n+1}^{(i)} = d^{(i)}x(t_{n+1})/dt^{(i)}$.

The Obreshkov formula yields [11]

$$x_{n+1} - x_n = -\sum_{i=1}^m \alpha_i h^i x_{n+1}^{(i)} + \sum_{i=1}^l \beta_i h^i x_n^{(i)} + O(h^{l+m+1}), \quad (10)$$

where $x_n = x(t_n)$, $x_{n+1} = x(t_{n+1})$,

$$\alpha_i = (-1)^i \frac{(l+m-i)!}{(l+m)!} \frac{m!}{i!(m-i)!},$$

$$\beta_i = \frac{(l+m-i)!}{(l+m)!} \frac{l!}{i!(l-i)!}.$$

The formula (10) uses the values of the function and its l derivatives at t_n and its first m derivatives at t_{n+1} . From expression (10) it is clear that several methods with various order $p=l+m$ can be obtained. It has been proven that Obreshkov methods are A-stable with $m-2 \leq l \leq m$ and L-stable if $m-2 \leq l < m$ [12, 14].

We will restrict ourselves to the case of $l \leq 2$, $m \leq 2$. This can be seen below. Next table shows coefficients and stability properties of some methods.

Table 1.

Coefficients and properties of several methods with $l \leq 2$, $m \leq 2$.

order	l	m	α_1	α_2	β_1	β_2	stability
4	2	2	-1/2	1/12	1/2	1/12	A
3	1	2	-2/3	1/6	1/3	-	A,L
2	0	2	1	1/2	-	-	A,L
2	1	1	-1/2	-	1/2	-	A
1	0	1	-1	-	-	-	A,L

Note that the method with $l=0$, $m=1$ is the backward Euler, and $l=m=1$ yields the trapezoidal rule formulas.

IV. APPLICATION TO TRANSIENT ANALYSIS

Using chain rule one can obtain from (1)

$$C(v)\dot{v}(t) + i(v) + u(t) = 0, \quad (11)$$

Differentiation of (1) yields

$$\ddot{q}(v) + G(v)\dot{v}(t) + \dot{u}(t) = 0, \quad (12)$$

The transient analysis finds the solution of (1) over a specified time interval from specified initial conditions.

Applying the Obreshkov formula (2) one can obtain a system of discretized difference equations corresponding to a certain integration method order. Note that because the first and second derivatives of charge vector are available only the numbers m and l must be less or equal 2. In particular for $l=m=2$ one can obtain for charge vector

$$q(v_{n+1}) - q_n = -\alpha_1 h \dot{q}(v_{n+1}) - \alpha_2 h^2 \ddot{q}(v_{n+1}) + \beta_1 h \dot{q}_n + \beta_2 h^2 \ddot{q}_n, \quad (13)$$

where $q_n = q(v_n)$, $\dot{q}_n = \dot{q}(v_n)$, $\ddot{q}_n = \ddot{q}(v_n)$ are known.

Taking into account (1), (11) one can rewrite (13) in the form

$$\begin{aligned} r_1(v_{n+1}, \dot{v}_{n+1}) &= q(v_{n+1}) - q_n + \alpha_1 h [-i(v_{n+1}) - u_{n+1}] + \\ \alpha_2 h^2 [-G(v_{n+1})\dot{v}_{n+1} - \dot{u}_{n+1}] - \beta_1 h \dot{q}_n - \beta_2 h^2 \ddot{q}_n &= 0 \end{aligned} \quad (14)$$

Equation (12) yields

$$r_2(v_{n+1}, \dot{v}_{n+1}) = i(v_{n+1}) + \frac{C(v_{n+1})}{h} (h\dot{v}_{n+1}) + u_{n+1} = 0 \quad (15)$$

Expressions (14), (15) are nonlinear equations with respect to vectors v_{n+1} , and $\dot{v}_{n+1}$. The similar equations can be obtained for the cases $l=1$, $m=2$ and $l=0$, $m=2$.

Applying the Newton-Raphson method to solve nonlinear equations (14), (15) one can obtain the following linear system of equations

$$J(v_{n+1}^{(j)}, \dot{v}_{n+1}^{(j)}) \begin{bmatrix} v_{n+1}^{(j+1)} - v_{n+1}^{(j)} \\ h(\dot{v}_{n+1}^{(j+1)} - \dot{v}_{n+1}^{(j)}) \end{bmatrix} = - \begin{bmatrix} r_1(v_{n+1}^{(j)}, \dot{v}_{n+1}^{(j)}) \\ r_2(v_{n+1}^{(j)}, \dot{v}_{n+1}^{(j)}) \end{bmatrix}, \quad (16)$$

where $J = J(v_{n+1}^{(j)}, \dot{v}_{n+1}^{(j)})$ is the Jacobian matrix

$$J = \begin{bmatrix} C(v_{n+1}^{(j)}) - \alpha_1 h G(v_{n+1}^{(j)}) - \alpha_2 h^2 G'(v_{n+1}^{(j)}) \dot{v}_{n+1}^{(j)} & -\alpha_2 h G(v_{n+1}^{(j)}) \\ G(v_{n+1}^{(j)}) + C'(v_{n+1}^{(j)}) \dot{v}_{n+1}^{(j)} & C(v_{n+1}^{(j)})/h \end{bmatrix} \quad (17)$$

j is the iteration number and $G'(v) = \frac{\partial G(v)}{\partial v}$,

$$C'(v) = \frac{\partial C(v)}{\partial v}.$$

Introducing vectors $y_1 = v_{n+1}^{(j+1)} - v_{n+1}^{(j)}$,

$y_2 = h(\dot{v}_{n+1}^{(j+1)} - \dot{v}_{n+1}^{(j)})$, and omitting iteration index and functions arguments the linear system (16) in simplified notation has the following form

$$\begin{bmatrix} C - \alpha_1 h G - \alpha_2 h^2 G' \dot{v}_{n+1} & -\alpha_2 h G \\ G + C' \dot{v}_{n+1} & C/h \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -r_1 \\ -r_2 \end{bmatrix} \quad (18)$$

Now neglecting the terms with G' , C' in the Jacobian matrix (17) the final linear system of equations to be solved at every iteration is as follows

$$\begin{bmatrix} C - \alpha_1 h G & -\alpha_2 h G \\ G & C/h \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -r_1 \\ -r_2 \end{bmatrix} \quad (19)$$

Thus, these methods do not require the computation of high-order derivatives and hence can be implemented into a general purpose circuit simulator.

V. SHOOTING-NEWTON WITH OBRESHKOV METHODS

Now we shall address the periodic boundary-value problem with condition (2) for equation (1). Consider the computation of $\Phi(T)$ with initial condition $\Phi(0) = E$.

Differentiating (14) with respect to $v_0 = v(0)$ one can write

$$\begin{aligned} \frac{\partial r_1(v_{n+1}, \dot{v}_{n+1})}{\partial v_0} = & \left[C(v_{n+1}) - \alpha_1 h G(v_{n+1}) - \alpha_2 h^2 G'(v_{n+1}) \dot{v}_{n+1} \right] \frac{\partial v_{n+1}}{\partial v_0} + \\ & -\alpha_2 h^2 G(v_{n+1}) \frac{\partial \dot{v}_{n+1}}{\partial v_0} + \\ & \left[-C(v_n) + \beta_1 h G(v_n) + \beta_2 h^2 G'(v_n) \dot{v}_n \right] \frac{\partial v_n}{\partial v_0} + \\ & \beta_2 h^2 G(v_n) \frac{\partial \dot{v}_n}{\partial v_0} \end{aligned} \quad (20)$$

Differentiation of (15) with respect to $v_0 = v(0)$ yields

$$\begin{aligned} \frac{\partial r_2(v_{n+1}, \dot{v}_{n+1})}{\partial v_0} = & \left[G(v_{n+1}) + C'(v_{n+1}) \dot{v}_{n+1} \right] \frac{\partial v_{n+1}}{\partial v_0} + \\ & C(v_{n+1}) \frac{\partial \dot{v}_{n+1}}{\partial v_0} \end{aligned} \quad (21)$$

Doing in a similar way as before in obtaining linear system (19), one can write

$$\begin{bmatrix} C - \alpha_1 h G & -\alpha_2 h G \\ G & C/h \end{bmatrix} \begin{bmatrix} \frac{\partial v_{n+1}}{\partial v_0} \\ h \frac{\partial \dot{v}_{n+1}}{\partial v_0} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \quad (22)$$

where

$$p_1 = \left[C(v_n) - \beta_1 h G(v_n) \right] \frac{\partial v_n}{\partial v_0} - \beta_2 h G(v_n) h \frac{\partial \dot{v}_n}{\partial v_0} \quad (23)$$

$$p_2 = 0 \quad (24)$$

Thus the matrix $\Phi(t)$ can be efficiently computed by solving the linear system (22). Since the matrix of this linear system is already computed and factored in solving (16), the computation of $\Phi(t)$ requires only operations of forward and backward substitutions.

VI. SOLVING THE DENSE LINEAR SYSTEM

The linear system with dense matrix (4) is solved at each iteration of the shooting-Newton method. If direct methods based on LU decomposition are applied to solve (4), the number of operations grows with the cube of the number of unknowns N . Thus this approach will become computationally inefficient for analysis of large sized circuits. To reduce the computational cost when solving the linear system with dense matrix, the iterative methods based on Krylov subspace can be applied. The matrix-vector product computed at each GMRES iteration requires N^2 operations and hence the computational cost is of order N^2 .

To further reduce the computational cost, a matrix free approach was proposed [5]. This approach avoids forming the dense matrix $[\Phi(T) - E]$ and directly calculates the

matrix-vector product. It was shown that this approach reduces the cost of computing the matrix-vector product to nearly order N operations [5].

However, the matrix-free Krylov-subspace technique requires additional storage to store matrices $C(v_n)$ and the factored matrix $[G(v_{n+1}) + \frac{C(v_{n+1})}{h}]$ at each time point.

For large sized circuits this matrix storage can be significant.

Another disadvantage of GMRES method is related with its slow convergence for typical circuit simulation problems [6]. The efficiency and robustness of iterative techniques can be improved by using preconditioning [15].

Thus the GMRES method with preconditioner is applied to solve the linear system with dense matrix (4). The preconditioner based on incomplete LU factorization has computational cost about of order N^2 [15]. Another alternative is to apply FGMRES method that allows the preconditioner to vary from step to step and hence to apply another iterative process as a preconditioner.

VII. EXPERIMENTAL RESULTS

The algorithm of periodic steady state analysis implemented in experimental circuit simulator is based on shooting-Newton method and single-step Obreshkov methods.

In addition, the transient analysis (TRAN) and periodic steady state analysis with conventional integration methods (backward Euler and trapezoidal) are implemented in the circuit simulator (Table 2).

Table 2.

Implemented analyses.

name	Analysis	Method
TRAN	Transient	Backward Euler, trapezoidal
PSS	Periodic steady state	Shooting-Newton, Backward Euler, trapezoidal
PSS2	Periodic steady state	Shooting-Newton, Obreshkov methods

The periodic steady-state analysis is demonstrated on three circuit examples. Circuits 1, 2 are shown in Fig. 1, 2. These circuits are chosen as test circuits because their time constants determine long settling time. For example the transient response in the first circuit is shown in Fig. 3. It is clear that in order to determine the steady state the simulation over time interval of 4000-6000 periods of input signal is required. Here the period is 100 us. The input frequency in the second circuit is 60 Hz and the transient response is shown in Fig. 4. The third circuit is operational amplifier UA741 [1] driven by input signal with frequency 100 kHz. The transient response is given in Fig. 5.

The maximum time step in transient analysis is bounded by the value $0.1T$. The similar limitation on step size is applied in periodic steady-state analysis. Tables 3, 4 contain statistics of periodic steady-state analysis (number of shooting-Newton iterations #it-sn, number of Newton iterations #it-tran, total number of time points and number of accepted time points #tpnts). For comparison Table 5 shows the number of Newton iterations (#it-tran), total number of time points and number of accepted time points (#tpnts) in the transient analysis performed until steady state is reached.

The tables show that periodic steady-state analysis PSS2 is up to 10 times more efficient than PSS analysis with conventional integration methods by the criterion of total time points. In comparison the total number of time points in PSS2 is 25-800 times less than that in transient analysis. This wide range is explained by different integration interval required to reach the steady state. This interval is 6000, 100 and 1000 periods of input signal for circuit 1-3 respectively. The output signals in steady state obtained by PSS2 and TRAN analyses are shown in Fig. 6-8. Note the good agreement between results.

Table 3.

Statistics of PSS2 analysis.

Circuit	PSS2		
	#it-sn	#it-tran	#tpnts
1	2	1139	537(525)
2	5	1576	717(676)
3	3	644	325(325)

Table 4.

Statistics of PSS analysis.

Circuit	PSS		
	#it-sn	#it-tran	#tpnts
1	5	14040	5138(5083)
2	6	3541	1341(1320)
3	4	1797	733(717)

Table 5

Statistics of TRAN analysis.

Circuit	TRAN	
	#it-tran	#tpnts
1	1493534	464820(304863)
2	25604	9688(6576)
3	15240	5511(3575)

Fig. 1. Amplifier. $R1=7K$, $R2=3K$, $R3=2K$, $R4=1K$, $R5=1M$, $C1=0.1\mu F$, $C2=0.1\mu F$, $VDD=10V$, $V1=5\sin(2\pi 10^4 t)$.

Fig. 2. Rectifier. $R23=5$, $R40=1K$, $C12=1\mu F$, $C30=1000\mu F$, $C40=1000\mu F$, $L34=0.1H$, $V1=10\sin(2\pi 60t)$

Fig. 3. Transient response in amplifier

Fig. 4. . Transient response in rectifier

Fig. 5. Transient response in OpAmp UA741

Fig. 6. The output signal of amplifier in steady state

Fig. 7. The output signal of rectifier in steady state

Fig. 8. The output signal of UA741 in steady state

ACKNOWLEDGEMENT

This work was supported in part by the Russian Foundation for Basic Research under grant 19-07-00509.

REFERENCES

- [1] Aktualnye problemy modelirovaniya v systemah shemotekhnicheskogo proektirovaniya (Actual simulation problems in computer-aided circuit design systems) - Moscow: Nauka, 2003 (in Russian).
- [2] Kundert K.S., White J., Sangiovanni-Vincentelli A. Steady-State Methods for Simulating Analog and Microwave Circuits - Boston, Kluwer Academic Publishers, 1990.
- [3] Aprille T. J., Trick. T. N. A computer algorithm to determine the steady-state response of nonlinear oscillators. IEEE Trans. on Circuit Theory. 1972. Vol. CT-19. No. 4. P. 354-360.
- [4] Gourary M.M., Rusakov S.G., Zarudniy D.I. Modelirovaniye na ECVm periodicheskikh processov v integralnykh shemah (Computer simulation of periodic processes in integrated circuits). Avtomatika i vychislitel'naya tekhnika. 1973. No. 1. P. 83-85 (in Russian).
- [5] Telichevsky R., Kundert K., White J. Efficient Steady-State Analysis based on Matrix-Free Krylov-Subspace Methods. in Proc. of IEEE Design Automation Conf. 1995. P. 480-484.
- [6] Liu X., Yu H, Tan S. A robust periodic Arnoldi shooting algorithm for efficient analysis of large-scale RF/MM ICs. in Proc. of IEEE Design Automation Conf. 2010. P. 573-578.
- [7] Vlach J. and Singhal K. Computer Methods for Circuit Analysis and Design. - New York: Van Nostrand Reinhold, 1983.
- [8] Gear C. W. Numerical Initial Value Problems in Ordinary Differential Equations - Englewood Cliffs, NJ: Prentice-Hall, 1971.
- [9] Brayton R. K., Gustavson F. G., and Hachtel G. D., A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas. Proc. IEEE. 1972. vol. 60. No. 1. P. 98-108.
- [10] Butcher J. C. Numerical methods for ordinary differential equations, 2nd ed. - John Wiley & Sons, 2008.
- [11] Hairer E., Norsett S. P. and Wanner G. Solving Ordinary Differential Equations I: Nonstiff Problems - Berlin: Springer-Verlag, 1993.
- [12] Gad E., Nakhla M., Achar R., and Zhou Y. A-Stable and L-Stable highorder integration methods for solving stiff differential equations. IEEE Trans. Computer-Aided Design of Integrated Circ. Sys. 2009. vol. 28. issue 9. P. 1359-1372.
- [13] M.M. Gourary, S.G. Rusakov, S.L. Ulyanov, M.M. Zharov The Implementation of High-Order Single-Step Integration Technique Into Circuit Simulator. Proc. of IEEE East-West Design & Test Symposium, Kazan, Russia, 2018, P. 123-128.
- [14] Ehle B. L. A-Stable Methods and Padé Approximations to the Exponential. SIAM Journal on Mathematical Analysis, 1973. vol. 4. No. 4. P. 671-680.
- [15] Y. Saad. Iterative methods for Sparse Linear Systems, PWS publishing company, Boston, 1996.