Study of Two-Memcapacitor Circuit Model with Semi-Explicit ODE Solver

Denis N. Butusov, Valerii Y. Ostrovskii, Artur I. Karimov, Dmitriy A. Belkin
Department of Computer-Aided Design
Saint Petersburg Electrotechnical University "LETI"
Saint Petersburg, Russia
dnbutusov@etu.ru

***Abstract*—This article discusses software tools for studying non-linear dynamical systems. For a detailed analysis of the behavior of chaotic systems stepsize-parameter diagrams are introduced. A new self-adjoint semi-explicit algorithm for the numerical integration of differential equations is described. Two modifications of the proposed method are represented. A two-memcapacitor circuit is selected as a test dynamical system. Symmetry, accuracy and performance analysis of semi-explicit extrapolation ODE solver are considered in a series of computational experiments. Phase space of the two-memcapacitor circuit model, stepsize-parameter diagrams and dynamical maps are given as experimental findings.**

I. INTRODUCTION

The theory of elements, that exhibit the property of changing their main characteristics upon electrical stimuli in a nonvolatile manner, originates from the prediction of the fourth basic passive element in 1971 [1]. This element was called a *memristor* by L. O. Chua, who placed it on a par with resistors, capacitors and inductors. Later, in 1976, the mathematical description of the memristor was extended to a wider class of memristive systems [2], within the framework of which the modern terms of other passive two-terminal circuit elements, a *memcapacitor* and a *meminductor* [3], were introduced. A slightly different view on the relationship between the four basic electrical quantities is presented by J. Shen et al. in paper [4], where *transtors* and *memtranstors* are also included into the list of the basic elements.

Nowadays, ideal memristive elements are considered as a possible circuit description of resistive switching devices [3] – the promising electrical components in nanotechnology. It is necessary to note that relevant models of the real devices cannot be implemented as a single memristive system [5]. In order to achieve the realistic device behavior in a circuit model one can combine a basic memristive system with other circuit elements. Since present resistive switching devices are fabricated in the parallel-plate capacitor geometry, it reasonable to add a virtual capacitive element, a linear capacitor or a nonlinear memcapacitor, depending on observed operation in a specific real device. Additionally, due to the internal chemical processes in the device ionic subsystem, one can also expect an inclusion of a meminductive element as a battery-like component. Thus, memcapacitors and meminductors should receive no less attention than memristors during development of circuit models.

Electrical circuits with memristive elements are also interesting objects in the field of nonlinear dynamics. Nonlinear nature of these elements leads to complex phenomena, including deterministic chaotic behavior.

Mathematical models of nonlinear dynamical systems are usually represented by ordinary differential equations which rarely have analytical solutions. In this case numerical ODE solvers are widely used being the basis of analysis tools, from calculation of Lyapunov exponents to the construction of multidimensional dynamical maps. However, most of researchers do not consider the properties of finite difference chaotic schemes obtained with various integration methods. In the process of chaotic systems simulation, the properties of the applied discrete operator can make a significant impact on model behavior [6]. Neglect of this fact may lead to incorrect simulation results in SPICE environments and packages for modeling of nonlinear dynamical systems.

The goal of this study is to propose and investigate the performance of a new semi-explicit scheme, suitable for solving nonlinear ODEs and estimate its properties on the two-memcapacitor circuit model in comparison with other integration methods. The investigation of oscillation modes and the detection of hidden features of the circuit model are also of our particular interest.

II. TWO-MEMCAPACITOR CIRCUIT

The paper [7] presents a chaotic circuit obtained by introducing several changes into the Chua's circuit. In this circuit (Fig. 1) the Chua's diode is replaced by a negative conductance $-G_0$ (active element), and the both capacitors are substituted by memcapacitors C_{m1} and C_{m2} (nonlinear energy-storage elements). The circuit also includes an inductor L and a resistor with conductance G, just as the origin.

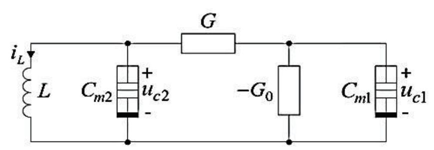

Fig. 1. Schematic of the circuit with two memcapacitors

The model of each smooth memcapacitor [7] is represented by the equations:

$$\begin{cases} u_c = (m + n\sigma^2)q \\ \varphi_c = m\sigma + (n\sigma^3/3) = a\sigma + b\sigma^3 \end{cases} \tag{1}$$

where σ is the time integral of the memcapacitor charge q, and φ is the time integral of the voltage u_c across the terminals of the memcapacitor.

Applying the Kirchhoff's law and the volt–ampere characteristics of the circuit elements one can obtain a system of ordinary differential equations:

$$\begin{cases} L\dfrac{di_L}{dt} = u_{c2} \\ \dfrac{dq_1}{dt} = G(u_{c2} - u_{c1}) + G_0 u_{c1} \\ \dfrac{dq_2}{dt} = G(u_{c1} - u_{c2}) - i_L \end{cases} \tag{2}$$

Applying integral operation, (2) can be transformed to the following form:

$$\begin{cases} L\dfrac{dq_L}{dt} = \varphi_{c2} \\ \dfrac{d\sigma_1}{dt} = G(\varphi_{c2} - \varphi_{c1}) + G_0 \varphi_{c1} \\ \dfrac{d\sigma_2}{dt} = G(\varphi_{c1} - \varphi_{c2}) - q_L \end{cases} \tag{3}$$

Finally, defining $x = q_L$, $y = \sigma_1$, $z = \sigma_2$, $c = 1/L$, $d = G_0$, $e = G$ and introducing a_1, b_1, a_2, b_2 referred to (1), the system described by (3) can be normalized:

$$\begin{cases} \dot{x} = c(a_2 z + b_2 z^3) \\ \dot{y} = (d - e)(a_1 y + b_1 y^3) + e(a_2 z + b_2 z^3) \\ \dot{z} = e(a_1 y + b_1 y^3 - a_2 z - b_2 z^3) - x \end{cases} \tag{4}$$

The system (4) can be used as a test problem for research of numerical integration methods since it has complex nonlinearities and chaotic solution for many parameter values. According to [7], for the initial conditions (0.1, 0, 0) the possible parameters of the system to exhibit chaotic behavior are: $a_1 = 0.25$, $b_1 = 0.6$, $a_2 = -0.17$, $b_2 = 10$, $c = 8.96$, $d = 4$, $e = 7$.

III. Integration methods

A method for integration of chaotic ODEs should satisfy several requirements. One of the most important is that its ability to drive a system into a quasi-chaotic (long-periodical oscillations) mode should be as less as possible. Also, it has to be numerically efficient since computation of bifurcation diagrams and dynamical maps is a complicated task.

From [8], a family of semi-implicit extrapolation methods is known. These methods are derived from the Euler-Cromer algorithm and are called the *D*-methods ("*D*" is for "diagonally implicit"). They usually need less computational efforts than the other extrapolation methods and are better suited for chaotic problems [9]. These methods exist for systems of order $N \geq 2$. For a general form of IVP,

$$\frac{d}{dt}\mathbf{x} = \mathbf{F}(t, \mathbf{x}) \tag{5}$$

where a bold font denotes a vector, the basic *D*-method is written as follows.

Each derivative of a state variable x_i is related to a right-hand side function as

$$\frac{d}{dt}x_i = f_i(t, x_1, x_2 ... x_i ... x_N) \tag{6}$$

Integrate (5) consequently with respect to x_1, x_2,… using the explicit "Euler-Cromer" integration, when previously calculated state variables are taken into the right-hand side function:

$$x_{i,n+1} = x_{i,n} + hf_i(t, x_{1,n+1}, x_{2,n+1} ... x_{i,n} ... x_{N,n}) \tag{7}$$

We take "Euler-Cromer" in quotes since the original Euler-Cromer method is usually applied only to Hamiltonian systems. The *D*-methods are suitable for a general case of IVP. Denote $f_i(t_n, \mathbf{x}_n) = f_{i,n}$. The Taylor expansion of $x_{i,n+1}$ on one integration step of (7) yields

$$\begin{aligned} & x_{1,n+1} = x_{1,n} + hf_{1,n} \\ & x_{2,n+1} = x_{2,n} + hf_{2,n} + h^2 f_{1,n} f'_{2,n} + O(h^3) \\ & ... \\ & x_{N,n+1} = x_{N,n} + hf_{N,n} + h^2 \sum_{j=1}^{N-1} f_{j,n} f'_{N,n} + O(h^3) \end{aligned} \tag{8}$$

From (8) one can see, that as an ordinal number of a state variable increases, the quantity of terms by h^2 increases correspondingly. The method (7) is called the explicit *D*-method $D_E(t,h,\mathbf{x})$. To complete the Taylor series, an application of an *adjoint* method is required. This method is implicit and will be denoted as $D_I(t,h,\mathbf{x})$. The implicit *D*-method is almost similar to (7), but the integration of each variable is implicit, and their order is reverse x_N, x_{N-1}…

$$x_{i,n+1} = x_{i,n} + hf_i(t, x_{1,n+1}, x_{2,n+1} ... \overbrace{x_{i,n+1}}^{implicit} ... x_{N,n}) \tag{9}$$

This gives another Taylor expansion

$$\begin{aligned} & x_{N,n+1} = x_{N,n} + hf_{1,n} + h^2 f_{N,n} f'_{N,n} + O(h^3) \\ & x_{N-1,n+1} = x_{N-1,n} + hf_{N-1,n} + \\ & \quad + h^2\left(f_{N-1,n} f'_{N-1,n} + f_{N,n} f'_{N-1,n}\right) + O(h^3) \\ & ... \\ & x_{1,n+1} = x_{1,n} + hf_{1,n} + h^2 \sum_{j=1}^{N} f_{j,n} f'_{1,n} + O(h^3) \end{aligned} \tag{10}$$

Composition of (7) and (9) with a stepsize $h/2$

$$D_2(t,h,\mathbf{x}) = (D_{II} \circ D_I)(t, \frac{h}{2}, \mathbf{x})$$

completes Taylor series for each state variable. Moreover, thus obtained D_2-method is symmetric which follows from the fact that the integration (8) with negative stepsize yields the integration (10) and vice versa. Also, this method is reversible for ρ-reversible systems. As its accuracy order is 2, the abbreviation D_2 is used for it.

Implicit integration in (6) can be inconvenient in case the function $f_i(t_n,\mathbf{x}_n)$ is nonlinear. This motivates to replace an implicit calculation by an explicit one in a following way.

For simpler notation, denote $f_i(t, x_{1,n+1}, x_{2,n+1},\ldots, x_{i,n+1},\ldots, x_{N,n}) = g(x_{i,n+1})$. Recall that the expression

$$x_{i,n+1} = x_{i,n} + hg(x_{i,n+1}) \tag{11}$$

can be expanded as

$$x_{i,n+1} = x_{i,n} + hg(x_{i,n} + hg(x_{i,n} + hg(.x_{i,n} + hg(...))))$$

This is an infinite series of substitutions which is equivalent to the implicit formula (11). But, to achieve a second order, only one substitution is required, which allows using an approximation

$$\begin{aligned} \bar{x}_{i,n+1} &= x_{i,n} + hg(x_{i,n}) \\ x_{i,n+1} &= x_{i,n} + hg(\bar{x}_{i,n+1}) \end{aligned} \tag{12}$$

instead of (11). Extrapolation methods of higher order based on the D_2 integrator require a quantity of expansions (12) proportional to their order of accuracy. Such a fully explicit method retains symmetry. An abbreviation "SED" ("SE" is for semi-explicit) is used for D_2-method with the explicit substitution (12) further.

The D_2-method is then used as a basic method for extrapolation. The idea of extrapolation is as follows. Suppose, a solution on one "macro" step h should be obtained with accuracy $O(h^{2k})$. Take a sequence $\{n_j\}$of natural numbers which will be divisors of the stepsize, for instance,

$$\{n_j\} = 1,2,3,4...$$

Compose a first column of a table of solutions on one "macro" time step $\mathbf{x}(t_n+h) = T_{j1}$ obtained by iterating the basic method with "micro" stepsizes h/n_j

$$\begin{array}{l} T_{11} \\ T_{21}T_{22} \\ T_{31}T_{32}T_{33} \\ \ldots\ldots\ldots T_{kk} \end{array}$$

The second and the following columns are obtained by the Aitken-Neville extrapolation formula

$$T_{j,k}(t_{N+1}) = T_{j,k\ 1} \quad \frac{T_{j,k-1} - T_{j-1,k-1}}{(\frac{n_j}{n_{j-k\ 1}})^p - 1},$$

where p equals 2 for symmetric methods like the D_2-method.

From the table, $x_{N+1} = T_{k,k}$ is a solution. The chosen sequence $\{n_j\}$ affects the efficiency of the method, but usually a simple row of doubled natural numbers is enough.

Consider simple 2[nd] order system integration as an illustrative example for the D_2 method. Let the system be as

$$\begin{aligned} \dot{x} &= f(x,y) \\ \dot{y} &= g(x,y) \end{aligned}.$$

For this case, the D_2 method will be written as

$$\begin{aligned} x_{n+1/2} &= x_n + hf(x_n, y_n) \\ y_{n+1/2} &= y_n + hg(x_{n+1}, y_n) \\ y_{n+1} &= y_{n+1/2} + hg(x_{n+1}, y_{n+1}) \\ x_{n+1} &= x_{n+1/2} + hf(x_{n+1}, y_{n+1}) \end{aligned},$$

while the SED_2 algorithm will look as

$$\begin{aligned} x_{n+1/2} &= x_n + hf(x_n, y_n) \\ y_{n+1/2} &= y_n + hg(x_{n+1}, y_n) \\ \bar{y}_{n+1} &= y_{n+1/2} + hg(x_{n+1}, y_n) \\ y_{n+1} &= y_{n+1/2} + hg(x_{n+1}, \bar{y}_{n+1}) \\ \bar{x}_{n+1} &= x_{n+1/2} + hf(x_n, y_{n+1}) \\ x_{n+1} &= x_{n+1/2} + hf(\bar{x}_{n+1}, y_{n+1}) \end{aligned}.$$

Consider an integration of the memcapacitor chaotic problem (4) by D_2 method. Written in C-like code, it is as shown in Listing 1.

Listing 1. Code for the memristive problem (4)

```
h=h/2;
float tmp;
int i;
int k=4;
x=x+h*c*(a2*z+b2*z**3);
y=y+h*((d-e)*(a1*y+b1*y**3)+...
+e*(a2*z+b2*z**3));
z=z+h*(e*(a1*y+b1*y**3-a2*z-b2*z**3)-x);
tmp=z;
for (i=0; i<k; i++)
z=tmp+h*(e*(a1*y+b1*y**3-a2*z-b2*z**3)-x);
tmp=y;
for (i=0; i<k; i++)
y= tmp+h*((d-e)*(a1*y+b1*y**3)+...
+e*(a2*z+b2*z**3));
x=x+h*c*(a2*z+b2*z**3);
```

For fully separable systems when derivatives have simpler form like

$$\frac{d}{dt}x_i = f_i(t) + \sum_{j=1}^{N} f_{ij}(x_j)$$

a more compact code can be written resulting in sufficient computational economy, see Listing 2.

Listing 2. Alternative code for the memristive problem (4)

```
h=h/2;
float tmp;
int i;
int k=4;
x=x+h*c*(a2*z+b2*z**3);
y=y+h*((d-e)*(a1*y+b1*y**3)+...
+e*(a2*z+b2*z**3));
z=z+h*(e*(a1*y+b1*y**3-a2*z-b2*z**3)-x);
tmp=z+h*(-x+e*(a1*y+b1*y**3));
for (i=0; i<k; i++)
z=tmp+h*e*(-a2*z-b2*z**3);
tmp=y+h*e*(a2*z+b2*z**3);
for (i=0; i<k; i++)
y=tmp+h*(d-e)*(a1*y+b1*y**3);
x=x+h*c*(a2*z+b2*z**3);
```

Notice that the number of calculations inside the loops is reduced through some pre-calculations. However, this technique can be implemented for ODEs with partially separable right-hand side functions.

IV. NUMERICAL RESULTS

All computational experiments described in this section were carried out in NI LabVIEW 2017 environment on the desktop class PC (Intel Core i5-4590, 8GB RAM, OS Windows 10).

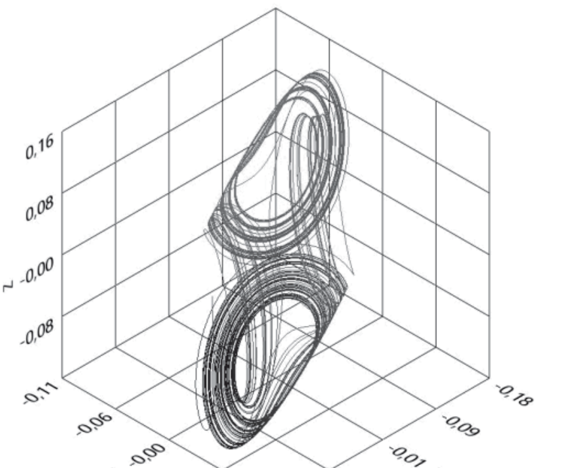

Fig. 2. Chaotic attractor of the system (4)

First, we obtained the chaotic attractor of the system (4), shown in Fig. 2. Simulation was performed with the semi-explicit integration method (SED) for the parameters and the initial conditions from Section II.

The system (4) was used as a test problem for efficiency estimation of integration methods. The main motivation of this part of the study was that construction of bifurcation diagrams and dynamical maps is a long, time-consuming process. Reducing computation time in nonlinear problems simulation is a challenging task. There is the evidence mentioned above that the semi-explicit methods being used as basic methods in extrapolation schemes have the best performance for many chaotic problems compared to the other methods. We tested several adaptive stepsize extrapolation solvers of order 8 based on the following basic methods: explicit midpoint rule (EMP), modified explicit midpoint rule (MEMP), linearly implicit midpoint rule (LIMP) and two versions of the semi-explicit method, corresponding to codes in Listing 1 (SED-1) and Listing 2 (SED-2).

The experimental results are shown in Fig. 3. We chose computational time as performance criterion (vertical axis) since it includes both evaluation time of right-hand side function and additional overheads caused by extrapolation procedure.

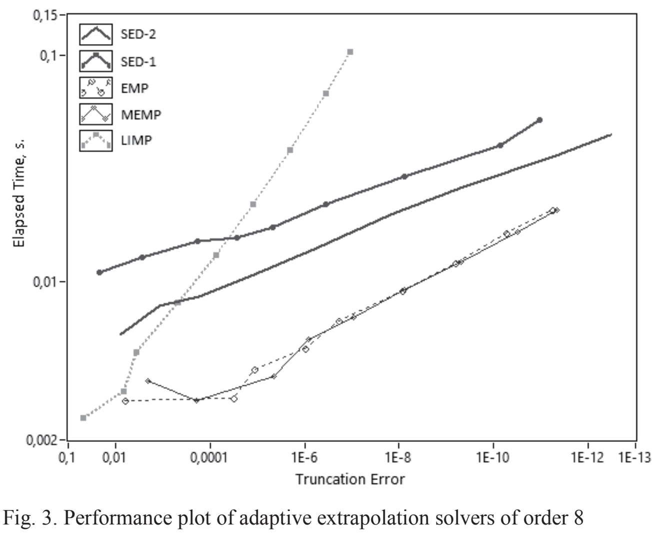

Fig. 3. Performance plot of adaptive extrapolation solvers of order 8

It can be seen that the performance of semi-explicit methods stands between implicit (LIMP) and explicit (EMP and MEMP) methods. Also, as we predicted, optimized SED-2 algorithm is more efficient than SED-1. However, the accuracy achieved by SED methods on double precision overcomes that of explicit solvers. Moreover, the slope of the efficiency line of SED-2 is lower than the slope of MEMP/EMP line which means that SED-2 code is likely to outperform explicit methods when extended data type and higher accuracy order are used.

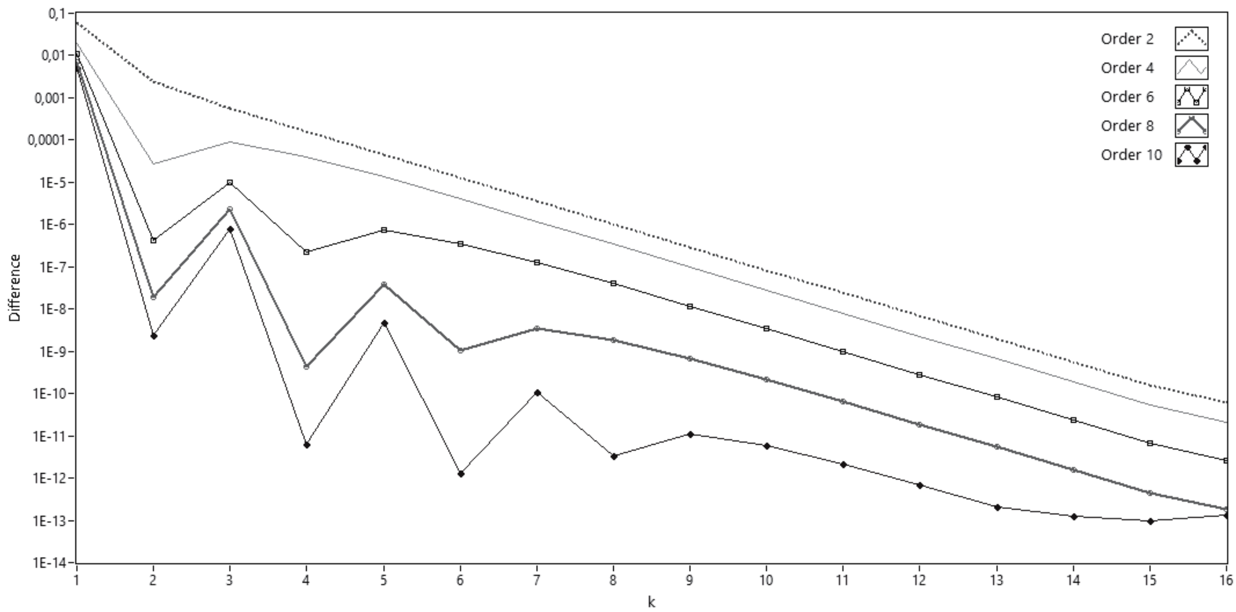

Fig. 4. Plots of difference between D- and SED-based extrapolation solvers

In Listings 1 and 2 the number of loops k is chosen according to results displayed in Fig. 4. Since the SED algorithm is the approximation of the D_2 for systems where diagonal implicitness cannot be expressed analytically, an increase in the number of calculations, determined by k, leads to their numerical convergence. One can see that even values of k correspond to reasonable minimum of difference for methods of respective orders. Thus efficient value of k should be matched with the accuracy order of extrapolation solver, e.g. $k = 2$ for order 4 and 6, $k = 4$ for order 8 etc. The paper [10] presents the proof of stability, symmetry and time-reversibility of the D_2-mehods, due to mentioned convergence the SED-methods also share these properties.

Further we present the results of multidimensional bifurcation analysis. Fig. 5 illustrates two-dimensional bifurcation diagrams (dynamical maps) for the system (4) in resolution 800x800 pixels. In order to plot dynamical maps we need to obtain a set of 1D bifurcation diagrams first and then analyze it to determine a number of limit cycles that the system has for each pair of parameter values. The darkest tint in the diagrams corresponds to chaotic behavior. Let us note the fact that the application of the SED methods made it possible to plot this diagrams accurately for an acceptable time compared to implicit methods.

Study of two-dimensional stepsize-parameter diagrams shows some interesting results. In Fig. 6 stepsize-parameter diagrams with respect to the parameter a_2 are presented. Dark color corresponds to chaotic regions. A special property of the semi-explicit method which cannot be found in the explicit or implicit methods can be seen in Fig. 6 (a). When the stepsize is small, a behavior of the system is chaotic during the interval of $a_2 \in [-0.22, -0.14]$, which is approximately equal to that of the continuous system. But, when the stepsize exceeds 0.3, the region of the chaotic behavior expands on the larger interval of parameter values. This change of chaotic properties is undesirable when an accurate simulation is needed but can be useful for a simple discrete chaotic map construction which would be able to work in a higher range of parameter values. Fig. 6 (b) corresponds to LIMP method and demonstrates the superior stability of this method comparing to the other tested methods. However, in a case of extrapolation solver the stability of LIMP algorithm can be slightly reduced. Fig. 6 (c) and Fig. 6 (d) refer to EMP and MEMP methods, respectively. They both lose stability for stepsizes higher than 0.7.

V. Conclusion

In this paper the semi-explicit modification of the D_2-method is proposed. The study of SED method shows that it can be more efficient than the other extrapolation methods on extended data types, especially for separable dynamical systems. These finite-difference schemes are symmetric and time-reversible for ρ-reversible ODEs. The main advantage of proposed methods is the possibility to construct single-step explicit composition solvers, which will be the subject of our further research. A two-memcapacitor chaotic circuit was simulated by SED method. Stepsize-parameter diagrams were introduced as a valuable tool for studying the influence of the chosen discrete operator to the finite-difference scheme behavior. It was found that the proposed semi-implicit method causes a special non-trivial behavior on large stepsizes, sufficiently changing the bifurcation map of the system which can be used in discrete chaotic pseudo-random generators. Despite the fact that numerical efficiency of the semi-explicit ODE solver for considered problem is relatively lower than efficiency of explicit solvers, its superior stability, symmetry and time-reversibility can be useful for many cases of nonlinear problems where application of semi-implicit or fully implicit methods is not possible.

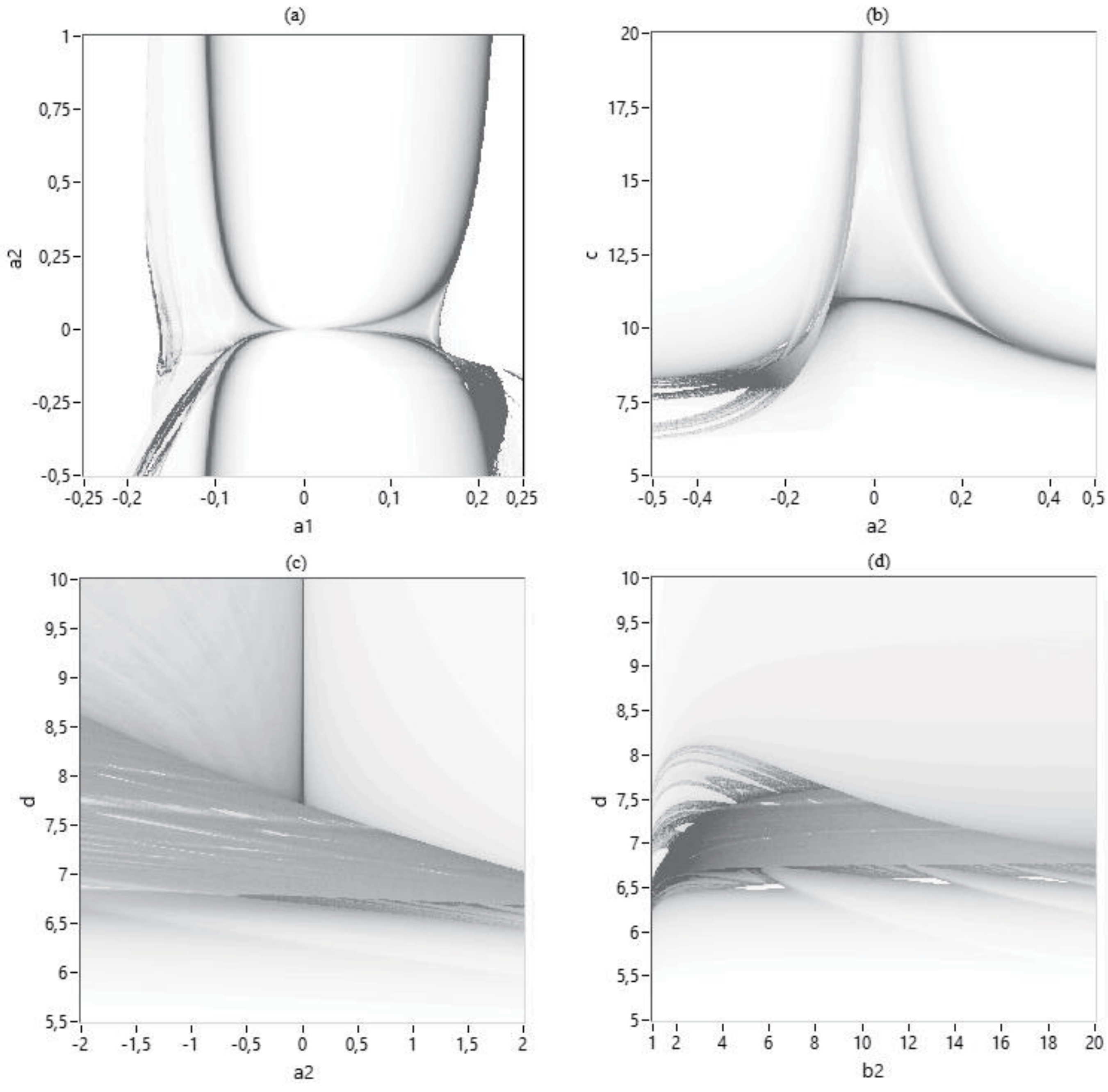

Fig. 5. Dynamical maps of the system (4); a1, a2, b2, c, d are the parameters of system (4)

ACKNOWLEDGMENT

The reported study was partially supported by RFBR, research project No. 17-07-00862.

REFERENCES

[1] L. O. Chua, “Memristor-the missing circuit element”, *in Proc. IEEE Transactions on Circuit Theory*, vol. CT-18, no. 5, Sept. 1971, pp. 507–519.

[2] L. O. Chua and S. M. Kang, “Memristive devices and systems”, *in Proc. IEEE*, vol. 64, no. 2, Feb. 1976, pp. 209–223.

[3] M. Di Ventra, Y. V. Pershin, and L. O Chua, “Circuit elements with memory: memristors, memcapacitors and meminductors”, *in Proc. IEEE*, vol. 97, Jan. 2009, pp. 1717–1724.

[4] J. Shen, D. Shang, Y. Chai, Y. Wang, J. Cong, S. Shen, L. Yan, W. Wang, and Y. Sun, “Nonvolatile multi-level memory and boolean logic gates based on a single memtranstor,” *Physical Review Applied*, vol. 6, no. 6, Dec. 2016, pp. 064028.

[5] Y. V. Pershin and M. Di Ventra, “Memory effects in complex materials and nanoscale systems,” *Advances in Physics*, vol. 60, Mar. 2011, pp. 145–227.

[6] A. I. Karimov, D. N. Butusov, V. G. Rybin, and T. I. Karimov, “The study of the modified Chirikov map”, *in Proc. XX IEEE Int.Conf. on Soft Computing and Measurements (SCM - 2017)*, May 2017, pp. 341–344.

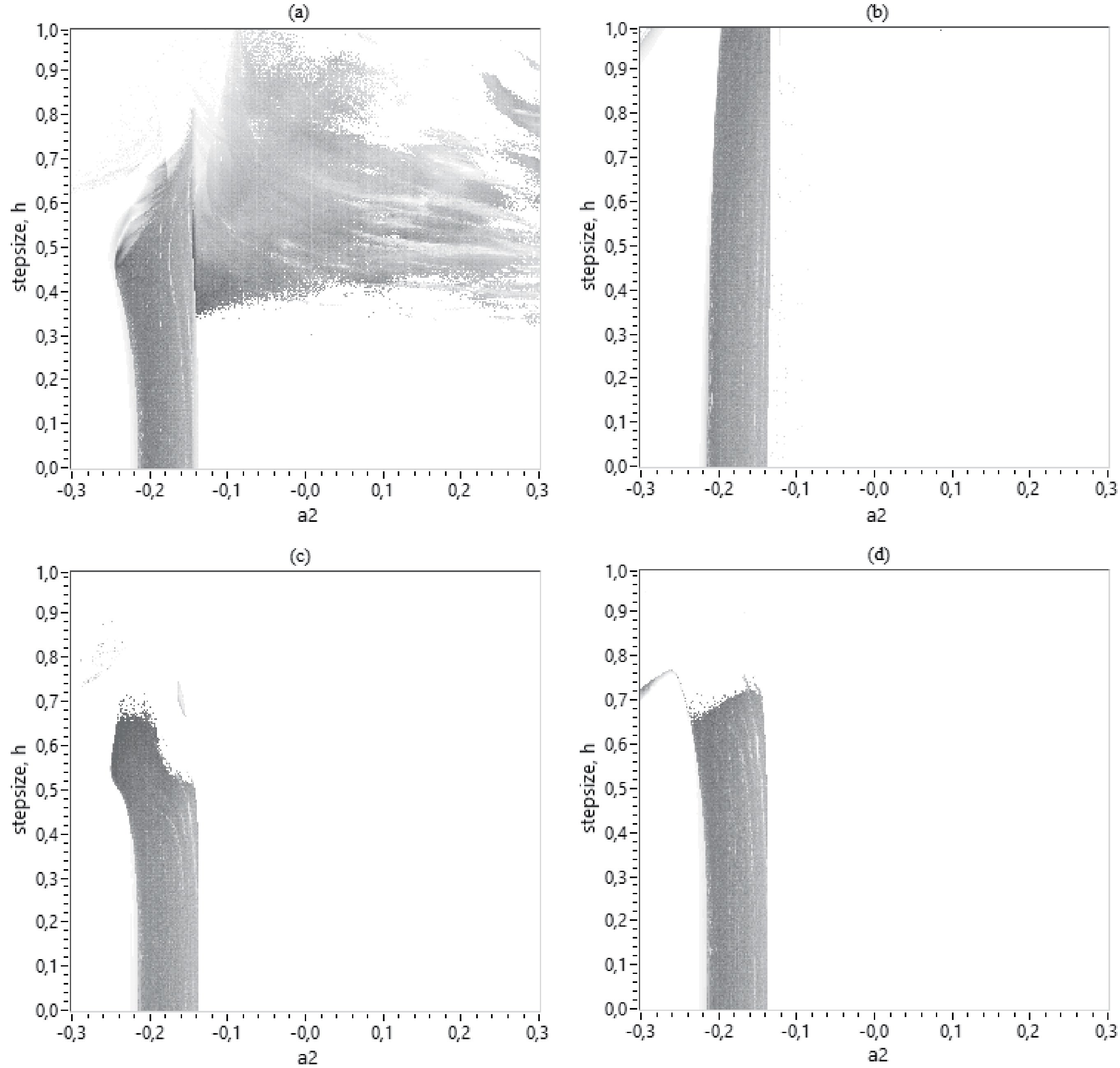

Fig. 6. Stepsize-parameter diagrams for SED-2 (a), LIMP (b), EMP (c) and MEMP (d) basic methods

[7] J. Mou, K. Sun, J. Ruan, and S. He, "A nonlinear circuit with two memcapacitors", *Nonlinear Dynamics*, vol. 86, no. 3, Nov. 2016, pp. 1735–1744.

[8] D. N. Butusov, A. V. Tutueva, and E. S. Homitskaya, "Extrapolation Semi-implicit ODE solvers with adaptive timestep", *in Proc. XIX IEEE Int.Conf. on Soft Computing and Measurements (SCM - 2016)*, May 2016, pp. 137–140.

[9] D.N. Butusov, V.Y. Ostrovskii, and D.O. Pesterev, "Numerical analysis of memristor-based circuits with semi-implicit methods", *in Proc. 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference (2017 ElConRus)*, Feb. 2017, pp. 271–276.

[10] D.N. Butusov, A.I. Karimov, and A.V. Tutueva, "Symmetric extrapolation solvers for ordinary differential equations,", *in Proc. 2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (2016 ElConRusNW)*, Feb. 2016, pp. 162–167.