IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 5, 2019, accepted August 12, 2019, date of publication August 19, 2019, date of current version August 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936175

Computing Two-Parameter Bifurcation Diagrams for Oscillating Circuits and Systems

WIESLAW MARSZALEK[1], HELMUT PODHAISKY[2], AND JAN SADECKI[1]
[1]Institute of Computer Science, Opole University of Technology, 45-758 Opole, Poland
[2]Institut für Mathematik, Martin-Luther-Universität, 06108 Halle (Saale), Germany

Corresponding author: Wieslaw Marszalek (w.marszalek@ po.edu.pl)

ABSTRACT Various oscillating (periodic and chaotic) circuits and systems show interesting responses whose nature changes with varying parameters. It often happens that a change of one element (i.e. resistor) of a circuit or system may cause a simultaneous change of two (or more) coefficients in the underlying mathematical model (i.e. a system of nonlinear ordinary differential equations, or ODEs). In this paper we present two-parameter bifurcation diagrams of such circuits and systems, obtained when two parameters vary simultaneously. Four different numerical techniques are applied to two selected dynamical systems (an active oscillating circuit with a memristive element and an electric arc circuit). The focus of this paper is on the computationally intensive calculations rather than on analytical analysis of the oscillatory responses. Two-parameter bifurcation diagrams require solving systems of nonlinear ODEs several hundred thousand (or even a few million) times (depending on the assumed resolution), plus additional work to distinguish periodic solutions from chaotic ones. Our computations are done using various combinations of the C++, Fortran/Python and Julia environments with Runge-Kutta order-4 and order-5 numerical solvers and the 0-1 test for chaos. Several two-parameter bifurcation diagrams are presented.

INDEX TERMS Julia and Python programming, oscillatory systems, periodic and chaotic signals, two-parameter bifurcation diagrams.

I. INTRODUCTION

Many oscillating dynamical systems in physics, electrical, mechanical, chemical, biological and financial processes bifurcate with varying parameters. One-parameter bifurcation diagrams are typical graphical representations of those bifurcations, showing the nature of responses (i.e. periodic, chaotic, asymptotically stable or unstable) [1]–[11]. Within the periodic responses one typically identifies the type of periodic response (i.e. period-n[1] or mixed mode oscillations). Periodic responses are further analyzed through local maximum (or minimum) values, while chaotic ones have positive Lyapunov exponents in certain intervals of a varying parameter. Specific mixed-mode L^s periodic oscillations are characterized by the natural numbers L and s denoting the number of large and small amplitude oscillations in one period [12], [13]. Other techniques used in analyzing oscillating circuits and systems are the Poincaré return maps [14], Lyapunov exponents [4], [15], machine learning recurrence method [16], [17] and the graphs of the K values in the 0-1 test for chaos ($K \approx 0$ for periodic responses while $K \approx 1$ for chaotic ones) [18]. Each of the above tools gives different information, and one usually uses two or three of those tools to fully characterize the analyzed responses.

A more challenging task (mostly due to the computing time and memory requirements) is to compute two-parameter bifurcation diagrams illustrating how a nonlinear circuit or system behaves when two parameters of its model vary simultaneously [19]–[21]. In this paper we present such two-parameter diagrams obtained by using four different numerical techniques (tools). Those techniques are applied to two particular electrical circuits whose mathematical models are systems of three ODEs with either one or two, rather mild, nonlinear terms. A mixed bag of tools is used in the four techniques. We begin with the classical Runge-Kutta algorithm of order 4 used with a C++ code in the first technique. Then, a more advanced and accurate Runge-Kutta

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Lai.

[1]*Period-n* response is a periodic response with *n* local maximum values in one period.

algorithm of order 5 (Dormand/Prince algorithm) is used in the Fortran and Python codes (technique II) and Julia programming language (technique III). Finally, a combination of order 4 Runge-Kutta algorithm and the 0-1 test for chaos [22]–[25] is used in technique IV. The goal of our paper is to test the four techniques and compare the obtained two-parameter bifurcation diagrams. To the authors knowledge no such comparison has been done thus far, although various single algorithms have been used to create two-parameter bifurcation diagrams. Creation of two-parameter bifurcation diagrams requires solving the underlying model (system of nonlinear ODEs) several hundred thousand or even a few million of times (depending on the assumed resolution in the two-dimensional space of parameters) [26]. Such a resolution of the two-parameter space is impacted by the circuits or systems under consideration. Some of them change rapidly their responses with minuscule changes of parameters, while others show less dramatic changes in the bifurcation diagrams. Another issue impacting the resolution is our own goal - a desire to compute the bifurcation diagrams to a certain level of accuracy (zooming), particularly when a circuit or system is of a fractal type. See, for example, the modified Chua's circuits in [12] having fractal properties. The issues discussed in this paper are relevant to a large group of researchers studying bifurcations in nonlinear dynamical systems, nonlinear circuit analysts, signal processing engineers working in chaotic cryptography and secure transmission of signals, and computer programmers. A variety of the techniques presented here make the paper distinctive relative to the previously published works. In one of the examples studied in this paper by technique IV (electric arc circuit) we emphasize the importance of the selection of special parameter T for the 0-1 test for chaos in relation to the step size dt used in the Runge-Kutta algorithm of order 4. The choice of T for the assumed dt should be done by analyzing the maximum frequency of the analyzed signal, as discussed in [18]. A link to a publicly available Julia code used to create some of our two-parameter diagrams in this paper, is given in Section III. The code can be used to recreate some of the figures (bifurcation diagrams) presented in this paper. Also, it is worth mentioning that when looking and analyzing two-parameter bifurcation diagrams it is often beneficial to supplement such diagrams with one-parameter ones, as the later diagrams contain slightly different details. They can be obtained through various software packages, among which XPPAUT and AUTO/Matlab combination are perhaps the most widely used [27], [28].

We would like to emphasize that the topic of this paper is different from the bifurcations of the first-order distributed-delay systems [29] and planar cases of limit cycle bifurcations [30], which are both chaos free problems.

Finally, although in our illustrative examples we used rather simple ODE systems with one or two nonlinear terms, the obtained two-parameter diagrams are quite complicated. Even more complicated two-parameter diagrams from the 0-1 test could be expected for the recently formulatted interesting chaotic systems with infinite number of coexisting attractors [31] and the 4D chaotic systems used in image encryption applications [32].

II. COMPUTATIONAL ALGORITHMS AND SOFTWARE USED

Two relatively simple systems of nonlinear ODEs in the Appendix are used as test models. Using the dimensionless parameter models (see Buckingham's π theorem [33]) one can show that the first system, the active memristive circuit presented in Appendix A, has three parameters a, b and c [9]. The second system presented in Appendix B has four parameters R, L, C and m [26]. Suppose that any two parameters, say a and b in (2), vary simultaneously with certain steps within their minimum and maximum bounds: $a_{min} \leq a_i \leq a_{max}$ and $b_{min} \leq b_j \leq b_{max}$. To examine the impact of the two selected parameters on the response of the model (2) we consider each of the several hundred thousand (or a few million) points (a_i, b_j), with $c = const$, solve (2), identify the type of response (periodic, chaotic, unstable or asymptotically stable). Within the perodic type of response we identify the number n in the period-n response. To each period-n response, with $n = 1, 2, \ldots, n_{max}$, we assign a specific color. Then a two-parameter color diagram is created. We used $n_{max} = 16$ and $n_{max} = 64$ for systems (2) and (4), respectively. All our bifurcation diagrams are of size 600×600, that is $1 \leq i \leq 600$, $1 \leq j \leq 600$. Notice that neither system (2) nor (4) is stiff. The following three techniques (with explicit numerical solvers) have been used in our computations and analysis of (2).

I: Classical Runge-Kutta order 4 algorithm [34] (or RK-4 for short) with C++ language.

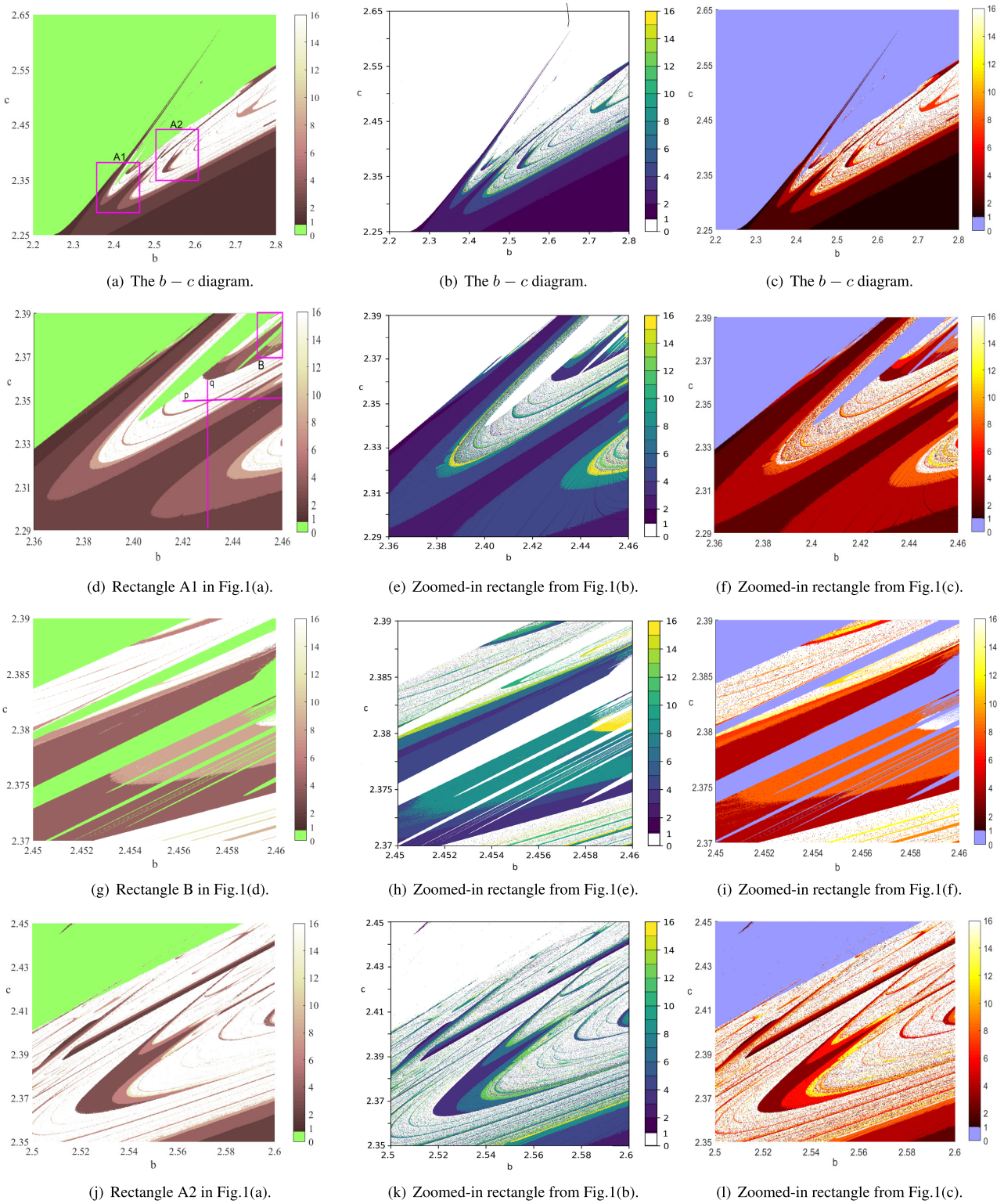
II: Runge-Kutta order 5 Dormand/Prince method [35] (RK-5 for short; equivalent to ode45 in Matlab) with Fortran language and Python for graphics purposes.

III: RK-5 algorithm (the same as in technique II) with Julia language [36].

In addition, to analyze (4) we used both the first technique given above, and

IV: the 0-1 test for chaos applied to the responses obtained with the RK-5 algorithm and C++ language.

In the first technique above the identification of the local maximum points for one of the three variables in (2) was done (we chose the y variable) in a certain time interval $[t_{min}, t_{max}]$ using a simple algorithm based on the three consecutive discrete values, say $y(t_{k-1})$, $y(t_k)$ and $y(t_{k+1})$. The identification was done backwards from t_{max} to t_{min}. The step size dt in the RK-4 solver was selected such that $dt << T$, where T is the period of oscillations. The value of $dt = 0.002$ was proved to be small enough for the intervals of parameters a, b and c used in this paper. Once, the maximum points are identified for $t_{min} \leq t \leq t_{max}$, then, if possible, the type of periodic response is established as period-n oscillations,



(a) The $b - c$ diagram. (b) The $b - c$ diagram. (c) The $b - c$ diagram.

(d) Rectangle A1 in Fig.1(a). (e) Zoomed-in rectangle from Fig.1(b). (f) Zoomed-in rectangle from Fig.1(c).

(g) Rectangle B in Fig.1(d). (h) Zoomed-in rectangle from Fig.1(e). (i) Zoomed-in rectangle from Fig.1(f).

(j) Rectangle A2 in Fig.1(a). (k) Zoomed-in rectangle from Fig.1(b). (l) Zoomed-in rectangle from Fig.1(c).

FIGURE 1. Two-parameter bifurcation diagrams obtained with the techniques I (left column [20]), II (middle column) and III (right column), time step 0.002, $a = 1.655$, $[x(0), y(0), z(0)] = [0.1, 0.0, 0.1]$ and 600 × 600 discrete points. Unstable solutions exist in the green, white and blue areas for the left, middle and right columns, respectively.

$n = 1, 2, \ldots 16$. All responses with $n \geq 16$ are grouped together and given the status of period–16. This group also includes chaotic responses. Finally, if a response is unstable for certain parameters, then the areas with such parameters are marked by the green color (see the four diagrams in the left column in Fig. 1 [20].

In the second and third techniques (RK-5 with Fortran/ Python and RK-5 with Julia, respectively) let assume that (2) has $u \equiv [x, y, z] \in \mathbb{R}^3$ and $p = [a, b, c]$ and consider the parameter-dependent initial value problem $u' = f(u, p)$, $u(0) = u_0$. The classification of the response for a given choice of the parameter vector p (with two varying and one fixed parameters) was based on the maximum values in the y-component of $u(t)$ (the same as in technique I). For this, we needed to compute the roots of $y'(t) = 0$, that is the time instants where $y'(t)$ changes its sign. Assume that the system is integrated numerically by a one–step method $u_{n+1} = \varphi(u_n, h)$ (the explicit Runge–Kutta method of order 5 by Dormand/Prince [34]). In an interval $[t_n, t_{n+1}]$, if $y'_n > 0$ and $y'_{n+1} \le 0$, linear interpolation yields

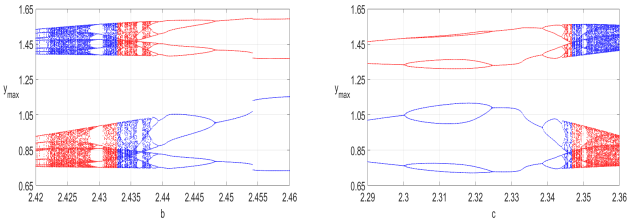
$$h_E = h\frac{y'_n}{y'_n - y'_{n+1}}, \quad t_E = t_n + h_E, \; u_E = \phi(u_n, h_E). \tag{1}$$

Since we chose to analyze variable $y(t)$ in u, the mapping $(h, u_{n+1}) \mapsto (h_E, u_E)$ defined by (1) is iterated until y'_E is zero or close to it (which is typically achieved with $|y'_E| < 10^{-8}$ within two iteration steps). The event (t_E, u_E) is saved in a queue to be used later to detect cycles. An approximate period $T = t_{\widehat{E}} - t_E$ is found when another event $(t_{\widehat{E}}, u_{\widehat{E}})$ with $\|u_E - u_{\widehat{E}}\| < \varepsilon$ is detected. The method has been implemented in Fortran and Python (technique II) and in Julia (technique III). All computations were performed using a PC with an Intel(R) E5-2643 CPU running Debian 10.

Since the overall performance will be dominated by the cost of time stepping, consider as a micro-benchmark problem the CPU times for taking 10^7 steps $u \mapsto u + hf(u)$ of the explicit Euler method with $h = 10^{-4}$ for system (2). This task takes 0.062 sec in Fortran 90 (gfortran), 1.2 sec in Matlab, 40 sec in Python and 330 sec in Octave. In Julia, using the built-in vector type `Array{Float64,1}`, it takes disappointing 1.8 sec. The cost is dominated by memory allocations. However, this can be overcome, using `StaticArray.jl` and `@SVector` for u – which is a one-line change – reduces the running time down to 0.07 sec.

III. TWO-PARAMETER BIFURCATION DIAGRAMS

Fig. 1 shows four two-parameter bifurcation diagrams obtained by each of the first three techniques described in the previous section (twelve diagrams total). Three different colorbars were chosen so the readers can select their preferred one. Note that the diagram shown at **https://sim.mathematik.uni-halle.de/helmut/2019/bif** was created by the code available there. With a slight modification of that code one can also create all four diagrams shown in the third column in Fig. 1. Notice that the corresponding diagrams in the middle and right columns are practically the same (apart for the different colors used to denote the same type of response). Those eight diagrams were computed with the same RK-5 algorithm (techniques II and III described above). However, the speed of computations of the four diagrams obtained from Fortran (middle column) and the speed of computations of the four diagrams obtained from Julia (right column) are different, as illustrated by the micro-benchmark problem in Section II. Besides, the diagrams in the middle column were obtained by a combination of Fortran (computations) and Python (graphics), while the diagrams in the right column were done (both computations and graphics) by using Julia only. Also, notice the difference between the diagrams obtained by the RK-4 algorithm (left column diagrams in Fig. 1 and the diagrams obtained by using the RK-5 algorithm (middle and right columns). This is expected, as the RK-5 algorithm has better properties (error control and accuracy) than the RK-4 algorithm. Another reason for the difference, as described in the previous section, is the difference in identifying maximum values of $y(t)$ in techniques I and II (and III).



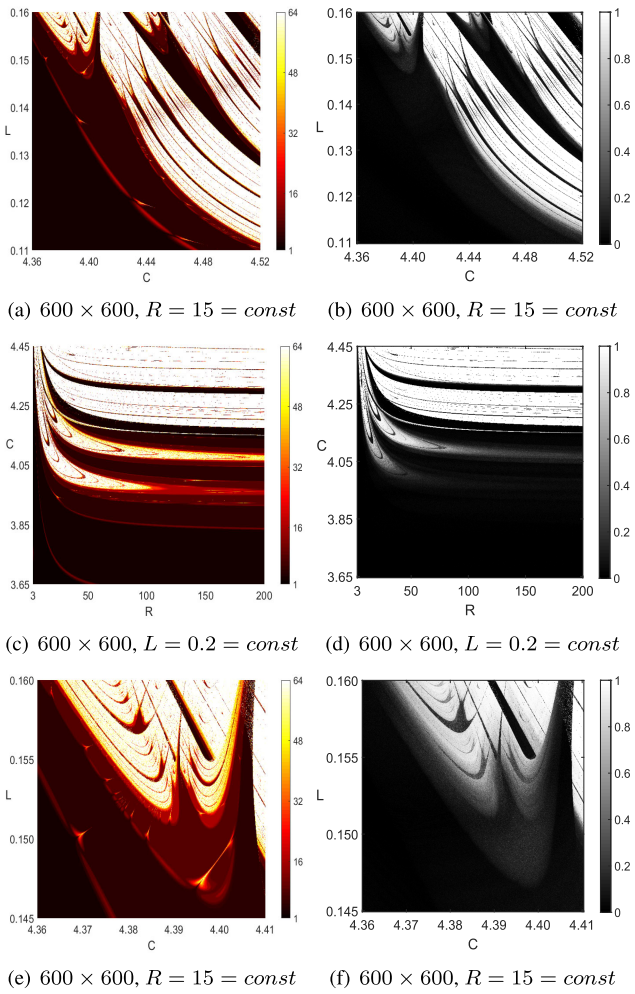
(a) Diagram along line p in Fig. 1(d) for (2) with $c = 2.35$ and $a = 1.655$.

(b) Diagram along line q in Fig. 1(d) for (2) with $b = 2.43$ and $a = 1.655$.

FIGURE 2. Bifurcation diagrams of y with the initial condition [0.1, 0, 0.1] (blue) and [−0.1, 0, −0.1] (red). Based on solutions with RK-4 for $0 \le t \le 1000$ with $dt = 0.01$.

A quick verification of two-parameter diagrams is shown in Fig. 2 which presents one-parameter diagrams computed along the lines p and q as marked in Fig.1(d). The agreement between the two- and one-parameter diagrams is excellent as, for the same pairs of parameters, both diagrams show the same period-n or chaotic oscillations. Additionally, the one-parameter diagrams show the maximum values of the analyzed signal in each type of period-n oscillations.

Next, notice the one-to-one correspondence of the diagrams shown in Fig. 3. Gray diagrams in the right column were computed by solving (4) 360,000 times (by RK-5) for various pairs of parameters R, L and C and to each of those solutions the 0-1 test for chaos was applied (technique IV). We took care of the typical oversampling issue associated with the 0-1 test for chaos (when applied to continuous nonlinear systems) by using the FFT to select the T parameter (see [18]) guaranteeing the proper outcome of the 0-1 test and preventing the oversampling phenomenon in the 0-1 test. When the underlying mathematical model (4) is solved numerically and the time series is available at certain $n \cdot dt$ instants, then to feed the time series values into the 0-1 test, one needs to use the solution values at every T instant, where the positive integer T depends, in general, on the system being analyzed, the value of dt and maximum frequency of the analyzed chaotic signal. Thus, the time series at the instants $n \cdot T \cdot dt$ is used in the 0-1 test. Examples reported in [18], [26] show that the values of T could range from a single integer value ($T = 7$ in [18]) to thousands ($T = 3000$ in [26]).



(a) 600×600, $R = 15 = const$ (b) 600×600, $R = 15 = const$

(c) 600×600, $L = 0.2 = const$ (d) 600×600, $L = 0.2 = const$

(e) 600×600, $R = 15 = const$ (f) 600×600, $R = 15 = const$

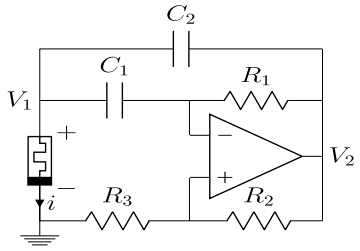
FIGURE 3. Bifurcation diagrams obtained via technique I (left column) and technique IV (right column; 256 gray levels used), $[x(0), y(0), z(0)] = [0.5, 4.0, 1.0]$. The T in the 0-1 test was equal 1700.

The danger is, that the oversampling phenomenon in the 0-1 test, when not properly addressed (incorrect value of T chosen), may give false results of the 0-1 test. The graybar on the right hand side of the diagrams in the right column of Fig. 3 corresponds to the K values obtained from the 0-1 test for chaos ($K \approx 0$ for periodic responses and $K \approx 1$ for chaotic ones).

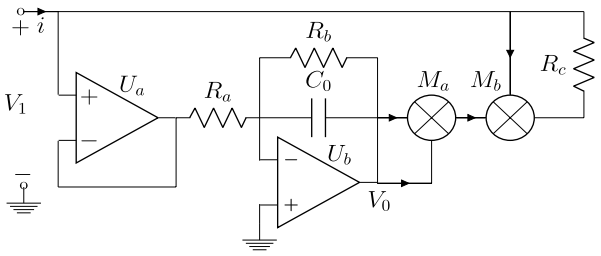
Finally, even if we underline a need to use parallel computations to obtain some of the two-parameter diagrams (mainly those from the 0-1 test), since our systems in the Appendix have only three variables, our approach is different from the one presented in [37], in which systems with thousands of variables are considered. We can only parallelize the independent integrations for different set of parameters. There is no communication needed and the speed-up is only limited by our setup to create the jobs and to collect the results.

IV. CONCLUSION

Two-parameter bifurcation diagrams for nonlinear dynamical systems have been computed by using various numerical techniques in different programming environments (RK-algrorithms, C++, Fortran/Python, Julia, 0-1 test for chaos). Some computations can be done on typical laptops, while others (i.e. the 0-1 test for chaos) require more advanced (possibly parallel) computations [26]. The running time for the adaptive code varies for different set of parameters. A typical run for the 600×600 graph of the Julia code with a single thread on Intel Xeon E5-2643 takes about 4 minutes and consumes 250 MBytes. The computation itself takes about the same time with the Fortran 90 code. It is a matter of personal and subjective preference to use either of the two approaches (methods). The authors prefer to use the Julia code, although the Julia environment is not as mature at the present time as the Fortran/Python environment is. Computing two-parameter bifurcation diagrams with any of the techniques presented in this paper for other chaotic systems, e.g. those reported in [38], seems quite straightforward. Future research should focus on three- and multi-parameter bifurcations diagrams. Such a task looks quite challenging from the points of view of graphical presentation, accessing diagrams on flat computer screen surfaces and storing data.



(a) Active oscillating circuit with memristor.



(b) Memristor emulator with two op-amps and two multipliers.

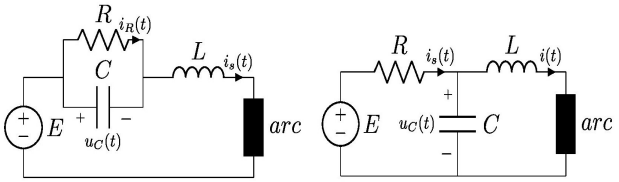
FIGURE 4. An oscillatory active circuit with memristor.

APPENDIX A
OSCILLATORY MEMRISTIVE CIRCUIT

Consider the memristive circuit shown in Fig.4(a) for which the memristive element is emulated as shown in Fig.4(b), and the overall dimensionless model is

$$x' = -x + y, \quad y' = -ay + bz - x^2 y, \; z' = c(z - y) \qquad (2)$$

with the properties (equilibria, symmetry, hysteretic characteristic of memristor, etc.) described in details in [20]. The x, y and z variables are all linear combinations of the signals V_0, V_1 and V_2 (see Fig.4), and the three parameters in (2) depend



(a) Circuit A (b) Circuit B

FIGURE 5. Electric arc circuits. Circuit B is also described by (4) with a suitable change of variables.

of the resistors, capacitors and the combined scaling factor of the multipliers M_a and M_b. The emulator in Fig. 4(b) is described by

$$\begin{aligned} i &= (1 - gV_0^2)V_1/R_c \\ V_0' &= -V_0/(R_bC_0) - V_1/(R_aC_0) \end{aligned} \tag{3}$$

and it satisfies the three basic fingerprints of memristors, see [9], [20].

APPENDIX B
ELECTRIC ARC CIRCUITS

The electric arc circuit in Fig.5(a) is described by the system of three equations on the left side of (4) below and its dimensionless version on the right side [39]

$$\begin{array}{l} \frac{di}{d\tau} = \frac{1}{L}(u_C - \frac{U(i_\theta)}{i_\theta} i) \\ \frac{du_C}{d\tau} = \frac{1}{RC}(E - u_C - Ri) \\ \frac{di_\theta^2}{d\tau} = \frac{1}{\theta}(i^2 - i_\theta^2) \end{array} \rightarrow \begin{array}{l} \frac{dx}{dt} = \frac{1}{L}(y - xz^m) \\ \frac{dy}{dt} = \frac{1}{RC}(R+1-y-Rx) \\ \frac{dz}{dt} = x^2 - z \end{array} \tag{4}$$

where $x = i/I_0$, $y = u_C/U_0$, $z = i_\theta^2/I_0^2$, and i_θ, i, u_C are the arc current, current through L and voltage across C, respectively. The U_0 and I_0 are constants from the static arc voltage-current characteristic $U(i_\theta) = U_0(i_\theta/I_0)^n$ with $n < 0$. It is known (see [26]) that the ODE system on the right hand side of (4) has two equilibrium points: (1, 1, 1) and (x_a, x_a^n, x_a^2), where x_a is the solution of $1 + R - Rx_a - x_a^n = 0$. The second equilibrium is unstable provided that $R + n > 0$, which is assumed to hold true. Further analytical properties of (4) are given in [39].

ACKNOWLEDGMENT

The authors would like to thank four annonymous reviewers for their helpful and constructive comments.

REFERENCES

[1] R. E. Dolmetsch, K. Xu, and R. S. Lewis, "Calcium oscillations increase the efficiency and specificity of gene expression," *Nature*, vol. 392, pp. 933–936, Apr. 1998.
[2] L. H. A. Monteiro, D. N. F. Filho, and J. R. C. Piqueira, "Bifurcation analysis for third-order phase-locked loops," *IEEE Signal Process. Lett.*, vol. 11, no. 5, pp. 494–496, May 2004.
[3] Q. Xu, Y. Lin, B. Bao, and M. Chen, "Multiple attractors in a non-ideal active voltage-controlled memristor based Chua's circuit," *Chaos, Solitons Fractals*, vol. 83, pp. 186–200, Feb. 2016.
[4] B. Bao, N. Wang, Q. Xu, H. Wu, and Y. Hu, "A simple third-order memristive band pass filter chaotic circuit," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 64, no. 8, pp. 977–981, Aug. 2017.
[5] Z. T. Njitacke, J. Kengne, and L. K. Kengne, "Antimonotonicity, chaos and multiple coexisting attractors in a simple hybrid diode-based jerk circuit," *Chaos, Solitons Fractals*, vol. 105, pp. 77–91, Dec. 2017.
[6] G. Li, L. Zeng, L. Zhang, and Q. M. J. Wu, "State identification of Duffing oscillator based on extreme learning machine," *IEEE Signal Process. Lett.*, vol. 25, no. 1, pp. 25–29, Jan. 2018.
[7] B. C. Bao, P. Y. Wu, H. Bao, Q. Xu, and M. Chen, "Numerical and experimental confirmations of quasi-periodic behavior and chaotic bursting in third-order autonomous memristive oscillator," *Chaos, Solitons Fractals*, vol. 106, pp. 101–170, Jan. 2018.
[8] X. Li, S. Zhang, X. Liu, X. Wang, A. Zhou, and P. Liu, "Dynamic analysis on the calcium oscillation model considering the influences of mitochondria," *Biosystem*, vol. 163, pp. 36–46, Jan. 2018.
[9] Z. Galias, "Study of amplitude control and dynamical behaviors of a memristive band pass filter circuit," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 5, pp. 637–641, May 2018.
[10] E. Voit, *A First Course in Systems Biology*, 2nd ed. New York, NY, USA: Garland Sci., 2017.
[11] W. Szumiński, "Integrability analysis of chaotic and hyperchaotic finance systems," *Nonlinear Dyn.*, vol. 94, pp. 443–459, Oct. 2018.
[12] W. Marszalek and Z. Trzaska, "Mixed-mode oscillations and chaotic solutions of *jerk* (Newtonian) equations," *J. Comput. Appl. Math.*, vol. 262, pp. 373–383, May 2014.
[13] W. Marszalek, *Applied Differential-Algebraic Equations*. Opole, Poland: Opole Univ. of Technology Publishing Office, 2018.
[14] M. Henon, "On the numerical computation of Poincaré maps," *Phys. D, Nonlinear Phenomena*, vol. 5, pp. 412–414, Sep. 1982.
[15] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," *Chaos*, vol. 27, Nov. 2017, Art. no. 121102.
[16] Y. Zou, R. V. Donner, J. F. Donges, N. Marwan, and J. Kurths, "Identifying complex periodic windows in continuous-time dynamical systems using recurrence-based methods," *Chaos*, vol. 20, Nov. 2010, Art. no. 043130.
[17] Z. Lu, B. R. Hunt, and E. Ott, "Attractor reconstruction by machine learning," *Chaos*, vol. 28, Jun. 2018, Art. no. 061104.
[18] W. Marszalek and M. Melosik, "On the 0/1 test for chaos in continuous systems," *Bull. Polish Acad. Sci., Tech. Sci.*, vol. 63, no. 3, pp. 521–528, 2016.
[19] W. Marszalek and H. Podhaisky, "2D bifurcations and newtonian properties of memristive Chua's circuits," *Europhys. Lett.*, vol. 113, no. 1, 2016, Art. no. 10005.
[20] W. Marszalek and J. Sadecki, "Complex two-parameter bifurcation diagrams of a simple oscillating circuit," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 4, pp. 687–691, Apr. 2019.
[21] L. Junges and J. A. C. Gallas, "Intricate routes to chaos in the Mackey–Glass delayed feedback system," *Phys. Lett. A*, vol. 376, pp. 2109–2116, Jun. 2012.
[22] G. A. Gottwald and I. Melbourne, "On the implementation of the 0–1 test for chaos," *SIAM J. Appl. Dyn. Syst.*, vol. 8, no. 1, pp. 129–145, 2009.
[23] G. A. Gottwald and I. Melbourne, "On the validity of the 0–1 test for chaos," *Nonlinearity*, vol. 22, no. 6, pp. 1367–1382, 2009.
[24] G. Litak, A. Syta, M. Budhraja, and I. M. Saha, "Detection of the chaotic behaviour of a bouncing ball by the 0–1 test," *Chaos, Solitons Fractals*, vol. 42, pp. 1511–1517, Nov. 2009.
[25] G. Litak, D. Bernardini, A. Syta, G. Rega, and A. Rysak, "Analysis of chaotic non-isothermal solutions of thermomechanical shape memory oscillators," *Eur. Phys. J. Special Topics*, vol. 222, pp. 1637–1647, Sep. 2013.
[26] W. Marszalek and J. Sadecki, "Parallel computing of 2-D bifurcation diagrams in circuits with electric arcs," *IEEE Trans. Plasma Sci.*, vol. 47, no. 1, pp. 706–713, Jan. 2019.
[27] B. Ermentrout, *Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students*. Philadelphia, PA, USA: SIAM, 2002.
[28] E. Coetzee, B. Krauskopf, and M. Lowenberg, "The dynamical systems toolbox: Integrating AUTO into MATLAB," in *Proc. 16th US Nat. Congr. Theor. Appl. Mech.*, State College, PA, USA, 2010, Paper USNCTAM2010-827.
[29] Y. Cao, "Bifurcations in an Internet congestion control system with distributed delay," *Appl. Math. Comput.*, vol. 347, pp. 54–63, Apr. 2019.
[30] T. Chen, L. Huang, P. Yuc, and W. Huang, "Bifurcation of limit cycles at infinity in piecewise polynomial systems," *Nonlinear Anal., Real World Appl.*, vol. 41, pp. 82–106, Jun. 2018.

[31] Q. Lai, C. Chen, X.-W. Zhao, J. Kengne, and C. Volos, "Constructing chaotic system with multiple coexisting attractors," *IEEE Access*, vol. 7, pp. 24051–24056, 2019.
[32] Q. Lai, B. Norouzi, and F. Liu, "Dynamic analysis, circuit realization, control design and image encryption application of an extended Lü system with coexisting attractors," *Chaos, Solitons Fractals*, vol. 114, pp. 230–245, Sep. 2018.
[33] J. C. Gibbings, *Dimensional Analysis*. London, U.K.: Springer, 2011.
[34] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. New York, NY, USA: Wiley, 2008.
[35] J. R. Dormand, *Numerical Methods for Differential Equations: A Computational Approach*. Boca Raton, FL, USA: CRC Press, 1996, pp. 82–84.
[36] *The Julia Programming Language*. Accessed: Jun. 28, 2019. [Online]. Available: https://julialang.org/
[37] A. Al-Omari, J. Arnold, T. Taha, and H.-B. Schüttler, "Solving large nonlinear systems of first-order ordinary differential equations with hierarchical structure using multi-GPGPUs and an adaptive Runge Kutta ODE solver," *IEEE Access*, vol. 1, pp. 770–777, 2013.
[38] G.-H. Xu, Y. Shekofteh, A. Akgül, C. Li, and S. Panahi "A new chaotic system with a self-excited attractor: Entropy measurement, signal encryption, and parameter estimation," *Entropy*, vol. 20, no. 2, p. 86, 2018.
[39] I. V. Pentegov and V. N. Sydorets, "Comparative analysis of models of dynamic welding arc," *Paton Welding J.*, vol. 12, pp. 45–48, Dec. 2015.

WIESLAW MARSZALEK received the Ph.D. and D.Sc. degrees in electrical engineering from the Warsaw University of Technology, Poland, and the Ph.D. degree in applied mathematics from North Carolina State University, Raleigh, NC, USA. He was a Humboldt Research Fellow in Bochum, from 1990 to 1992, Hamburg in 1996, and Halle in 2011, all in Germany, and a Fulbright Research Fellow in Warsaw, from 2005 to 2006, and Opole in 2018, both in Poland. He currently teaches computer science at the Opole University of Technology, Poland, and mathematics at Rutgers University, New Brunswick, NJ, USA. He has published over 100 journal and conference papers in the areas of 2D discrete equations, differential-algebraic equations, nonlinear circuits and electric arcs, memristors, and singularity induced bifurcations and chaos.

HELMUT PODHAISKY received the Ph.D. degree in mathematics from Martin-Luther-Universität, Halle-Wittenberg, Germany, where he is currently with the Institute of Mathematics. He is an author of numerous research journal and conference papers in the areas of numerical analysis, solving differential equations, and scientific programming. He serves as a Co-chair of the NUMDIFF conferences and an Associate Editor of the journal *Applied Numerical Mathematics*.

JAN SADECKI received the Ph.D. and D.Sc. degrees in automatic control and robotics from the Warsaw University of Technology, Poland, in 1988 and 2004, respectively. He is currently with the Institute of Computer Science, Department of Electrical Engineering, Opole University of Technology, Opole, Poland. From1992 to 1993, he spent a sabbatical year at the Centre for Mathematical Software Research, University of Liverpool, U.K. From 2012 to 2016, he was an Associate Chair of the Department of Electrical Engineering, Opole University of Technology. Since 2017, he has been a Director of the Institute of Computer Science, Opole University of Technology. His research and teaching interests includes parallel computing, distributed computing networks, and optimization.