ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 303-308

Simulation of Arrays of Chua Circuits by Composition Methods

Yves Moreau and Joos Vandewalle

Department of Electrical Engineering
Katholieke Universiteit Leuven
Kardinaal Mercierlaan 94
B-3001 Heverlee, Belgium

Abstract.
Composition methods are methods arising from differential geometry for the integration of ordinary differential equations. We apply them here to arrays of Chua circuits. In these methods, we split the vector field of the array of Chua circuits into its linear part and its nonlinear part. We then solve the elementary differential equation for each part separately, and recombine these contributions in a sequence of compositions. This gives rise to simple integration rules for arrays of Chua circuits, which we compare to more classical approaches: the fixed time-step explicit Euler and adaptive fourth-order Runge-Kutta methods.

1. Introduction

The standard methods for the time-integration of ordinary differential equations (ODEs) are schemes such as the Euler method or the Runge-Kutta method. These schemes are straightforwardly applied to the integration of continuous-time recurrent neural networks, in particular Cellular Neural Networks consisting of arrays of Chua circuits. However, physicists have recently introduced a completely different class of methods for the solution of ordinary differential equations: the composition methods [2, 3]. The spirit of these methods is that if the vector field of the differential equation is the superposition of elementary vector fields, we can approximate its solution by composing the flows of the elementary contributions. Composition methods are particularly useful for numerically integrating ODEs when the equations have some simple structure. The authors have recently introduced them in the field of neural networks [4].

We will first present the elements of Lie algebra theory necessary for the exposition of composition methods. We will then show how to use this theory for the integration of ordinary differential equations. We will apply these methods to an array of coupled Chua circuits and we will compare the performance of these methods with those of standard methods: fixed time-step explicit Euler and adaptive fourth-order Runge-Kutta.

2. Arrays of Chua Circuits

To show the potential relevance of composition methods to the field of neural networks, we present their application to the simulation of Cellular Neural Networks (CNN) of the type arrays of Chua Circuits. These neural networks are effective for parallel signal processing and real-time simulation of nonlinear spatio-temporal phenomena.

The Chua circuit is a nonlinear electrical circuit. We can write it as a state-space model or ordinary differential equation of the form

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{bmatrix} -a & a & 0 \\ b & -b & 1 \\ 0 & -c & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} g(x) \\ 0 \\ 0 \end{pmatrix}$$

with $g(x)$ a piecewise linear function: $g(x) = -(a/2)[(s_1+s_2)x + (s_0 - s_1)(|x - \beta_1| - |\beta_1|) + (s_2 - s_0)(|x - \beta_2| - |\beta_2|)]$.

Moreover, we can consider the more general systems of Cellular Neural Networks. For example, we can have a CNN formed by an array of Chua circuits, where the neighbors are coupled by linear coupling. If we have n circuits having each three state variables, the equations describing the evolution of the system are

$$\begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{pmatrix} = \begin{bmatrix} -a & a & 0 \\ b & -b & 1 \\ 0 & -c & 0 \end{bmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \begin{pmatrix} g(x_i) \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \rho.\Sigma_{j=1}^{n} \xi_{ij} x_j \\ 0 \\ 0 \end{pmatrix}$$

with $i = 1, \ldots n$ and where the coupling term ξ_{ij} is equal to one if circuit i is connected to circuit j, and equal to zero else. What we can remark is that, however large the number of circuits is, the system is made of a large linear part (the first and third terms) and a nonlinear part that consist of n identical decoupled piecewise linear equations.

3. Lie algebra theory

The Cellular Neural Networks under consideration here are described by a system of ordinary differential equations $\dot{x}(t) = A(x(t))$. We write the ODE as $\dot{x}(t) = A(x(t))$ to follow the conventions of Lie algebra theory. We can write its solution, as a function of the initial condition x_0, in two forms: as a flow $x(t) = \Phi(x_0, t)$ (as is standard in the dynamical system literature), or as an exponential solution $x(t) = e^{tA}x_0$ (as in the Lie algebra literature [1]). The latter notation should read: "$x(t)$ is the image after time t of the initial condition x_0 under the flow of $\dot{x} = A(x)$".

Lie algebra theory is an important tool in physics [1] and an essential part of nonlinear system theory. We need to define a new operation: the Lie bracket of two vector fields, which is again a vector field. In the particular case of vector fields on $\mathbb{R}^n$, we can express the bracket as

$$[A, B] = B\nabla A - A\nabla B$$

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 303-308

The main mathematical tool we need is the Baker-Campbell-Hausdorff (BCH). This formula gives an expansion for the product of two exponentials of elements of the Lie algebra [1]:

$$e^{tA}e^{tB} = e^{t(A+B) + \frac{t^2}{2}[A,B] + \frac{t^3}{12}([A,[A,B]] + [B,[B,A]]) + \cdots}. \quad (1)$$

3.1. Integration of ordinary differential equations by compositions

We now look at how to solve ordinary differential equations using compositions. We refer the reader to [4] for more details. Suppose we want to solve the ODE $\dot{x}(t) = X(x(t))$ for a time-step of Δt. The problem becomes that of building an approximation to $e^{\Delta t X}$ as we have that $x(t) = e^{\Delta t X} x_0$. Suppose, in a first step, that the vector field X is the sum of two vector fields: $X = A + B$, where you can integrate A and B analytically or much more easily than X. Then we can use the BCH formula to produce a first-order approximation to the exponential map:

$$\textbf{BCH:} \quad e^{t.X} = e^{t.A}.e^{t.B} + o(t^2). \quad (2)$$

The relation of first-order approximation (2) between the solution of A and B, and the solution of X is the essence of the method since it shows that we can approximate the mapping arising from the solution of an ODE by composing simpler maps (Fig.1). By using the BCH formula repeatedly, we can show that

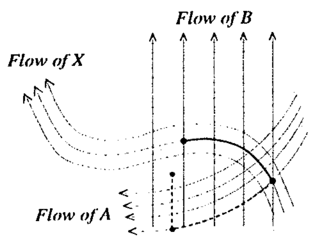

Figure 1: $e^{t.X}(x_0) \approx e^{t.A}.e^{t.B}(x_0)$.

the following symmetric leapfrog scheme is second order:

$$\textbf{Leapfrog:} \quad \begin{aligned} e^{t.X} &= e^{\frac{t}{2}.A}.e^{t.B}.e^{\frac{t}{2}.A} + o(t^3), \\ &= S(t) + o(t^3). \end{aligned}$$

4. Composition integrators

Higher-order integrators are possible as we can see in Table 1. The error associated to each method is the effective error constant defined in [3]. We

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 303-308

Order	Substeps	Error	Formula
2	3	0.070	$S(\Delta t) = e^{(\Delta t/2)A} e^{\Delta t B} e^{(\Delta t/2)A}$
2	5	0.026	$e^{a_1 \Delta t A} e^{b_1 \Delta t B} e^{a_2 \Delta t A} e^{b_1 \Delta t B} e^{a_1 \Delta t A}$ $a_1 = 0.1932,\ b_1 = 0.5,\ a_2 = 1 - 2a_1$
4	7	0.098	$S(w_1 \Delta t) S(w_2 \Delta t) S(w_1 \Delta t)$ $w_1 = 1.3512,\ w_2 = 1 - 2w_1$
4	11	0.033	$S(w_1 \Delta t) S(w_2 \Delta t) S(w_3 \Delta t) S(w_2 \Delta t) S(w_1 \Delta t)$ $w_1 = 0.28,\ w_2 = 0.6254,\ w_3 = 1 - 2w_1 - 2w_2$
4	11	0.0046	$e^{a_1 \Delta t A} e^{b_1 \Delta t B} e^{a_2 \Delta t A} e^{b_2 \Delta t B} e^{a_3 \Delta t A} e^{b_3 \Delta t B}$ $\times e^{a_3 \Delta t A} e^{b_2 \Delta t B} e^{a_2 \Delta t A} e^{b_1 \Delta t B} e^{a_1 \Delta t A}$ $a_1 = 0.0893,\ b_1 = 0.4,$ $a_2 = -0.0973,\ b_2 = -0.1,$ $a_3 = 1/2 - a_1 - a_2,\ b_3 = 1 - 2b_1 - 2b_2$

Table 1: Formulas for the composition methods

present two types of methods: the symmetric methods and the symmetric methods composed of symmetric steps. The symmetric methods are presented directly as a product of exponentials, while the symmetric methods composed of symmetric steps are presented as a product of symmetric leapfrog steps $S(\Delta t)$. The leapfrog is the simplest method to implement, while the fourth-order symmetric method (error=0.0046) is the most efficient. A remark of fundamental importance is that we can replace the exact solutions of $e^{\Delta t A}$ and $e^{\Delta t B}$ in the symmetric methods by any first-order integrator (e.g., $I + \Delta t A = e^{\Delta t A} + o(\Delta t^2)$) and still guarantee the same order of approximation. Similarly, in the symmetric methods composed of symmetric steps, we can replace the leapfrog $S(\Delta t)$ by any second-order symmetric integrator.

5. Simulations

To show that the composition method is an efficient integration method, we will use it to simulate the behavior of an array of Chua circuits producing a spiral wave. We have compared it to a Runge-Kutta adaptive time-step method and a simple Euler fixed-step method. We have chosen a fourth-order composition method with different time-steps ($\Delta t = 0.01, \Delta t = 0.02, \Delta t = 0.05$), a first-order Euler method with different time-steps ($\Delta t = 0.001, \Delta t = 0.002, \Delta t = 0.005$), and a fourth-order adaptive Runge-Kutta method (where the time-step is automatically adaptively set). In a previous experiment [5], a fixed-step Euler method with time-step $\Delta t = 0.0005$ had been used. The appropriate parameters and initial conditions to generate the spiral wave were the same as in this previous experiment.

We found that the Euler method with time-step $\Delta t = 0.002$ was twice as fast as the adaptive Runge-Kutta method while it had a sufficient accuracy. We show in Figure 2 snapshots of the activity of the x variable on a grid of

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 303-308

50×50 Chua circuits at times $t_1 = 20, t_2 = 60, t_3 = 100$ for the Euler method. If we try to increase the time-step of the Euler method, we see that instabilities

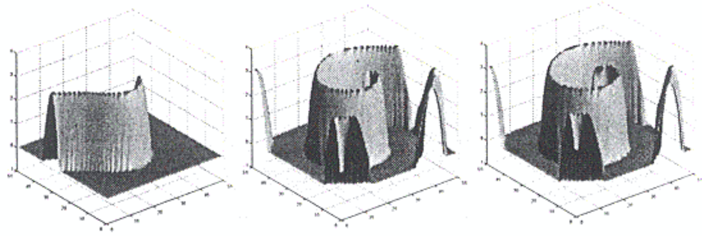

Figure 2: Snapshots of the spiral wave for the Euler method ($t_1 = 20, t_2 = 60, t_2 = 100$).

appear for $\Delta t = 0.005$, which make the integration unreliable as we can see in Figure 3 where we show snapshots of the activity of the x variable at $t = 10$.

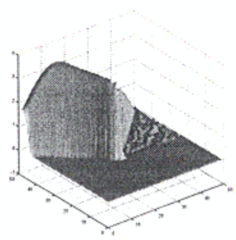

Figure 3: Snapshot for the Euler method with $\Delta t = 0.005$ at time $t = 10$.

In contrast, we can use a much larger time-step for the composition method because of its good conservation properties. This larger time-step comes at the price of a higher number of operations per iteration since the method contains a number of sub-steps. But as a whole, the fourth-order composition method with a time-step of $\Delta t = 0.02$ was twice as fast as the Euler method with a time-step of $\Delta t = 0.001$ for a similar accuracy (Fig. 4). At $\Delta t = 0.05$,

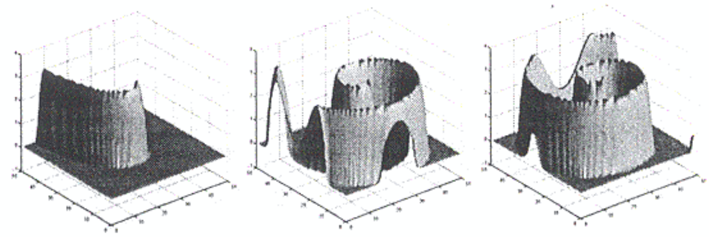

Figure 4: Snapshots of the spiral wave for the composition method ($t_0 = 0, t_1 = 20, t_2 = 40, t_3 = 60, t_4 = 80, t_5 = 100$).

instabilities are too severe to make the integration acceptable (Fig. 5).

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 303-308

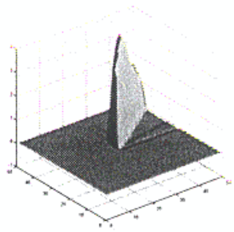

Figure 5: Snapshot for the composition method with $\Delta t = 0.05$ at time $t = 10$.

6. Conclusions

We have presented a new class of integration rules for arrays of Chua circuits: the composition methods. We derived these methods from Lie algebra theory through the use of the Baker-Campbell-Hausdorff formula. Not only do these methods shed new light on the dynamics of arrays of Chua circuits, but they provide enhanced performances for the simulation of these arrays.

Acknowledgments

Joos Vandewalle is a full Professor at the K.U. Leuven. Yves Moreau is a research assistant of the F.W.O. – Vlaanderen. This work is supported by several institutions: 1) The Flemish Government: Concerted Research Action GOA-MIPS, 2) The Belgian State, Prime Minister's Office: Interuniversity Poles of Attraction Programme IUAP P4-02: Modeling, Identification, Simulation and Control of Complex Systems, 3) The FWO Research Community: ICCoS (Identification and Control of Complex Systems) and Advanced Numerical Methods for Mathematical Modeling.

References

[1] V. Arnold. *Mathematical methods of classical mechanics.* Springer-Verlag, New York, NY, 1989.

[2] E. Forest and R. Ruth. Fourth-order symplectic integration. *Physica D*, 43:105–117, 1990.

[3] R. I. McLachlan. On the numerical integration of ordinary differential equations by symmetric composition methods. *SIAM J. Sci. Comput.*, 16(1):151–168, 1995.

[4] Y. Moreau and J. Vandewalle. System modeling using composition networks. In K. Warwick and M. Karny, editors, *Computer Intensive Methods in Control and Signal Processing*, pages 119–140. Birkhauser, Boston, 1997.

[5] A. P. Muñuzuri, V. Pérez-Muñuzuri, G.-G. M., C. L. O., and P.-V. V. Spatio-temporal structures in discretely-coupled arrays of nonlinear circuits: A review. *International Journal of Bifurcation and Chaos*, 1995.