Computing Morse decomposition of ODEs via Runge-Kutta method

Yuki Chiba¹, Tomoyuki Miyaji²* and Toshiyuki Ogawa³

¹Meiji Institute for Advanced Study of Mathematical Sciences, Meiji University, 4-21-1 Nakano, Nakanoku, Tokyo 164-8525, Japan
²Department of Mathematics, Kyoto University, Kitashirakawa Oiwake-cho, Sakyo-ku, Kyoto 606-8502, Japan
³School of Interdisciplinary Mathematical Sciences, Meiji University, 4-21-1 Nakano, Nakanoku, Tokyo 164-8525, Japan

*Corresponding author: miyaji.tomoyuki.4m@kyoto-u.ac.jp

Received April 08, 2021, Accepted May 26, 2021

Abstract

A method of computing combinatorial Morse decomposition for a system of ordinary differential equations is proposed. It uses numerical solutions by Runge-Kutta method, and it is based on an affine approximation and QR decomposition. In contrast to interval arithmetic, it enables us to compute Morse decomposition at lower computational costs sacrificing for mathematical rigor. Numerical examples for time-T map of 3D ODE and a 3D Poincaré map for 4D ODE are presented for comparison between existing and proposed methods.

Keywords Morse decomposition, dynamical systems, ordinary differential equations, Poincaré map, set-oriented computation

Research Activity Group Scientific Computation and Numerical Analysis

1. Introduction

Conley’s fundamental theorem of dynamical systems tells us that any dynamical system can be decomposed into recurrent parts and gradient-like structure connecting each recurrent part. Such a decomposition provides a topological understanding of the global structure of dynamical system. Although it may be practically impossible to obtain the decomposition completely for a mathematical model at hand, Conley’s theorem gives a guiding principle toward understanding of the mathematical model.

There are two approaches toward the analysis of global structure. One approach is to apply path-following algorithm and bifurcation analysis for invariant sets, such as equilibria, limit cycles, etc., and connecting orbits among them. AUTO [1] is a de facto standard software for this purpose. One can obtain invariant sets by the parameter continuation using the Newton method but an isolated branch may be a blind spot. Another approach is to obtain outer approximations of invariant sets. It is carried out by the following steps: (1) divide the phase space into finite grids, (2) construct a digraph, which is called a combinatorial multivalued map, that represents transition among grid elements, (3) find strongly connected components (SCC) in the digraph by graph algorithm. A SCC and connection among them correspond to an outer approximation of a chain recurrent set and gradient-like structure, respectively. This computation is called Morse decomposition. GAIO developed by Dellnitz et al. [2] is an early software that implements such set-oriented computation. Their study was focused on detection of chain recurrent sets rather than gradient-like structure. Arai et al. [3] has proposed a more sophisticated framework by combining interval arithmetic, computational homology, and Conley index. One can grasp global structure of a dynamical system with mathematical rigor but at coarse resolutions.

These two approaches are complementary to each other. Combination of these two approaches will enhance the study of a mathematical model. However, compared to the former approach, the latter one is not widely used. As a reason for that, it is difficult to apply to ordinary differential equations. The grid decomposition of the phase space is easily suffered from dimensionality curse, and interval arithmetic requires higher computational costs than usual floating-point arithmetic. Moreover, due to wrapping effects, the resultant Morse sets can be too large. In order to reduce computational costs, we resort to floating-point arithmetic for computing combinatorial multivalued map. Even if one apply approximate computation, one can obtain a Morse decomposition that captures the phase space structure of the dynamical system.

In this letter, a novel method of computing the combinatorial multivalued map mentioned above based on the Runge-Kutta method is proposed. As it is shown below, the proposed method is capable of capturing the phase space structure at lower computational costs. Although Conley indices are also important in the framework of Arai et al., we focus on the computation of the combinatorial multivalued map, which consumes most of computation time. In particular, we apply the proposed method to 3D ODE and a Poincaré map for 4D ODE in order to make a comparison with the existing method proposed by [2].
2. Methods

For m-dimensional vectors $x = (x_1, \ldots, x_m)$ and $r = (r_1, \ldots, r_m)$, let $B(x, r)$ be defined by

$$B(x, r) = [x_1 - r_1, x_1 + r_1] \times \cdots \times [x_m - r_m, x_m + r_m],$$

where $r_j \geq 0$ for all $j = 1, \ldots, m$.

### 2.1 Morse decomposition

Morse decomposition is briefly introduced in this section. The following explanation is based on [1, §1C]. See [3] for more details of the theory.

Let $X \subset \mathbb{R}^m$ be an m-dimensional interval vector, and let $f$ be a map on $X$. $X$ is a bounding box, in which the dynamics defined by iteration of $f$ are considered. Subdivide $X$ into uniform cubical grids: $X$ is decomposed into $N := 2^d \times \cdots \times 2^m$ interval vectors. We define a digraph $G = (V, E)$ whose vertex set $V = \{1, \ldots, N\}$ is identified with the set of all grid elements $\{X_1, \ldots, X_N\}$ and the edge set $E$ is defined in the following way. For two vertices $a$ and $b$, a directed edge from $a$ to $b$ is assigned if $f(X_a) \cap X_b \neq \emptyset$ for the corresponding grid elements $X_a$ and $X_b$. The next two subsections explain this step in more details.

Next steps is to find SCCs of $G$. The union of interval vectors consisting of a SCC is an outer-approximation of a chain recurrent set of $f$. The condensed graph of $G$ is a directed acyclic graph (DAG), which has vertices corresponding to SCCs of $G$. This condensed graph is called a (combinatorial) Morse graph, and each vertex of the Morse graph is called a (combinatorial) Morse set. The process to obtain a Morse graph is called Morse decomposition.

The following strategy is employed for computational efficiency [3]. Let $d_i$ and $d_f$ be the initial and final depth of subduction. For $d = d_i, d_i + 1, \ldots, d_f$, compute the combinatorial multivalued map and find the invariant part of the digraph. Restrict the phase space to the invariant part, and subdivide it into subcubes. The Morse decomposition is computed at the final subduction depth. This allows us to reduce the total number of cubes.

We implemented the Morse decomposition in the following manner [3]. Let $X$ be a bounding box of the phase space, and let $d_i$ and $d_f$ be initial and final depths of subduction, respectively.

Let $X = \{x\}$. For $j = 1, \ldots, d_i$, repeat the followings:

1. Divide each cell in $X$ in half along $j$th (mod $m$)-th axis.

2. Let $\mathcal{X}$ be the collection of the subdivided cells, and increment $j$.

Once the above procedure is carried out, one has a grid decomposition of the phase space. For $j = d_i, d_i + 1, \ldots, d_f$, repeat the followings:

1. Compute a digraph $G = (V, E)$ corresponding to combinatorial multivalued map on $\mathcal{X}$.

2. a. If $j < d_f$, find invariant part of the digraph $G$, subdivide the cells in the invariant part along $j$th (mod $m$)-th axis, let $\mathcal{X}$ be the collection of the subdivided cells, and increment $j$.

b. If $j = d_f$, find SCCs of $G$ and return the condensed graph as a Morse graph.

### 2.2 Existing methods

In this section, existing methods of computing a combinatorial multivalued map $G = (V, E)$ are explained. Let $X_a = B(c, r)$ be a grid element in $X$, where $c, r \in \mathbb{R}^m$. The question is how to determine grid elements which intersect $f(X_a)$.

Interval arithmetic provides an m-dimensional interval vector $Y_a$ which encloses $f(X_a)$ with mathematical rigor. An edge from $a \in V$ to $b \in V$ is assigned if $Y_a \cap X_b \neq \emptyset$. If $f$ is time-T map or Poincaré map, one can compute it by Lohner’s method [5]. Interval arithmetic approach has been used in [3] and [6].

The method proposed by Dellnitz et al. [2, Appendix A], which is referred to as the existing method in the present letter, is as follows. The basic idea is to collect grid elements which intersect $B(f(x), r)$ for some sample points $x \in X_a$. Let $h = (h_1, \ldots, h_m) \in \mathbb{R}^m$ be a vector of the minimum distance between two grid points and let $L = (L_i)$ be a matrix defined by $L_{ij} = \max_{c \in X_a} \{d \mid f_i(c)\}$. We have

$$|f_i(x) - f_i(c)| \leq \sum_{j=1}^m L_{ij} |x_j - y_j|$$

for any $x = (x_1, \ldots, x_m), y = (y_1, \ldots, y_m) \in X_a$ and all $i = 1, \ldots, m$. An edge from $a$ to $b$ is assigned if there exists $x = (x_1, \ldots, x_m)$ in

$$S = \{x \in X_a \mid (x_i - c_i) \in h_i \mathbb{Z}, i = 1, \ldots, m\},$$

such that $X_b \cap B(f(x), r) \neq \emptyset$.

If $h$ are selected so that $\sum_{i=1}^m L_{ij} h_i \leq r_i$ holds for all $i = 1, \ldots, m$, then the collection of grid elements which intersect $\bigcup_{x \in S} B(f(x), r)$ covers $f(X_a)$ (2, Proposition 22). It is possible to know $L$ for time-T map by the standard theory of ODEs, while it is not so easy to estimate $L$ a priori for Poincaré map.

We applied the existing method in the following way. For a grid element $X_a = B(c, r)$ and an integer $N$, we selected sample points

$$\{(x_{n_1}, \ldots, x_{n_m}) \mid n_j = 0, \ldots, N, j = 1, \ldots, m\},$$

where $x_{n_j} = c_j + r_j + 2n_j r_j/N$ for $j = 1, \ldots, m$. The sample points coincide with those explained above for even $N$.

### 2.3 Proposed method

This section presents the proposed method of computing the digraph $G = (V, E)$. By Taylor’s theorem,

$$f(x) = f(c) + f'(c) (x - c) + O(|x|^2)$$

holds for $x \in X_a$, where $c$ is the center of $X_a$. Therefore, the set $f(X_a)$ can be approximated by

$$f(X_a) \approx f(c) + f'(c) (X_a - c).$$

Once $f(c)$ and $f'(c)$ are obtained, we evaluate upper and lower bounds of the right hand side via floating-point arithmetic to obtain an interval vector $Y$ which approximates an interval enclosure of $f(X_a)$. Next, ap-
ply (approximate) QR decomposition to obtain an orthogonal matrix $Q$ and an upper triangle matrix $R$ satisfying $f'(c) = QR$. Compute interval vectors $Z$ and $W$ such that $(Q^T f(c) + R(X_a - c)) \subset Z$ and $Q^T X_b \subset W$. Assign an edge from $a \in V$ to $b \in V$ if the following two conditions hold:

1. $X_b \cap Y$ is nonempty, and
2. $Z \cap W$ is nonempty.

The proposed method requires the function value $f(c)$ and its Jacobian matrix $f'(c)$. If $f$ is an evolution operator or a Poincaré map for a given ODE $\dot{x} = F(x)$, these can be evaluated by solving

$$
\begin{align*}
\dot{x}(t) &= F(x(t)), \\
\dot{A}(t) &= F'(x(t))A(t),
\end{align*}
$$

where $A(t)$ is an $m \times m$ matrix and $I_m$ is the identity matrix. One can select any ODE solver. In this work, an 8th order Runge-Kutta method called DOP853 [7] is employed.

### 2.4 ODE models

In order to compare the existing and the proposed methods, we compute Morse decomposition for two ODE models. One example is the Lorenz model [8]:

$$
\dot{x} = \sigma(y - x), \quad \dot{y} = x(r - z) - y, \quad \dot{z} = xy - \beta z. \tag{5}
$$

In this work, the parameters are $\rho = 18$, $\sigma = 10$, $\beta = 8/3$, which are different from the classical ones. It is known that there are three equilibria and two unstable limit cycles at these parameter values. We consider the time-$T$ map $\varphi^T$ for the Lorenz system with $T = 0.2$, where $\{\varphi^t\}_{t \in \mathbb{R}}$ is the solution operator defined by the initial value problem of the Lorenz system. Fig. 1 presents a Conley-Morse graph and Morse sets for $\varphi^{0.2}$ on the phase space $X = [-20, 20] \times [-25, 25] \times [-5, 45]$, which are obtained by cmgraphs software developed by Pawel Pilarczyk [9]. Three equilibria and two unstable limit cycles are covered by the Morse sets.

Another example is the following system [10, Ch. 8.6]:

$$
\begin{align*}
\dot{z}_1 &= \left(\frac{1}{2} + i\omega_1\right)z_1 - z_1 \left(|z_1|^2 + \frac{|z_2|^2}{2}\right), \\
\dot{z}_2 &= \left(\frac{1}{2} + i\omega_2\right)z_2 - z_2 \left(|z_2|^2 + \frac{|z_1|^2}{2}\right),
\end{align*}
$$

\begin{equation}
\tag{6}
\end{equation}

where $z_1 = x_1 + iy_1$ and $z_2 = x_2 + iy_2$ for $x_1, x_2, y_1, y_2 \in \mathbb{R}$. Let $P$ be a Poincaré map on $S = \{(x_1, y_1, x_2, y_2) \mid y_2 = 0, \dot{y}_2 > 0\}$. We selected two sets of parameter values: $(\omega_1, \omega_2) = (0.99, 1.01)$ and $(\omega_1, \omega_2) = (0.74, 0.76)$ in both cases, one unstable fixed point and one stable invariant closed circle exist on $S$.

### 3. Results

The existing method with $N = 2, \ldots, 5$ and the proposed methods are applied to time-$T$ map of (5) and Poincaré map for (6) with the above described parameter settings. In both cases, Morse decompositions were computed by the algorithm described in Section 2.1. The initial and final depths were $d_i = 15$ and $d_f = 27$. These two methods are used for computing a digraph $G = (V, E)$ corresponding to combinatorial multivalued map. The computation time for Morse decomposition, the number of Morse sets($\#$ MS), the order of $V$ at $d = d_f$ are measured.

The computer environment was as follows: HP Z440 workstation, Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz, 12cores, Ubuntu 20.04.1, 32GB RAM, GNU Compiler Collection version 9.3.0. The computation was parallelized by OpenMP.

Table 1 presents the results for time-$T$ map of (5). Tables 2 and 3 present the result for Poincaré map of (6). Issues 1–3 appearing in the comment column of the tables are as follows:

**Issue 1** It was not capable of computing the Conley index for the Morse set containing the fixed point.

**Issue 2** The Morse set containing the unstable fixed point was missed.

**Issue 3** The gradient structure connecting two Morse sets was missed.

We would like to mention about comparison between the proposed method and method based on interval arithmetic. For the Lorenz system, the cmgraphs software [9], which cannot parallelize with respect to grid elements, took about 2.2 hours for Morse decomposition (45 Morse sets consisting of 556360 cubes) from depth 5 to 9 ($d_i = 15$ to $d_f = 27$ in our definition). Serial computation by the proposed method took 36.158 sec (23 Morse sets with 506396 cubes). The computation speed is beyond comparison with interval arithmetic.
The proposed method with $\omega$ on the boundary of a grid element. The proposed method failed in Table 3. In this case, the unstable fixed point suggests the importance of $N$. Depth. As an ad hoc remedy, adjusting $X$ to $\omega$ is well in Tables 1 and 2. In particular, the proposed method was faster than the existing method. The existential $f$ is contained in the affine set given by the proposed method. In fact, there is a possibility that the affine set is smaller than $f(X_a)$. Since the error is $O((\text{rad}(X_a))^2)$ by Taylor’s theorem, such underestimation will be harmless if the grid resolution is sufficiently fine. However, such resolution is unknown a priori. Recall that the Lipschitz estimate of the existing method requires $f'(c)$. When we have $f'(c)$, there are two options: (i) determine sample points and apply the existing method, or (ii) compute an affine approximation to apply the proposed method. In order to avoid an underestimation, one may take the option (i) for small subdivision depth, and take the option (ii) for sufficiently large subdivision depth. Or, one may apply the proposed method taking the remainder term into account. An improved implementation is left as a future study.

## Acknowledgments

The first and third authors were supported by JSPS Kakenhi 20K03739 and 19K22232. The second author was supported by JSPS Kakenhi 18H01139 and 20K03747. Figures in the present letter were drawn by gnuplot [12] and Graphviz [13].

## References

[1] E. J. Doedel and B. E. Oldeman, AUTO07-P: continuation and bifurcation software for ordinary differential equations, http://indy.cs.concordia.ca/auto/. (accessed 22 Mar. 2021)

[2] M. Dellnitz et al., The algorithm behind GAIO – Set oriented numerical methods for dynamical systems, in: B. Fiedler (eds) Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems, pp.145–174, Springer-Verlag, Berlin, Heidelberg, 2001.

[3] Z. Arai et al., A database schema for the analysis of global dynamics of multiparameter systems, SIAM J. Appl. Dyn. Syst., 8 (2009), 757–789.

[4] T. Miyaji et al., Human photoplethysmogram through the Morse graph: Searching for the saddle point in experimental data, Chaos, 29 (2019), 043121.

[5] R. J. Lohner, Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems, in: J.R. Cash, I. Gladwell (eds) Computational Ordinary Differential Equations, pp.425–436, Clarendon Press, Oxford, 1992.

[6] T. Miyaji et al., A study of rigorous ODE integrators for multi-scale set-oriented computations, Appl. Numer. Math., 107 (2016), 34–47.

[7] E. Hairer et al., Solving Ordinary Differential Equations I: Nonstiff Problems, Springer-Verlag, Berlin, Heidelberg, 1993.

[8] E. N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci., 20 (1963), 130–141.

[9] P. Pilarczyk, Conley-Morse Graphs Computation Software, https://www.pawelpilarczyk.com/cmgraphs/ (accessed 26 Mar. 2021)

[10] Y. Kuznetsov, Elements of Applied Bifurcation Theory, Third Edition, Springer-Verlag, New York, 2004.

[11] P. Pilarczyk, The Original CHomP Software, http://chomp.rutgers.edu/Projects/ComputationalHomology/OriginalCHomP/software/ (accessed 26 Mar. 2021)

[12] T. Williams and C. Kelley, gnuplot homepage, http://www.gnuplot.info/. (accessed 26 Mar. 2021)

[13] J. Ellison et al., Graphviz - Graph Visualization Software, https://graphviz.org/. (accessed 26 Mar. 2021)