CHAOS **26**, 033102 (2016)

Algorithms for concatenating templates

Robert Gilmore[1,a)] and Martin Rosalie[2,b)]

[1]*Physics Department, Drexel University, Philadelphia, Pennsylvania 19104, USA*
[2]*Univ. Bordeaux, LaBRI, UMR5800, F-33400 Talence, France*

(Received 17 September 2015; accepted 15 February 2016; published online 1 March 2016)

We present two algorithms for concatenating two branched manifolds. There exist two common ways to describe representations of branched manifolds with linking matrices. The algorithms provided permit to concatenate branched manifolds with each representation: a linking matrix with an array or only one linking matrix. We also provide algorithms to switch between these two descriptions. © *2016 AIP Publishing LLC*. [http://dx.doi.org/10.1063/1.4942799]

This paper describes how to concatenate branched manifolds that are used to describe three-dimensional chaotic attractors. This is a particularly useful tool for constructing branched manifolds for dynamical systems that occur when one stretch-and-squeeze mechanism is iterated one or more times as occurs to create "symmetric attractors," for example, for the periodically driven Duffing and van der Pol attractors and the autonomous Lorenz, Chua, and Burke and Shaw attractors. Two different conventions have been introduced to describe branched manifolds. We also provide an algorithm for translating back and forth from one convention to the other.

I. INTRODUCTION

There are various ways to study chaotic attractors and their properties (basins of attraction, fixed points and their stability, periodic orbits, Lyapunov exponents, bifurcation diagrams, etc.). In this paper, we focus on using the topological properties of their periodic orbits to describe their structure using branched manifolds or templates. Templates are constructed from drawings made of splitting charts, torsions, permutations, and joining charts.[1–3] They are adapted to describe the flow of chaotic attractors. Linking matrices are used to describe the torsions and permutations with integers.[4,5] In this article, we use the conventional representation described in Fig. 1 to define the sign of a torsion or a permutation.

The purpose of this paper is to construct concatenations of two (or more) templates. First, we introduce two distinct but equivalent descriptions of templates: one with a matrix and an array and the other with only one matrix. We provide algorithms to switch between these two descriptions of a branched manifold. Second, we present algorithms for concatenating two templates and constructing the matrix and array for the resulting branched manifold. Finally, these algorithms are illustrated with examples.

a)robert.gilmore@drexel.edu
b)martin.rosalie@labri.fr

II. MATRIX DESCRIPTIONS OF TEMPLATES

For attractors bounded by a genus 1-torus, a template is described by a square matrix and an array as first proposed.[4–8] We use a right-handed convention for the matrix; and another convention described by an array, "the higher the further behind," to describe the insertion mechanism. There is another way to describe a template using only one square matrix using the "standard insertion convention."[9,10] This convention indicates that after torsions and permutations, the branch order from the left to right corresponds to the bottom to top order in the insertion mechanism. Fig. 2 details two topologically equivalent templates with these two conventions. The two following linking matrices describe these templates (Fig. 2) of an attractor solution to the Rössler system[11] established by Letellier *et al.*:[12]

$$\left.\begin{array}{l} TM(\text{Rö}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & -2 \\ 0 & -2 & -2 \end{bmatrix} \\ Ar(\text{Rö}) = \begin{bmatrix} 1 & 3 & 2 \end{bmatrix} \end{array}\right| T(\text{Rö}) = \left[\begin{bmatrix} 0 & -1 & -1 \\ -1 & -1 & -2 \\ -1 & -2 & -2 \end{bmatrix}\right]. \tag{1}$$

In order to avoid any confusion in the matrix description, we use a notation introduced in Ref. 13 to emphasize the insertion mechanism respecting the standard insertion convention with a double bracket on the right side.

These two conventions coexist in literature since the 1990s. The linking matrix with an array is constructed using the linking numbers between period one orbits. Moreover, in this conventional representation, the branch order before torsions and permutations and after is the same (Fig. 2). On the other hand, the description with one linking matrix eliminates

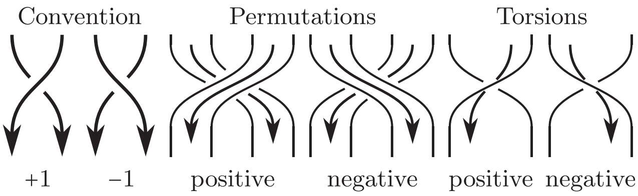
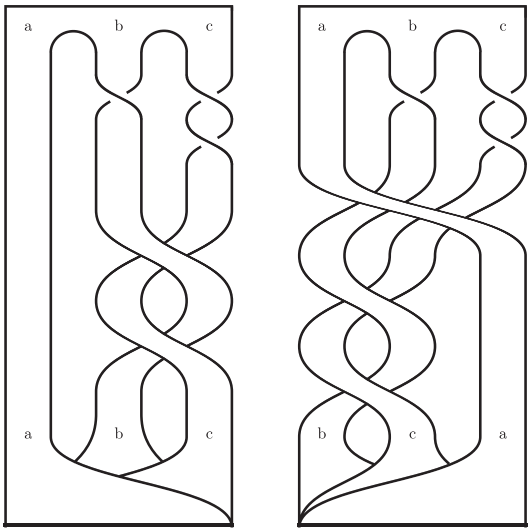


FIG. 1. Conventional representation of oriented crossings. The permutation between two branches is positive if the crossing generated is equal to +1, otherwise it is a negative permutation. We use the same convention for torsions.



(a) a matrix and an array (b) a matrix

FIG. 2. Templates of an attractor solution of the Rössler system. (a) Template described by a matrix and an array (TM(Rö) and Ar(Rö) of (1)). (b) Template described by a matrix (T(Rö) of (1)).

the use of the array by providing a unique way to order branches when the stretching and squeezing mechanism occurs. With this representation using only one linking matrix, Melvin and Tufillaro provide an algorithm[9] to obtain the order of the branches after the torsions and permutations. These two conventions have been used to describe the classical Lorenz attractor template. For instance, recently, this template was described with a matrix and an array by Barrio *et al.* (Fig. 4 of Ref. 14, see also Refs. 7 and 8), while only one matrix was used to describe this chaotic mechanism in Ref. 15.

In the Appendix, we propose an algorithm to obtain the matrix T from TM and Ar, and vice-versa. This is useful to compare the results presented in different papers in the literature. For instance, the template of the Burke-Shaw Attractor[16] can be re-written with an array and a matrix to be compared with other results with this description

$$T(BSA) = \begin{bmatrix} 3 & 2 & 2 & 3 \\ 2 & 2 & 2 & 3 \\ 2 & 2 & 3 & 3 \\ 3 & 3 & 3 & 4 \end{bmatrix}, \tag{2}$$

becomes

$$TM(BSA) = \begin{bmatrix} 3 & 2 & 2 & 4 \\ 2 & 2 & 2 & 4 \\ 2 & 2 & 3 & 4 \\ 4 & 4 & 4 & 4 \end{bmatrix}, \quad Ar(BSA) = \begin{bmatrix} 3 & 2 & 1 & 4 \end{bmatrix}. \tag{3}$$

III. CONCATENATION OF TEMPLATES

We start with a branched manifold A with p_A branches, followed by a second B with p_B branches. To illustrate these algorithms of concatenation, we apply this algorithm to A with three branches describing the inside to outside scroll[7,8]

Inside to Outside Scroll

$$\begin{array}{c|c} \multicolumn{2}{c}{A} \\ \begin{array}{l} TM(A) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 2 \end{bmatrix} \\ Ar(A) = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \end{array} & T(A) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \end{array} \tag{4}$$

and B with two branches describing the horseshoe mechanism

Horseshoe

$$\begin{array}{c|c} \multicolumn{2}{c}{B} \\ \begin{array}{l} TM(B) = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} \\ Ar(B) = \begin{bmatrix} 1 & 2 \end{bmatrix} \end{array} & T(B) = \begin{bmatrix} 0 & -1 \\ -1 & -1 \end{bmatrix}. \end{array} \tag{5}$$

Fig. 3 shows the concatenation of these templates with the two descriptions: with a matrix and an array and with a matrix.

The template A is divided into p_A branches, and at the branch line A, each is further subdivided into p_B pieces. The final template will have $p_A \times p_B$ branches. In our example, the template resulting from the concatenation will have six branches.

IV. CONCATENATION: DESCRIPTION WITH A MATRIX AND AN ARRAY

A. Array *Ar*

The ordering of the pieces $a_i b_j$ takes into account the orientations demanded by the torsions (Fig. 4(a))

$$a_1b_1 \; a_1b_2 \quad a_2b_2 \; a_2b_1 \quad a_3b_1 \; a_3b_2. \tag{6}$$

For A branch, a_2 is orientation-reversing, and for B branch b_2 is orientation reversing, so the ordering is reversed after a_2 and b_2. The ordering along branch line B is

$$b_1a_1 \; b_1a_2 \; b_1a_3 \quad b_2a_3 \; b_2a_2 \; b_2a_1. \tag{7}$$

The ordering array for the concatenated branched manifold is given by the equation

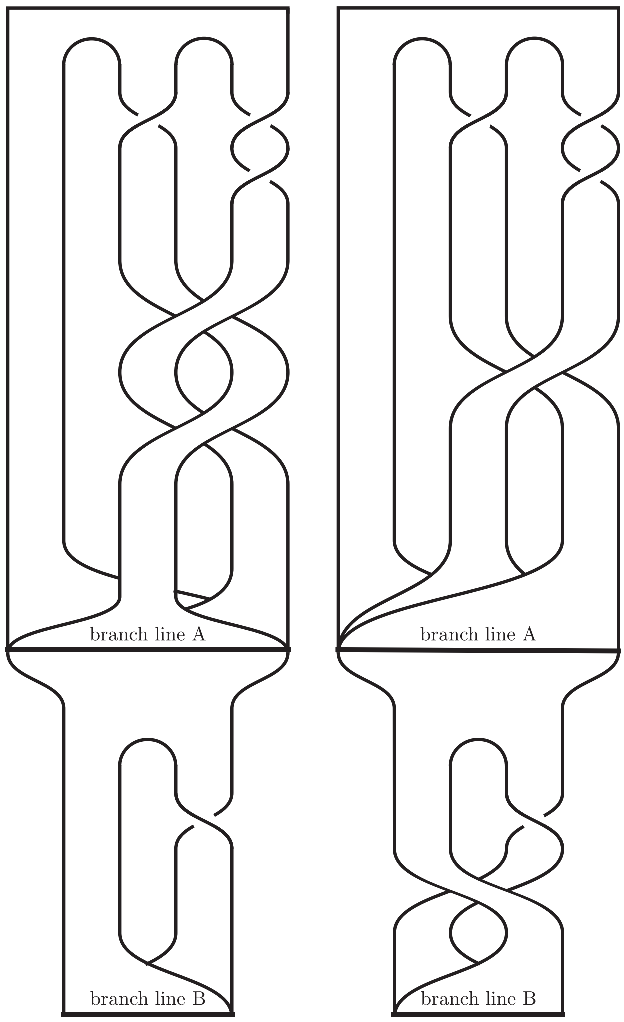
$$Ar(AB)(a_ib_j) = \begin{cases} p_A \times (Ar(b_j) - 1) + Ar(a_i) & \text{if } TM(B)_{jj} \text{ is even} \\ p_A \times Ar(b_j) - (Ar(a_i) - 1) & \text{else.} \end{cases} \tag{8}$$

For example,

$$(a_1b_1, a_1b_2, a_2b_2, a_2b_1, a_3b_1, a_3b_2) = (3, 4, 6, 1, 2, 5). \tag{9}$$

B. Matrix *TM*

The $(p_Ap_B \times p_Ap_B)$ matrix is determined from the linking numbers of the $p_A \times p_B$ period-one orbits in the concatenated template. We find these as follows.



(a) a matrix and an array (b) a matrix

FIG. 3. Concatenation of templates A before B. (a) Templates described by a matrix and an array ($TM(A)$,$Ar(A)$) and ($TM(B)$,$Ar(B)$). (b) Templates described by a matrix ($T(A)$ and $T(B)$).

Follow the branches a_i as constructed from the template matrix and array for A. This is shown at the top in Fig. 4(a). All of the crossings and torsions are read from the template matrix $TM(A)$. At the conclusion of this part of the flow divides each of the components a_i into p_B parts as indicated in (6) and follows these down to the branch line B subdivided into components, as indicated in (7) and the bottom part of Fig. 4. Since a_ib_j and b_ja_i belong to the same period one orbit, it is easy to "connect the dots." Doing so,

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 6 & 5 & 2 & 3 & 4. \end{array} \tag{10}$$

The segment $1(a_1b_1) \to 1(b_1a_1)$ does not cross any other segments. The segment $2(a_1b_2) \to 6(a_3b_2)$ crosses all segments $k \to l$ with $2 < k$ and $l < 6$. Since this segment lies in branch a_1, this segment is underneath all the segments it crosses. As a result, all crossings are right handed and are entered as $+1$ in the appropriate matrix elements of $X(A \downarrow B)$. The segment $3 \to 5$ crosses both the segments $5 \to 3$ and $6 \to 4$ with left-handed crossings. Crossings of segments in the same branch, e.g., $3 \to *, 4 \to **$, have already been counted in the matrix $TM(A)$.

In general, if $i < j$ and a_ib_* crosses a_jb_{**}, then the sign of the crossing is $\mathrm{sign}(Ar(a_i) - Ar(a_j))$.

The matrices determined in these two steps are $TM(A) \otimes I_{p_B}$ and $X(A \downarrow B)$

$$TM(A) \otimes I_{p_B} = \begin{array}{c} a_1b_1 \\ a_1b_2 \\ a_2b_2 \\ a_2b_1 \\ a_3b_1 \\ a_3b_2 \end{array} \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 1 & 1 & 2 & 2 \\ \hline 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 2 & 2 & 2 \end{array}\right],$$

$$X(A \downarrow B) = \left[\begin{array}{cc|cc|cc} & & 0 & 0 & 0 & 0 \\ & & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & & & -1 & -1 \\ 0 & 1 & & & 0 & 0 \\ \hline 0 & 1 & -1 & 0 & & \\ 0 & 1 & -1 & 0 & & \end{array}\right]. \tag{11}$$

This part of the algorithm involves progression from branch line A to branch line B and involves $TM(A)$, and some additional crossing information encoded in the matrix $X(A \downarrow B)$ obtained using the array $Ar(A)$.

The evolution from branch line B to branch line A involves exactly the same steps. The computation is shown in Fig. 4(b). We find

$$I_{p_A} \otimes TM(B) = \begin{array}{c} b_1a_1 \\ b_1a_2 \\ b_1a_3 \\ b_2a_3 \\ b_2a_2 \\ b_2a_1 \end{array} \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 \end{array}\right],$$

$$X(B \downarrow A) = \left[\begin{array}{ccc|ccc} & & & 0 & 0 & 0 \\ & & & 0 & -1 & -1 \\ & & & 0 & -1 & -1 \\ \hline 0 & 0 & 0 & & & \\ 0 & -1 & -1 & & & \\ 0 & -1 & -1 & & & \end{array}\right]. \tag{12}$$

Adding the matrices in (11) and (12) after suitable permutation, we find for the concatenated branched manifold

$$TM(AB) = \begin{array}{c} a_1b_1 \\ a_1b_2 \\ a_2b_2 \\ a_2b_1 \\ a_3b_1 \\ a_3b_2 \end{array} \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 2 \\ \hline 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 2 & 2 & 1 \end{array}\right], \tag{13}$$

$$Ar(AB) = \left[3 \quad 4 \;|6 \quad 1 \;|2 \quad 5\right].$$

In summary, the matrix $TM(AB)$ describing the branched manifold A concatenated with branched manifold B is

$$\begin{aligned} TM(AB) = {} & TM(A) \otimes I_{p_B} + X(A \downarrow B) \\ & + I_{p_A} \otimes TM(B) + X(B \downarrow A), \end{aligned} \tag{14}$$

and the array is given by Equation (8).

V. CONCATENATION WITH ONE MATRIX DESCRIPTION

In this section, we present the same algorithm using the description with only one matrix. There are still p_B branches in each branch of A and p_A branches in each branch of B. The ordering is the following at the beginning (Fig. 5):

$$a_1b_1 \; a_1b_2 \quad a_2b_2 \; a_2b_1 \quad a_3b_1 \; a_3b_2. \tag{15}$$

The ordering depends on the parity of the torsion of the branches of A.

After the torsion and permutations of the branches of A, the order of the strips before the insertion is

$$a_1b_1 \; a_1b_2 \quad a_3b_1 \; a_3b_2 \quad a_2b_1 \; a_2b_2. \tag{16}$$

This can be obtained using the algorithm given by Equation (2) of Ref. 9. Then, to respect the standard insertion, the branches are distributed in the branches of B. This distribution enables us to obtain this order before B

$$a_1b_1 \; a_3b_1 \; a_2b_1 \quad a_1b_2 \; a_3b_2 \; a_2b_2. \tag{17}$$

These three orders are illustrated in Fig. 5. We used them to split the template into three parts, each one described by a matrix.

A. Expansion of matrices

First, the matrix of A is expanded to a block $p_B \times p_B$ square matrix

$$T(A)_{\text{expand}} = \begin{array}{c} a_1b_1 \\ a_1b_2 \\ a_2b_2 \\ a_2b_1 \\ a_3b_1 \\ a_3b_2 \end{array} \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 1 & 1 & 2 & 2 \end{array}\right]. \tag{18}$$

The same can be done for B, but with branch order (17). This gives a block square $p_A \times p_A$ matrix

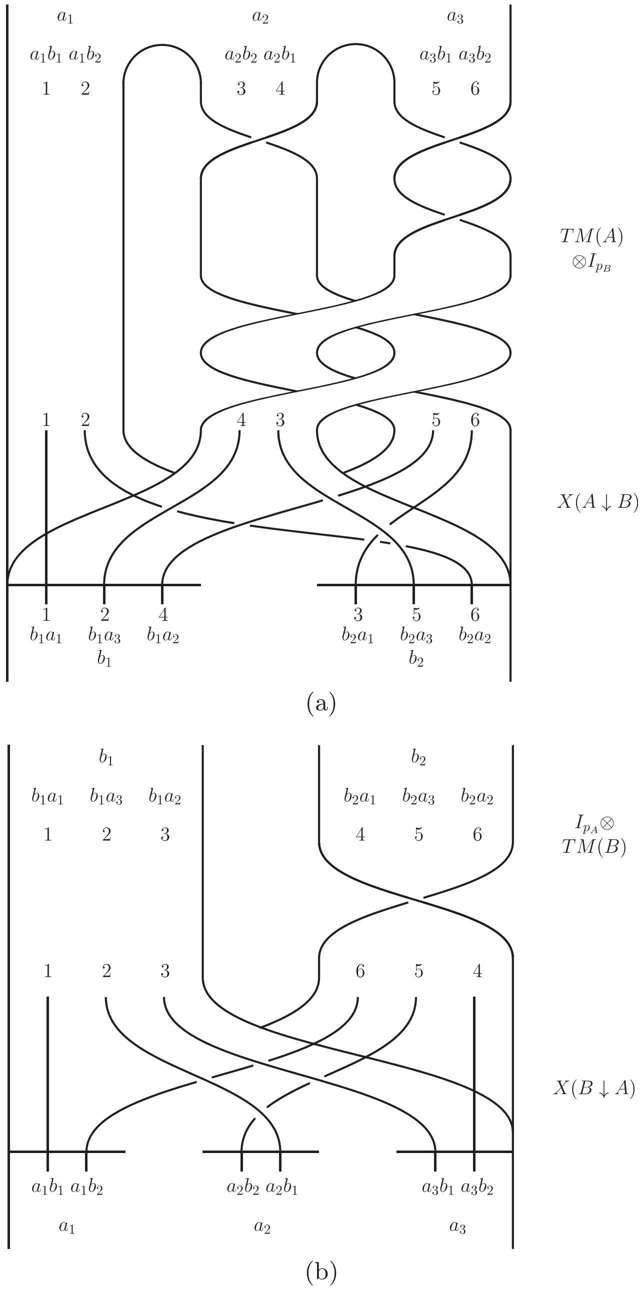


FIG. 4. (a) The branch line A is subdivided into p_A components and each is further subdivided into p_B components, following direct and reverse ordering (at top). The linking specified by the template matrix $TM(A)$ is implemented (top to middle). Further crossings are determined by connecting each component a_ib_j in branch line A to the component b_ja_i in branch line B (middle to bottom). The signs of the crossings are determined from $Ar(A)$, and this information is encoded in the symmetric crossing matrix $X(A \downarrow B)$. (b) Similar treatment for transition from branch line B to branch line A.

$$\begin{array}{c} a_1b_1 \\ a_3b_1 \\ a_2b_1 \\ a_1b_2 \\ a_3b_2 \\ a_2b_2 \end{array} \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{array}\right]. \tag{19}$$

After the rows and columns are permuted to the order given in Eq. (15), we find

$$T(B)_{\text{expand}} = \begin{array}{c} a_1b_1 \\ a_1b_2 \\ a_2b_2 \\ a_2b_1 \\ a_3b_1 \\ a_3b_2 \end{array} \left[\begin{array}{cc|cc|cc} 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & -1 & 0 & 0 & -1 \\ \hline 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{array}\right]. \tag{20}$$

The order of branches depends on the parity of the associated branch of A. For instance, consider the block for the second row and third column. Because $T(A)_{22}$ is odd, the row order is permuted, and because $T(A)_{33}$ is even, the column order is the same; this leads to the block

$$\begin{array}{c} a_2b_2 \\ a_2b_1 \end{array} \left|\begin{array}{cc} -1 & -1 \\ 0 & -1 \end{array}\right|. \tag{21}$$

The matrix $T(A)_{\text{expand}}$ and $T(B)_{\text{expand}}$ are easy to compute and contain permutations and torsions due to $T(A)$ and $T(B)$ (Fig. 5).

B. Permutation from insertion mechanism

Only the permutations due to the insertion mechanism of $T(A)$ are not taken into account with the expanded matrices. These permutations are computed using (16) and (17) by looking for permutations of branches in these arrays. The insertion mechanism respects the standard insertion convention, so only positive permutations or crossings can occur. If a permutation occurs, then it is a positive permutation between these two branches. For instance, a_1b_2 is before a_3b_1 in (16) and a_1b_2 is after a_3b_1 in (17), this implies a positive permutation between these branches. It leads to the following matrix:

$$T(A \to B)_{\text{insertion}} = \begin{array}{c} a_1b_1 \\ a_1b_2 \\ a_2b_2 \\ a_2b_1 \\ a_3b_1 \\ a_3b_2 \end{array} \left[\begin{array}{cc|cc|cc} & & 0 & 0 & 0 & 0 \\ & & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & & & 0 & 0 \\ 0 & 1 & & & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & & \\ 0 & 0 & 0 & 1 & & \end{array}\right]. \tag{22}$$

Note that no permutation occurs for branches coming from the same branch of A. Thus, diagonal blocks of $T(A \to B)_{\text{insertion}}$ are empty. There is no need to take into account the insertion mechanism of the branched manifold B, because the distribution in the insertion mechanism of A respects this insertion convention. The branches distributed in B also respect this convention. Then, the branches are ordered at the end with respect to this convention.

To conclude, the matrix $T(AB)$ describing the branched manifold A concatenated before the branched manifold B is the sum of the three matrices (18), (22), and (20)

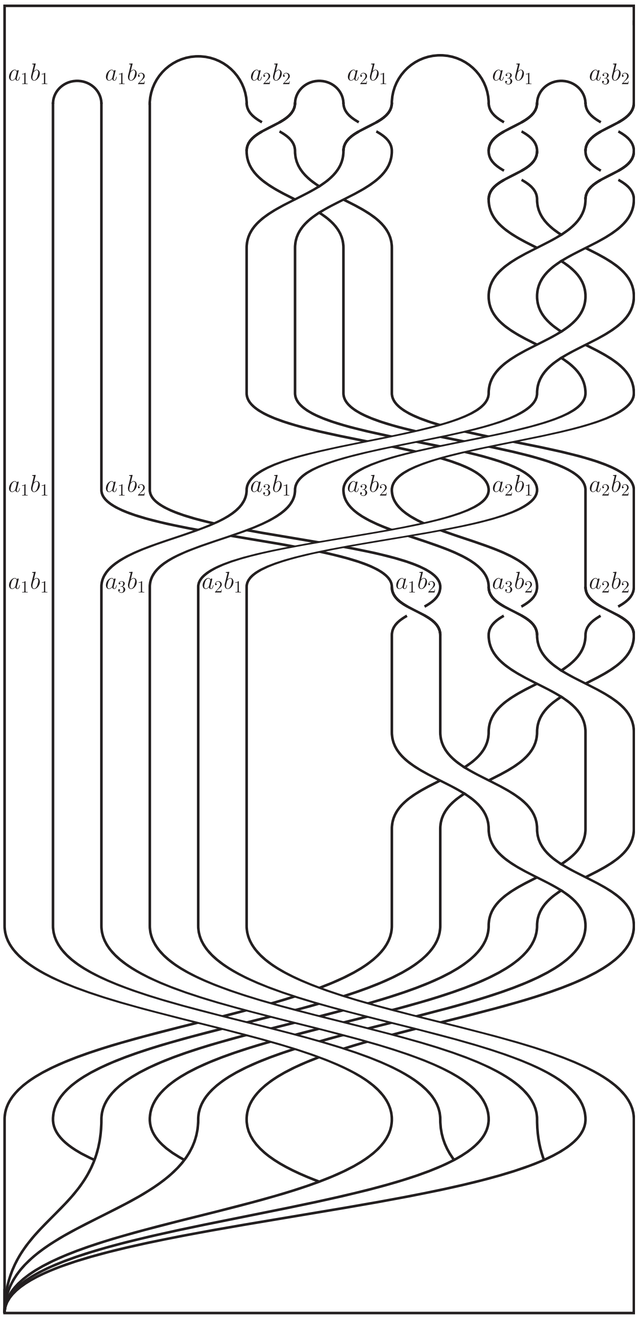


FIG. 5. Concatenation of branched manifold A before B. The three branches of A are split into two branches. Then, the insertion mechanism of A distributes the branches in B. Finally, the two branches of B are split in three branches. The order of the three branches, (15)–(17), is indicated. This order splits the template into three parts, the first part is described with matrix $T(A)_{\text{expand}}$ (18). The second part is described with $T(A \to B)_{\text{insertion}}$ (22). The last part is described by $T(B)_{\text{expand}}$ (20). The addition of these three matrices gives the result of the concatenation.

$$T(AB) = T(A)_{\text{expand}} + T(A \to B)_{\text{insertion}} + T(B)_{\text{expand}}$$
$$= \left[\begin{array}{cccccc} 0 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ -1 & -1 & 0 & 1 & 1 & 1 \end{array}\right]. \tag{23}$$

VI. ANOTHER APPLICATION

We will compute the concatenation for two oppositely oriented Smale horseshoe branched manifolds, each followed by a half-twist. The concatenation of such a pair can be used to construct the branched manifold for a chaotic attractor with inversion symmetry. One such attractor has been proposed by Malasoma,[17] and the concatenation has been applied for this attractor in Ref. 18.

The algebraic description of a Smale horseshoe is

$$\begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \qquad (24)$$
$$\begin{bmatrix} 0 & 1 \end{bmatrix},$$

for two branches labeled 0, 1. When this is followed by a half-twist and the branches are labeled r, s, this description is as shown on the left in (25) below:

$$\begin{array}{cc|cc} \multicolumn{2}{c}{A} & & \\ \mathrm{r}\ \ \mathrm{s} & & \mathrm{r}\ \ \mathrm{s} & \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} & & \left[\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\right] & \\ \begin{bmatrix} 2 & 1 \end{bmatrix} & & & \end{array} \quad \begin{array}{cc|cc} \multicolumn{2}{c}{B} & & \\ \mathrm{R}\ \ \mathrm{S} & & \mathrm{R}\ \ \mathrm{S} & \\ \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} & & \left[\begin{bmatrix} -1 & -1 \\ -1 & 0 \end{bmatrix}\right] & \\ \begin{bmatrix} 1 & 2 \end{bmatrix} & & & \end{array}. \qquad (25)$$

The second branched manifold is described on the right in (25).

A. Concatenation using *TM* and *Ar*

The order of the four segments on branch line A is $(rS, rR, sR, sS) = (1,2,3,4)$ and along branch line B it is $(Rs, Rr, Sr, Ss) = (1,2,3,4)$.

When the algorithm described above is applied to concatenate the two templates (lower case, followed by upper case), the results are

$$TM(A) \otimes I_2 = \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right],$$
$$X(A \downarrow B) = \begin{array}{c} \left[\begin{array}{cc|cc} & & 1 & 0 \\ & & 1 & 0 \\ \hline 1 & 1 & & \\ 0 & 0 & & \end{array}\right] \end{array} \begin{array}{l} rS \\ rR \\ sR \\ sS \end{array}. \qquad (26)$$

The crossing segments are shown to the right in (26). This information comes from $(rS, rR, sR, sS) \rightarrow (Sr, Rr, Rs, Ss)$ or $(1,2,3,4) \rightarrow (4,3,1,4)$, which implies two positive crossings.

For the transition from B to A, we find

$$I_2 \otimes TM(B) = \left[\begin{array}{cc|cc} -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right],$$
$$X(A \downarrow B) = \left[\begin{array}{cc|cc} & & -1 & 0 \\ & & -1 & 0 \\ \hline -1 & -1 & & \\ 0 & 0 & & \end{array}\right] \begin{array}{l} Rs \\ Rr \\ Sr \\ Ss \end{array}. \qquad (27)$$

In this step, $(Rs, Rr, Sr, Ss) \rightarrow (sR, rR, rS, sS)$ or $(1,2,3,4) \rightarrow (3,2,1,4)$, which implies two negative crossings.

Finally, rearranging the rows and columns as appropriate and adding all these matrices, we find

$$TM(AB) = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \begin{array}{l} rSr \\ rRr \\ sRs \\ sSs \end{array}, \qquad (28)$$
$$Ar(AB) = \begin{bmatrix} 4 & 1 & 2 & 3 \end{bmatrix}.$$

B. Concatenation using *T*

For the concatenation using the branched manifold description with only one matrix, we first compute the branch order. We first obtain the branch order at the beginning of A

$$rS\ rR \quad sR\ sS. \qquad (29)$$

Then, we obtain the order of the branches at the end of A and before the insertion mechanism

$$rR\ rS \quad sR\ sS. \qquad (30)$$

And finally, the order of the branches at the beginning of B is

$$rR\ sR \quad rS\ sS. \qquad (31)$$

We expand the matrices A and B

$$T(A)_{\text{expand}} = \begin{array}{c} rS \\ rR \\ sR \\ sS \end{array} \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right],$$
$$T(B)_{\text{expand}} = \begin{array}{c} rS \\ rR \\ sR \\ sS \end{array} \left[\begin{array}{cc|cc} 0 & -1 & -1 & 0 \\ -1 & -1 & -1 & -1 \\ \hline -1 & -1 & -1 & -1 \\ 0 & -1 & -1 & 0 \end{array}\right], \qquad (32)$$

and compute the matrix containing the permutations induced by the insertion mechanism from (30) and (31)

$$T(A \rightarrow B)_{\text{insertion}} = \begin{array}{c} rS \\ rR \\ sR \\ sS \end{array} \left[\begin{array}{cc|cc} & & 1 & 0 \\ & & 0 & 0 \\ \hline 1 & 0 & & \\ 0 & 0 & & \end{array}\right]. \qquad (33)$$

Finally, we add the three matrices (32) and (33) to obtain

$$T(AB) = T(A)_{\text{expand}} + T(A \rightarrow B)_{\text{insertion}} + T(B)_{\text{expand}}$$
$$= \begin{array}{c} rS \\ rR \\ sR \\ sS \end{array} \left[\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & -1 & -1 & -1 \\ 0 & -1 & -1 & 0 \end{bmatrix}\right]. \qquad (34)$$

VII. CONCLUSION

In this paper, we present algorithms to concatenate branched manifolds. These algorithms are described for two conventions: with a matrix and an array or only with one matrix. We take this opportunity to give algorithms to move from one convention to the other. The concatenation has already been described and applied as mentioned in this paper, and here we provide efficient algorithms, described in a simple way, to concatenate templates.

APPENDIX: ALGORITHMS TO OBTAIN THE EQUIVALENT REPRESENTATION

These algorithms are important because the two descriptions of branched manifold coexist in the literature. This will help researchers to compare their results.

The algorithms below are used to transform one mathematical description of a branched manifold A using $TM(A)$ and $Ar(A)$ to another using $T(A)$, and inversely. These algorithms were used to show the equivalence of the descriptions presented in (13) and (23).

1. From a matrix and an array to one matrix

The following algorithm transforms a template description from TM and Ar to T with n the number of branches:

$$
\begin{array}{l}
T \leftarrow TM \\
\text{For } j \text{ from } 1 \text{ to } n \\
\quad tab(j) \leftarrow Ar(j) \\
\text{end For} \\
swapped \leftarrow \text{true} \\
\text{While } swapped = \text{true do} \\
\quad swapped \leftarrow \text{false} \\
\quad \text{For } i \text{ from } 1 \text{ to } n-1 \\
\quad\quad \text{If } tab(i) < tab(i+1) \text{ then} \\
\quad\quad\quad k \leftarrow \text{position of } tab(i) \text{ in } Ar \\
\quad\quad\quad l \leftarrow \text{position of } tab(i+1) \text{ in } Ar \\
\quad\quad\quad T(k,l) \leftarrow T(k,l) - 1 \\
\quad\quad\quad T(l,k) \leftarrow T(l,k) - 1 \\
\quad\quad\quad swapped \leftarrow \text{true} \\
\quad\quad \text{end If} \\
\quad \text{end For} \\
\quad n \leftarrow n-1 \\
\text{end While}
\end{array}
\tag{A1}
$$

2. From a matrix to a matrix and an array

The following algorithm transforms a template description from T to TM and Ar with n the number of branches. First, we obtain Ar:

$$
\begin{array}{l}
\text{For } i \text{ from } 1 \text{ to } n \\
\quad positive \leftarrow 0 \\
\quad negative \leftarrow 0 \\
\quad \text{For } j \text{ from } 1 \text{ to } i-1 \\
\quad\quad \text{If } T(i,j) \text{ is odd then} \\
\quad\quad\quad negative \leftarrow negative + 1 \\
\quad\quad \text{end If} \\
\quad \text{end For} \\
\quad \text{For } j \text{ from } i+1 \text{ to } n \\
\quad\quad \text{If } T(i,j) \text{ is odd then} \\
\quad\quad\quad positive \leftarrow positive + 1 \\
\quad\quad \text{end If} \\
\quad \text{end For} \\
\quad Ar(i + positive - negative) \leftarrow n - i + 1 \\
\text{end For}
\end{array}
\tag{A2}
$$

Then, we obtain TM:

$$
\begin{array}{l}
tab \leftarrow Ar \\
TM \leftarrow T \\
swapped \leftarrow \text{true} \\
\text{While } swapped = \text{true do} \\
\quad swapped \leftarrow \text{false} \\
\quad \text{For } i \text{ from } 1 \text{ to } n-1 \\
\quad\quad \text{If } tab(i) < tab(i+1) \text{ then} \\
\quad\quad\quad k \leftarrow \text{position of } tab(i) \text{ in } Ar \\
\quad\quad\quad l \leftarrow \text{position of } tab(i+1) \text{ in } Ar \\
\quad\quad\quad TM(k,l) \leftarrow TM(k,l) + 1 \\
\quad\quad\quad TM(l,k) \leftarrow TM(l,k) + 1 \\
\quad\quad\quad swapped \leftarrow \text{true} \\
\quad\quad \text{end If} \\
\quad \text{end For} \\
\quad n \leftarrow n-1 \\
\text{end While}
\end{array}
\tag{A3}
$$

[1]J. Birman and R. F. Williams, "Knotted periodic orbits in dynamical systems—I: Lorenz's equations," Topology **22**, 47–82 (1983).
[2]J. Franks and R. F. Williams, "Entropy and knots," Trans. Am. Math. Soc. **291**, 241–253 (1985).
[3]R. W. Ghrist, P. J. Holmes, and M. C. Sullivan, *Knots and Links in Three-Dimensional Flows*, Books, Monographs and Lecture Notes Vol. 1 (Springer, 1997).
[4]H. G. Solari and R. Gilmore, "Relative rotation rates for driven dynamical systems," Phys. Rev. A **37**, 3096–3109 (1988).
[5]G. B. Mindlin, X.-J. Hou, H. G. Solari, R. Gilmore, and N. B. Tufillaro, "Classification of strange attractors by integers," Phys. Rev. Lett. **64**, 2350–2353 (1990).
[6]G. B. Mindlin, H. G. Solari, M. A. Natiello, R. Gilmore, and X.-J. Hou, "Topological analysis of time series data from the Belousov-Zhaborinskii reaction," J. Nonlinear Sci. **1**, 147–173 (1991).
[7]R. Gilmore, "Topological analysis of chaotic dynamical systems," Rev. Mod. Phys. **70**, 1455–1530 (1998).
[8]R. Gilmore and M. Lefranc, *The Topology of Chaos* (Wiley, NY, 2002).
[9]P. Melvin and N. B. Tufillaro, "Templates and framed braids," Phys. Rev. A **44**, R3419 (1991).
[10]N. B. Tufillaro, T. Abbott, and J. Reilly, *An Experimental Approach to Nonlinear Dynamics and Chaos* (Addison-Wesley, Redwood City, 1992).

[11]O. E. Rössler, "An equation for continuous chaos," Phys. Lett. A **57**, 397–398 (1976).
[12]C. Letellier, P. Dutertre, and B. Maheu, "Unstable periodic orbits and templates of the Rössler system: Toward a systematic topological characterization," Chaos **5**, 271–282 (1995).
[13]M. Rosalie and C. Letellier, "Systematic template extraction from chaotic attractors: I. Genus-one attractors with an inversion symmetry," J. Phys. A: Math. Theor. **46**, 375101 (2013).
[14]R. Barrio, A. Dena, and W. Tucker, "A database of rigorous and high-precision periodic orbits of the Lorenz model," Comput. Phys. Commun. **194**, 76–83 (2015).
[15]M. Rosalie and C. Letellier, "Toward a general procedure for extracting templates from chaotic attractors bounded by high genus torus," Int. J. Bifurcation Chaos **24**(4), 1450045 (2014).
[16]C. Letellier, P. Dutertre, J. Reizner, and G. Gouesbet, "Evolution of a multimodal map induced by an equivariant vector field," J. Phys. A: Math. Gen. **29**, 5359–5373 (1996).
[17]J.-M. Malasoma, "What is the simplest dissipative chaotic jerk equation which is parity invariant?," Phys. Lett. A **264**, 383–389 (2000).
[18]M. Rosalie and C. Letellier, "Systematic template extraction from chaotic attractors: II. Genus-one attractors with multiple unimodal folding mechanisms," J. Phys. A: Math. Theor. **48**, 235101 (2015).