OPEN ACCESS
Discrete Mathematics, Algorithms and Applications
Vol. 13, No. 2 (2021) 2150023 (32 pages)

DOI: 10.1142/S1793830921500233

Complete transition diagrams of generic Hamiltonian flows with a few heteroclinic orbits

Tetsuo Yokoyama

Department of Software Engineering, Nanzan University
Yamazato-cho 18, Showa-ku, Nagoya, Aichi 466-8673, Japan
tyokoyama@acm.org

Tomoo Yokoyama*

Department of Mathematics
Kyoto University of Education/JST PRESTO
1 Fujinomori, Fukakusa, Fushimi-ku, Kyoto 612-8522, Japan
tomoo@kyokyo-u.ac.jp

Received 30 September 2019
Revised 6 August 2020
Accepted 4 September 2020
Published 28 October 2020

We study the transition graph of generic Hamiltonian surface flows, whose vertices are the topological equivalence classes of generic Hamiltonian surface flows and whose edges are the generic transitions. Using the transition graph, we can describe time evaluations of generic Hamiltonian surface flows (e.g., fluid phenomena) as walks on the graph. We propose a method for constructing the complete transition graph of all generic Hamiltonian flows. In fact, we construct two complete transition graphs of Hamiltonian surface flows having three and four genus elements. Moreover, we demonstrate that a lower bound on the transition distance between two Hamiltonian surface flows with any number of genus elements can be calculated by solving an integer programming problem using vector representations of Hamiltonian surface flows.

Keywords: Graph; Hamiltonian flow; topology; automaton; formal language.

Mathematics Subject Classification 2020: 90C35, 37E35, 76B47

1. Introduction

The topological properties of Hamiltonian flows on compact surfaces have been studied by several authors. For instance, such flows with finitely many singular points

*Corresponding author.

have been topologically classified on an unbounded plane [2], on a sphere [5], on a torus [8], and on closed surfaces [4]. It is known that various kinds of fluid phenomena are modeled as incompressible fluids and that incompressible flows on spheres are Hamiltonian (cf. [6]). Structural stability is important from a dynamical system's point of view because the set of structurally stable Hamiltonian flows on a bounded or unbounded domain is open dense [6, 11]. In other words, any Hamiltonian surface flow can be approximated by such Hamiltonian surface flows, and the topological equivalence class of these flows is preserved under small perturbations. Structural stability is also important from an application point of view, because Hamiltonian flows observed in experiments and numerical calculations are structurally stable at almost every moment in time. Moreover, structurally stable Hamiltonian flows on bounded and unbounded domains are classified topologically [6, 11]. In other words, any structurally stable Hamiltonian flow on an unbounded (respectively, a bounded) domain can be constructed from a uniform flow (see Fig. 1) on the plane $\mathbb{U}(0)$ (respectively, a pointwise periodic flow (see Fig. 1) on the annulus $\mathbb{D}(1)$) by iteratively applying five operations, each of which creates a genus element. By *operation*, we mean one of several specific types of orbit structure replacement, as we will see later in Figs. 8 and 12.

Therefore, a word of operations is the set of structurally stable Hamiltonian flows generated by those operations. Generic transitions of such structurally stable Hamiltonian flows have also been characterized [10]. Such generic transitions form a graph, called a *transition graph*, whose vertices are topological equivalence classes of structurally stable Hamiltonian flows and whose edges are generic transitions.

In this paper, we algorithmically construct transition graphs for words of the same length, where an edge between two words exists if a Hamiltonian flow generated by one of the words can be transitioned to a Hamiltonian flow generated by the other word. The properties of the graphs and of the words can be analyzed using graph theory, formal language theory, and tools in those fields. This abstraction helps non-experts access the analysis of fluid dynamics. Using the transition graph, we can completely describe the time evaluations of generic Hamiltonian surface flows (e.g., fluid phenomena on multi-connected domains) as walks on the graph. The visualization provided by the graph is helpful when reasoning about fluid dynamics, as, for example, during the process of debugging simulation software.

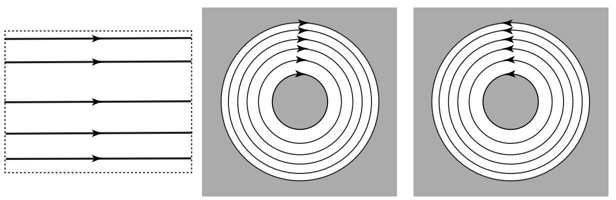


Fig. 1. Left: uniform flow; middle and right: pointwise periodic flows on $\mathbb{D}(1)$.

A lower bound on the transition distance between a pair of Hamiltonian surface flows with any genus elements can be calculated by solving an integer programming problem by vector representations, which are reduced representations of word representations of Hamiltonian surface flows. The vector representations of Hamiltonian surface flows with a transition distance are suitable for big data analysis.

This paper consists of six sections. In Sec. 2, we review the notation for dynamical systems and the framework of topological flow data analysis. In particular, several facts about representations of two-dimensional (2D) Hamiltonian flows and transition graphs are presented in a self-contained manner with prerequisites. In Sec. 3, vector representations of 2D Hamiltonian flows are introduced, and the necessary conditions for transitions among structurally stable Hamiltonian flows are stated. In Sec. 4, a list of all transition rules is given, and the complete transition graphs of lengths two, three, and four are presented. In Sec. 5, we explain a method for analyzing the transition distances of diagrams. In Sec. 6, we provide a supplementary explanation.

2. Preliminaries

2.1. *Notation for dynamical systems*

We recall some basic notation. Good references for most of what we describe are [6, 1].

In this paper, a domain is a connected surface with or without boundary contained in a 2D sphere or a plane. A boundary component is a connected component of the boundary ∂S. A flow on a domain is a continuous $\mathbb{R}$-action on the domain. In other words, a continuous mapping $v{:}\mathbb{R} \times S \to S$ on a domain S is a *flow* if $v(t, \cdot){:}S \to S$ is a homeomorphism with $v(t, v(s, x)) = v(t + s, x)$ for any $s, t \in \mathbb{R}$ such that $v(0, \cdot)$ is identical to it. For a point x in S, let $O(x) := \{v(t, x) \,|\, t \in \mathbb{R}\}$, called the *orbit* of x. A point x is *singular* if its orbit consists of one point, that is, if $x = v(t, x)$ for any $t \in \mathbb{R}$. An orbit is singular if it consists of a singular point. A non-singular orbit of a flow is a *separatrix* if it starts or ends at a singular point. A singular point p of a flow v generated by a C^1-vector field X is non-degenerate if its determinant of the Hesse matrix $(\partial X_i/\partial x_j(p))$ is non-zero (i.e., if $\partial X_1/\partial x_1(p) \cdot \partial X_2/\partial x_2(p) - \partial X_1/\partial x_2(p) \cdot \partial X_2/\partial x_1(p) \neq 0$). A non-degenerate singular point of v is a *center* if its eigenvalues of the Hesse matrix are purely imaginary, as in the two rightmost depictions in Fig. 2.

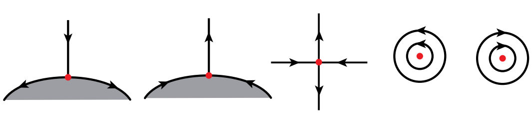


Fig. 2. Non-degenerate singular points: From left to right: two ∂-saddles, a saddle, and two centers.

A non-degenerate singular point of v outside of the boundary is a *saddle* if its eigenvalues of the Hesse matrix consist of a positive number, and a negative number as in the depiction in the middle of Fig. 2. A non-degenerate singular point of v on the boundary is a *∂-saddle*[a] if its eigenvalues of the Hesse matrix consist of a positive number and a negative number, as in the two leftmost depictions in Fig. 2. A saddle connection diagram is the union of the saddles, ∂-saddles, and separatrices. A saddle connection is a connected component of the saddle connection diagram. Note that the saddle connection diagram in the unbounded case is also called the ss-saddle connection diagram in [11]. A separatrix is self-connected if either it starts and ends at the same saddle or it connects two ∂-saddles on the same boundary component (see Fig. 3). Thus, a separatrix is non-self-connected if it connects two saddles, two distinct ∂-saddles, or a saddle and a ∂-saddle.

A point x is *periodic* if there is a positive number $T > 0$ such that $x = v(T, x)$ and $x \neq v(t, x)$ for any $t \in (0, T)$. Recall that a vector field X_1 on a domain S_1 and a vector field X_2 on a domain S_2 are topologically equivalent if there is a homeomorphism $h: S_1 \to S_2$ such that the image of an orbit of X_1 is an orbit of X_2 and h preserves the orientation of the orbits. Similarly, a flow v_1 on a domain S_1 and a flow v_2 on a domain S_2 are topologically equivalent if there is a homeomorphism $h: S_1 \to S_2$ such that the image of an orbit of v_1 is an orbit of v_2 and h preserves the orientation of the orbits.

2.2. *Notation for Hamiltonian flows*

A C^r $(r \geq 1)$ vector field X on a domain S is Hamiltonian if there is a C^{r+1} function $H : S \to \mathbb{R}$ such that $X(p) = (\partial H/\partial x_2, -\partial H/\partial x_1)$ for any point $p \in S$, where (x_1, x_2) is a local coordinate system of p. The definition of Hamiltonian vector field implies that the flow generated by a Hamiltonian vector field is incompressible. By *Hamiltonian flow*, we mean a flow generated by a Hamiltonian vector field. We say that a Hamiltonian flow is a Hamiltonian flow *on a bounded domain* if the domain is

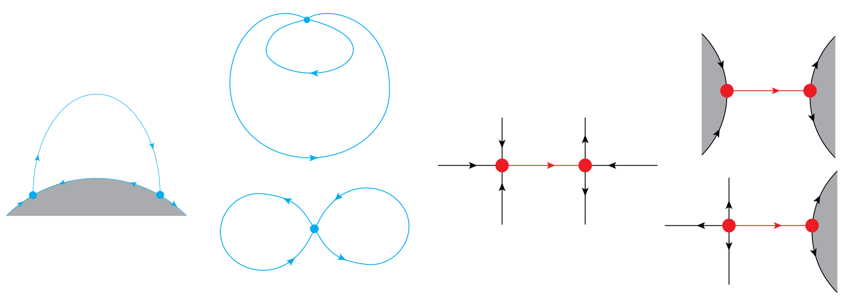


Fig. 3. (Color online) Left: self-connected separatrices (in blue); right: non-self-connected separatrices (in red).

[a] "∂-saddle" is pronounced "boundary-saddle".

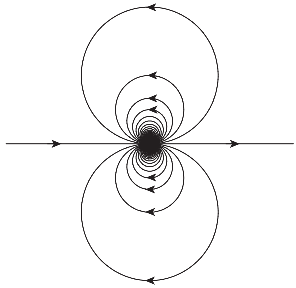


Fig. 4. A 1-source–sink.

compact. Notice that for the flow v on $\mathbb{R}^2$ generated by a vector field (1,0), which is a uniform flow, the point ∞ at infinity corresponds to a 1-source–sink (i.e., dipole) (Fig. 4) with respect to the extended flow of v on the one-point compactification $\mathbb{R}^2 \cup \{\infty\}$.

Therefore, we say that a Hamiltonian flow v on an unbounded domain $S \subseteq \mathbb{R}^2$ is a Hamiltonian flow with a 1-source–sink at infinity if the domain is the resulting surface by removing finitely many open disks from $\mathbb{R}^2$ such that the point ∞ at infinity corresponds to a 1-source–sink with respect to the extended flow of v on the one-point compactification $S \cup \{\infty\}$. For a natural number n, let $\mathbb{D}(n)$ be the resulting compact domain of a closed disk formed by removing n pairwise disjoint open disks, and $\mathbb{U}(n)$ the resulting domain of $\mathbb{R}^2$ formed by removing n pairwise disjoint open disks. Denote by $\mathcal{H}_{\mathrm{bd}}(n)$ the set of Hamiltonian flows on $\mathbb{D}(n)$, and by $\mathcal{H}_{\mathrm{ubd}}(n)$ the set of Hamiltonian flows with a 1-source–sink at infinity on $\mathbb{U}(n)$. A C^r $(r \geq 1)$ Hamiltonian vector field on $\mathbb{D}(n)$ (respectively, $\mathbb{U}(n)$) is structurally stable (in the set of C^r Hamiltonian vector fields on $\mathbb{D}(n)$ (respectively, $\mathbb{U}(n)$)) if it is invariant under small C^1-perturbations in the set of C^r Hamiltonian vector fields on $\mathbb{D}(n)$ (respectively, $\mathbb{U}(n)$) up to topological equivalence. A Hamiltonian flow in $\mathcal{H}_{\mathrm{bd}}(n)$ (respectively, $\mathcal{H}_{\mathrm{ubd}}(n)$) is structurally stable (in $\mathcal{H}_{\mathrm{bd}}(n)$ (respectively, $\mathcal{H}_{\mathrm{ubd}}(n)$)) if it is generated by a structurally stable Hamiltonian vector field in the set of C^r Hamiltonian vector fields on $\mathbb{D}(n)$ (respectively, $\mathbb{U}(n)$).

By the definitions of the types of singular points of Hamiltonian flows, incompressibility implies that a non-degenerate singular point must be either a saddle, a ∂-saddle, or a center. Therefore, a non-degenerate singular point is a saddle if and only if it has exactly four separatrices counted with multiplicity, and a non-degenerate singular point is a ∂-saddle if and only if it has exactly three separatrices. A non-singular orbit is an *ss-orbit* if it is not a separatrix but is unbounded. Note that an ss-orbit starts from and goes to the point at infinity. A non-singular orbit is an *ss-separatrix* if it is a separatrix and is unbounded. Note that an ss-orbit is between the point at infinity and either a saddle or a ∂-saddle. A saddle connection is *self-connected* if any separatrix in it either is self-connected or is an ss-separatrix. By the definition of self-connected saddle connection, any self-connected saddle connection is locally of the form shown in Fig. 5.

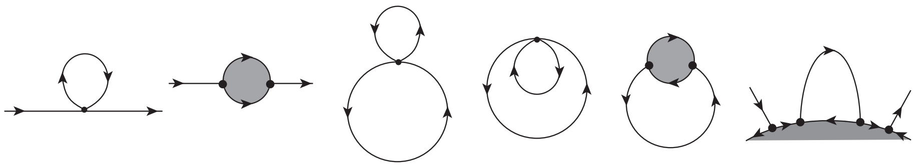


Fig. 5. List of the local structures of self-connected saddle connections. Boundaries are shaded for readability.

A saddle connection is called type a_0, a_2, b_0, b_2, or c (shown from left to right in the figure).

A genus element is either a center or a periodic orbit on a boundary.

2.3. *Structural stability of Hamiltonian flows*

The structural stability of Hamiltonian flows is generic and is topologically characterized as follows (see [6, 11] for details).

Lemma 2.1 ([6, Theorem 2.3.8, p. 74; 11, Theorem 3.2]). *Let $\mathcal{H}$ denote either $\mathcal{H}_{\mathrm{bd}}(n)$ or $\mathcal{H}_{\mathrm{ubd}}(n)$. The set of structurally stable Hamiltonian flows in $\mathcal{H}$ is open dense in $\mathcal{H}$. Moreover, the following statements are equivalent:*

(1) *A Hamiltonian flow in $\mathcal{H}$ is structurally stable in $\mathcal{H}$.*
(2) *Each singular point is non-degenerate and each saddle connection is self-connected (i.e., each separatrix either is self-connected or is an ss-separatrix).*

By *generic Hamiltonian flow*, we mean, therefore, a structurally stable Hamiltonian flow. Morse theory states that any Hamiltonian flow with non-degenerate singular points is determined by the saddle connection diagram, up to topological equivalence. More precisely, the following statement holds.

Lemma 2.2 (cf. [6, Theorem 1.4.6, p. 42; 11, Remark after Theorem 3.2, p. 10]). *Any Hamiltonian flow in $\mathcal{H}_{\mathrm{bd}}(n)$ or $\mathcal{H}_{\mathrm{ubd}}(n)$ is determined by the saddle connection diagram, up to topological equivalence. Moreover, any connected component of the complement of the union of the saddle connection diagrams is either an annulus consisting of periodic orbits or a flow box consisting of ss-orbits (see Fig. 6).*

Lemma 2.2 means that the topological equivalence class of stable Hamiltonian flows can be identified using the saddle connection diagram.

2.4. *Generic transitions between structurally stable Hamiltonian flows*

A *k-saddle* is an isolated singular point outside of the boundary of a domain with exactly $2k+2$ separatrices, counted with multiplicity for a non-negative integer k,

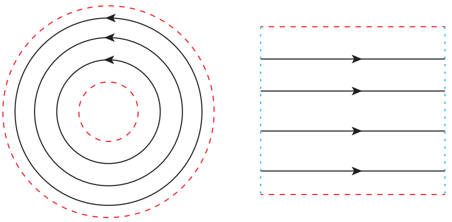


Fig. 6. Left: an open annulus; right: A flow box.

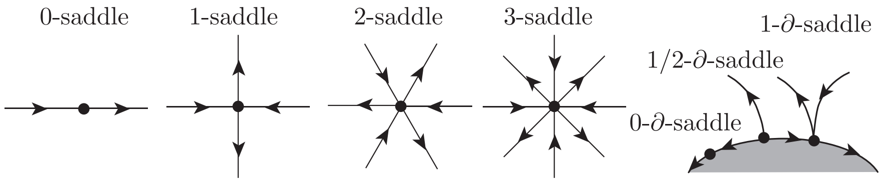


Fig. 7. Examples of multi-saddles.

as shown in Fig. 7. A ∂-$k/2$-*saddle* is an isolated singular point on the boundary of a domain with exactly $k+2$ separatrices, counted with multiplicity for a non-negative integer k, as shown in Fig. 7.

A *multi-saddle* is either a k-saddle or a ∂-$k/2$-saddle for some integer k. The multi-saddle connection diagram of a Hamiltonian flow is the union of multi-saddles and separatrices. It is known that each singular point of a Hamiltonian flow with finitely many singular points on a bounded domain is either a topological center or a multi-saddle because Hamiltonian flow on a compact surface is non-wandering (i.e., each point is non-wandering) (see [3, Theorem 3]). Here, a topological center is locally topologically equivalent to a center. In other words, a topological center is a center up to local topological equivalence. A 0-saddle is called a fake saddle, and a ∂-0-saddle is called a fake ∂-saddle. A multi-saddle is *fake* if it is either a fake saddle or a fake ∂-saddle. Here, a point is non-wandering if for any neighborhood U of the point and for any positive number N, there is a number $t \in \mathbb{R}$ with $|t| > N$ such that $v_t(U) \cap U \neq \emptyset$.

A Hamiltonian flow with self-connected saddle connections is *f-unstable* if it has just one fake multi-saddle and each singular point except for the fake multi-saddle is non-degenerate. The "f" in "f-unstable" stands for "fake." With the same fixed number of genus elements, any perturbation of an f-unstable Hamiltonian flow on a bounded or unbounded domain implies the same structurally stable Hamiltonian flow up to topological equivalence. In other words, a transition whose intermediate flow is f-unstable is trivial under fixing of the same number of genus elements. In addition, a Hamiltonian flow with self-connected saddle connections is *t-unstable* if it has just one topological center, and each singular point except for the topological center is non-degenerate. The "t" in "t-unstable" stands for "topological center".

With the same fixed number of genus elements, any perturbation of a t-unstable Hamiltonian flow on a bounded or unbounded domain implies the same structurally stable Hamiltonian flow up to topological equivalence. In other words, a transition whose intermediate flow is t-unstable is trivial under fixing of the same number of genus elements. We consider, therefore, the following condition for the non-existence of fake multi-saddles and topological centers to omit trivial transitions:

($A1$) There are neither fake multi-saddles nor topological centers.

This condition means that any singular point of a Hamiltonian flow with finitely many singular points is a multi-saddle under these assumptions. A Hamiltonian flow with non-degenerate singular points is *h-unstable* if it has exactly one non-self-connected orbit in the saddle connection diagram. A Hamiltonian flow with self-connected saddle connections is *p-unstable* if it has just one pinching point and each singular point except for the pinching point is non-degenerate. The "h" in "h-unstable" and the "p" in "p-unstable" stand for "heteroclinic" and "pinching", respectively. A generic transition between structurally stable Hamiltonian flows with the same number of genus elements is either p-unstable or h-unstable. More precisely, a characterization of generic transitions of Hamiltonian flows is described as follows (see [10] for details).

Lemma 2.3 ([10, Proposition 3.1]). *Let $\mathcal{H}(n)$ be either the set of flow in $\mathcal{H}_{\mathrm{bd}}(n)$ satisfying condition ($A1$) or the set of flows in $\mathcal{H}_{\mathrm{ubd}}(n)$ satisfying condition ($A1$). Denote by $\mathcal{H}_{\mathrm{str}}(n)$ the set of structurally stable Hamiltonian flows in $\mathcal{H}(n)$. Then the set of p-unstable or h-unstable Hamiltonian flows forms an open dense subset of the difference $\mathcal{H}(n) - \mathcal{H}_{\mathrm{str}}(n)$.*

Note that the difference $\mathcal{H}(n) - \mathcal{H}_{\mathrm{str}}(n)$ in this lemma is the set of non-structurally–stable Hamiltonian flows, which are intermediate flows of non-trivial transitions. Thus, a generic transition between structurally stable Hamiltonian flows in $\mathcal{H}(n)$ is either p-unstable or h-unstable. We therefore define a *transition graph* of structurally stable Hamiltonian flows as the graph whose vertices are topological equivalence classes of structurally stable Hamiltonian flows and whose edges are topological equivalence classes of p-unstable or h-unstable Hamiltonian flows.

2.5. *Word representation for orbits*

Physically, genus elements can be realized as centers, point vortices, or periodic boundaries (i.e., periodic orbits on the boundary of the domain). As in [11], for the sake of simplicity, we regard genus elements as periodic boundaries because periodic orbits around a center or a point vortex are topologically indistinguishable from those around a periodic boundary. Therefore, we assume that each genus element is a periodic boundary. In other words, we make the following assumption ($A2$):

($A2$) There are no centers.

Assumption ($A2$) means that a non-degenerate singular point is either a saddle or a ∂-saddle. By the definitions of the terms in ($A1$) and ($A2$), a Hamiltonian flow in $\mathcal{H}_{\mathrm{bd}}(n)$ or $\mathcal{H}_{\mathrm{ubd}}(n)$ satisfies both ($A1$) and ($A2$) if and only if each singular point is a non-fake multi-saddle. We use $\mathcal{H}_{\mathrm{bd},0}(n)$ (respectively, $\mathcal{H}_{\mathrm{ubd},0}(n)$) to denote the set of flows with finitely many singular points in $\mathcal{H}_{\mathrm{bd}}(n)$ (respectively, $\mathcal{H}_{\mathrm{ubd}}(n)$) satisfying the assumptions for ($A1$) and ($A2$). Hereinafter, we assume that a Hamiltonian flow belongs to $\mathcal{H}_{\mathrm{bd},0}(n)$ or $\mathcal{H}_{\mathrm{ubd},0}(n)$. Thus, a Hamiltonian flow on a bounded domain means a Hamiltonian flow in a bounded domain $\mathbb{D}(n)$ (for some n) whose singular points are non-fake multi-saddles and centers, and a Hamiltonian flow on an unbounded domain means a Hamiltonian flow with a 1-source–sink at infinity in the punctured plane $\mathbb{U}(n)$ whose singular points are non-fake multi-saddles and centers.

It is known that any structurally stable Hamiltonian flows on an unbounded domain can be generated from a uniform flow (see Fig. 1) (i.e., a flow that is topologically equivalent to the flow generated by a vector field $(1, 0)$) on the plane $\mathbb{U}(0)$ by iteratively applying five operations, each of which replaces a saddle connection diagram by adding one genus element: a_0, a_2, b_0, b_2, or c (see Fig. 8) [11].[b] Indeed, operations a_0, b_0, and c (respectively, a_2, b_2) remove a point and insert a closed annulus (respectively, circle) as a set. Similarly, any structurally stable Hamiltonian flow on a bounded domain can be generated from a pointwise periodic flow (see Fig. 1) on the annulus $\mathbb{D}(1)$ by iteratively applying three operations, each of which replaces a saddle connection diagram by adding one genus element: b_0, b_2, or c.

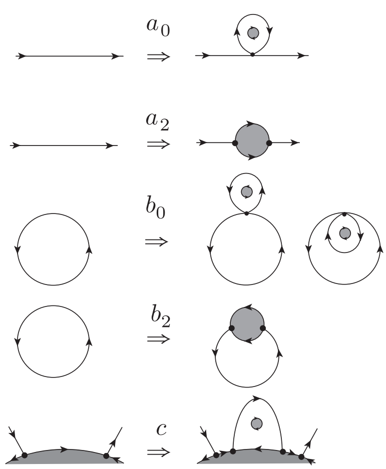


Fig. 8. The five fundamental operations a_0, a_2, b_0, b_2, and c.

[b]Following the convention in formal language theory, we write the letters in lower case letters. Moreover, to simplify words, we replace *I* (respectively, *II*) by a_2 (respectively, a_0) because *I* and a_2 (respectively, *II* and a_0) are equivalent as local structures.

To apply the five operations, we fix the outermost part. Indeed, the outermost part of an unbounded domain is the point at infinity, and the outermost part of a bounded domain is a boundary of an annulus. Recall the five operations a_0, a_2, b_0, b_2, and c that each adds one genus element:

(a_0) The operation a_0 replaces an ss-orbit by a saddle with two ss-separatrices and with a self-connected separatrix, as shown in the first depiction in Fig. 8.
(a_2) The operation a_2 replaces an ss-orbit by a boundary component that consists of two ∂-saddles and two separatrices, as shown in the second depiction in Fig. 8.
(b_0) The operation b_0 replaces a periodic orbit by a saddle with two self-connected separatrices, as shown in the third depiction in Fig. 8.
(b_2) The operation b_2 replaces a periodic orbit by a boundary component with a self-connected separatrix which consists of two ∂-saddles and two separatrices, as shown in the fourth depiction in Fig. 8.
(c) The operation c replaces a separatrix contained in the boundary by two ∂-saddles with four self-connected separatrices, as shown in the last depiction in Fig. 8.

Therefore, each of the operations a_0, a_2, b_0, b_2, and c adds a new saddle connection to the saddle connection diagram, and the operation c adds a self-connected separatrix between two ∂-saddles on the same boundary component. Each operation increments the number of genus elements by one. For example, a_2 adds a boundary component on an ss-orbit, and c adds a self-connected separatrix on the boundary component. Notice that operations are not always applicable; for instance, the operation c cannot be applicable if a_2 or b_2 has not been applied before it. In other words, each of the operations a_0, a_2, b_0, b_2, and c is a creation operation. On the other hands, we can define the inverse operation a_0^{-1} (respectively, a_2^{-1}, b_0^{-1}, b_2^{-1}, c^{-1}) of a_0 (respectively, a_2, b_0, b_2, c), which is an annihilation operation. Precisely, the five operations a_0^{-1}, a_2^{-1}, b_0^{-1}, b_2^{-1}, and c^{-1} that each deletes one genus element:

(a_0^{-1}) By the operation a_0^{-1}, a saddle with two ss-separatrices and with a self-connected separatrix is replaced by an ss-orbit, along the opposite direction of the arrow, as shown in the first depiction in Fig. 8.
(a_2^{-1}) By the operation a_2^{-1}, a boundary component that consists of two ∂-saddles and two separatrices is replaced by an ss-orbit, along the opposite direction of the arrow, as shown in the second depiction in Fig. 8.
(b_0^{-1}) By the operation b_0^{-1}, a saddle with two self-connected separatrices is replaced by a periodic orbit, along the opposite direction of the arrow, as shown in the third depiction in Fig. 8.
(b_2^{-1}) By the operation b_2^{-1}, a boundary component with a self-connected separatrix which consists of two ∂-saddles and two separatrices is replaced by a periodic orbit, along the opposite direction of the arrow, as shown in the fourth depiction in Fig. 8.

(c^{-1}) By the operation c^{-1}, two ∂-saddles with four self-connected separatrices contained in the boundary is replaced by a separatrix, along the opposite direction of the arrow, as shown in the last depiction in Fig. 8.

We say that a word for Hamiltonian flows is a sequence of the five operations a_0, a_2, b_0, b_2, and c. The length of a word is the number of letters it contains. For any positive integer n, each word of length n for Hamiltonian flows is a subset of structurally stable Hamiltonian flows in $\mathcal{H}_{\mathrm{bd},0}(n)$ or $\mathcal{H}_{\mathrm{ubd},0}(n)$. Indeed, a word $w_1 w_2 \cdots w_n$ for Hamiltonian flows is the set of structurally stable Hamiltonian flows generated by applying operations $w_1, w_2, \ldots, w_n$ in sequence. Therefore, a set of words for Hamiltonian flows is a finite set partially ordered with respect to the order of inclusion. We call that the maximal elements *maximal words*. In [11], as operation *I* corresponds to a_2 and operation *II* corresponds to a_0, the commutativity of a_0 and a_2 shown in [11, Table 1, Theorems 3.3–3.5] imply the following lemmas. Here, we denote empty word ϵ by x^0 and word xx^i by x^i $(i \geq 0)$, and we let $w(s,t,u) := (b_0)^s (b_2)^t (c)^u$.

Lemma 2.4. *Each structurally stable Hamiltonian flow in* $\mathcal{H}_{\mathrm{bd},0}(n)$ *belongs to a maximal word of the form*:

$$w(n_{b_0,1}, n_{b_2,1}, n_{c,1}) \cdots w(n_{b_0,k-1}, n_{b_2,k-1}, n_{c,k-1}) w(n_{b_0,k}, n_{b_2,k}, 0)$$

for some $k \in \mathbb{Z}_{\geq 1}$, $n_{b_0,i}, n_{b_2,i} \in \mathbb{Z}_{\geq 0} (1 \leq i \leq k)$ *and* $n_{c,i} \in \mathbb{Z}_{\geq 1} (1 \leq i \leq k-1)$ *such that* $n - 1 = \sum_{i=1}^{k-1} (n_{b_0,i} + n_{b_2,i} + n_{c,i}) + n_{b_0,k} + n_{b_2,k}$.

For instance, there are several maximal words for structurally stable Hamiltonian flows in bounded domains, as shown in Fig. 9. It should be noted that the maximal word is characterized by a regular expression $(b_0^* b_2^* c^+)^* b_0^* b_2^*$. (cf. the definition of the class of *regular expressions* in [7].)

Lemma 2.5. *Each structurally stable Hamiltonian flow in* $\mathcal{H}_{\mathrm{ubd},0}(n)$ *belongs to a maximal word of the form*:

$$(a_2)^{n_{a_2}} (a_0)^{n_{a_0}} w(n_{b_0,1}, n_{b_2,1}, n_{c,1}) \cdots w(n_{b_0,k-1}, n_{b_2,k-1}, n_{c,k-1}) w(n_{b_0,k}, n_{b_2,k}, 0)$$

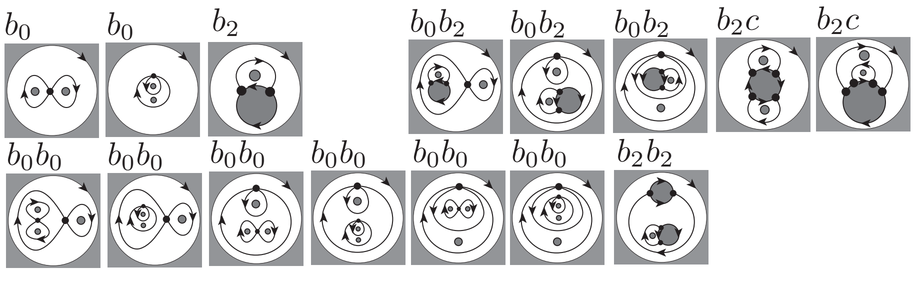


Fig. 9. Complete list of the structurally stable Hamiltonian flows on bounded domains with maximal words which are of length one or two.

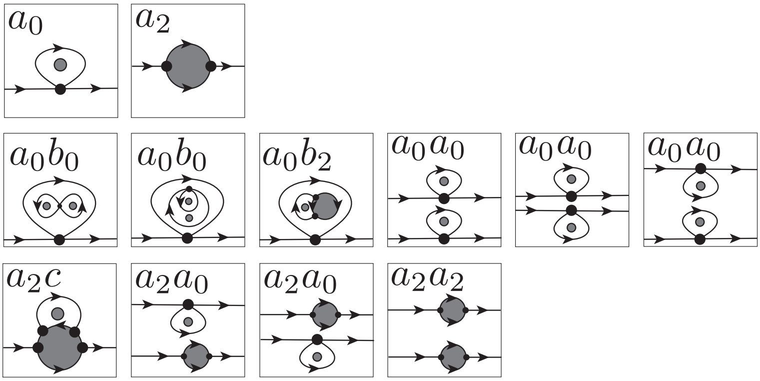


Fig. 10. Complete list of the structurally stable Hamiltonian flows on unbounded domains with maximal words which are of length one or two.

for some $k \in \mathbb{Z}_{\geq 1}$, $n_{a_2}, n_{a_0}, n_{b_0,i}, n_{b_2,i} \in \mathbb{Z}_{\geq 0}$ $(1 \leq i \leq k)$, *and* $n_{c,i} \in \mathbb{Z}_{\geq 1}$ $(1 \leq i \leq k-1)$ *such that* $n = n_{a_2} + n_{a_0} + \sum_{i=1}^{k-1}(n_{b_0,i} + n_{b_2,i} + n_{c,i}) + n_{b_0,k} + n_{b_2,k}$.

For instance, there are several maximal words for structurally stable Hamiltonian flows in unbounded domains, as shown in Fig. 10. It should be noted that the maximal word is characterized by a regular expression $a_2^* a_0^* (b_0^* b_2^* c^+)^* b_0^* b_2^*$.

Recall that an a_0 (respectively, a_2, c) structure is innermost if it is as shown in the first and second depictions (respectively, third depiction, fourth depiction) in Fig. 11.

It is necessary to represent each structurally stable Hamiltonian flow as a unique maximal word to facilitate the computational analysis of such flows. Therefore, we introduce Algorithms 1 and 2, which are variations of the algorithm in [9]; here, *I* and *II* are replaced with a_2 and a_0, respectively, and the auxiliary common procedure Delete is factored out.

Notice that there are several options of algorithms to use for computing the same function. Moreover, if the algorithm is changed, then so is the assignment of a structurally stable Hamiltonian flow to a maximal word in general. Although Algorithms 1 and 2 return words that are maximal but that generally differ from those returned by the algorithm in [9], these algorithms assign a structurally stable Hamiltonian flow to a unique maximal word such that the differences in the maximal

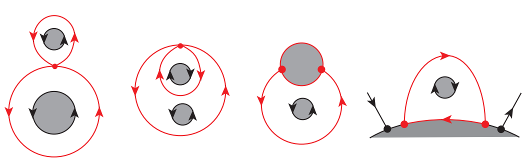


Fig. 11. List of the innermost orbit structures. From left to right: two innermost a_0 structures, an innermost a_2 structure, and an innermost c structure.

Algorithm 1. Construct a maximal word of the flow generated by the saddle connection diagram on a bounded domain (a_0 and a_2 are not contained).

Input: D: a saddle connection diagram on a bounded domain.
Output: w: a maximal word of the flow generated by D.

```
procedure ALGORITHM1(D)
    w ← []
    while D has a b_2, b_0, or c structure do
        DELETE([b_2, b_0, c], D, w)
    return w
```

Algorithm 2. Construct a maximal word of the flow generated by the saddle connection diagram on an unbounded domain (a_0 and a_2 may be contained).

Input: D: a saddle connection diagram on an unbounded domain.
Output: w: a maximal word of the flow generated by D.

```
procedure ALGORITHM2(D)
    w ← ALGORITHM1(D)
    DELETE([a_2, a_0], D, w)
    return w
```

words by the original algorithms and the new algorithms are in the positions of b_2 letters. The difference between the two new algorithms is their domains: Algorithm 1 does not accept trees containing a_0 and a_2, but Algorithm 2 does. Algorithm 2 is more general than Algorithm 1; if we let f_1 and f_2 be the functions computed by Algorithms 1 and 2, respectively, we have $f_1 \subset f_2$.

Recall that the saddle connection diagram is the union of saddles, ∂-saddles, and separatrices. The set difference is well defined on the saddle connection diagram D by the following procedure. The auxiliary procedure $\textsc{Delete}(xs, D, w)$ iteratively deletes the deletable structures corresponding to the operations xs one by one from the beginning and adds each corresponding operation to the front of a list w. Recall that structure a_0 is a homoclinic saddle connection with two ss-separatrices (as shown in Fig. 5(a_0)) and structure a_2 is two separatrices on a boundary with two ss-separatrices (as shown in Fig. 5(a_2)). Structures a_0 and a_2 are always *deletable* if they do not contain other structures within them. Structure b_0 is a bounded homoclinic saddle connection (as shown in Fig. 5(b_0)), and structure c is two ∂-saddles with a separatrix between them on a boundary (as shown in Fig. 5(c)). Structures b_0 and c are *deletable* if they are innermost, as shown in Fig. 11. When structure c is deleted, it can be seen that two ∂-saddles have been merged and then the resulting fake ∂-saddle has been deleted. The deletion of structure c is the inverse of operation c. Structure b_2 (see Fig. 5(b_2)) is *deletable* if it has a boundary with just two ∂-saddles connected by a separatrix. Ignoring orbits outside of the saddle connection diagram, the deletions of saddles, ∂-saddles, self-connected saddles, and self-connected ∂-saddles are the inverses of operations a_0, a_2, b_0, and c, respectively (cf. Fig. 8).

Algorithm 1 takes the saddle connection diagram on a bounded domain D and returns one of its maximal words. The word w is represented by a list of the five fundamental operations a_0, a_2, b_0, b_2, and c shown in Fig. 8. At line 2, w is initialized as an empty list $[\,]$, meaning that it is an empty word.

The call $\textsc{Delete}([x_1, x_2, \ldots, x_n], D, w)$ deletes as many of the structures of $x_1, x_2, \ldots, x_n$ as possible from D sequentially from the beginning. Sometimes, a structure x_i cannot be taken from the saddle connection diagram D even if the structure x_i remains. For example, if the structure is b_2 containing a boundary with more than two ∂-saddles connected by a separatrix, we cannot immediately delete structure b_2.

Here, the parameters are passed by reference. Hence, when procedure call $\textsc{Delete}(xs, D, w)$ is executed in Algorithms 1 and 2, the change made to D (respectively, w) in line 4 (respectively, 5) in the body of Algorithm 3 affects the actual argument D (respectively, w) that is passed.

By b_0 *structure*, we mean a saddle with two self-connected separatrices in the saddle connection diagram, as shown in the third depiction in Fig. 8. By b_2 *structure*, we mean a boundary component with a self-connected separatrix that consists of two ∂-saddles and two separatrices in the saddle connection diagram, as shown in the fourth depiction in Fig. 8. By c *structure*, we mean a circle in the saddle connection diagram that is a union of two ∂-saddles and two self-connected separatrices between them, as shown in the last depiction in Fig. 8.

A single call $\textsc{Delete}([b_2, b_0, c], D, w)$ might not delete all b_2, b_0, and c structures. It is not possible to delete all of the single structures in a single saddle connection diagram in the case in which there is no deletable structure, that is, when b_0 or c contains other structures within it and b_2 has more than a c structure on its boundary. In Algorithm 1, $\textsc{Delete}([b_2, b_0, c], D, w)$ is called as long as D has a b_2, b_0, or c structure. This iteration deletes as many b_2 structures as possible. Then, any remaining b_2 structure in the resulting saddle connection diagram will have more than a c structure on its boundary. Hence, in order to delete this b_2 structure, we

---

Algorithm 3. Delete the structure corresponding to one element of xs in the saddle connection diagram, and construct a maximal word.

Input: xs: a list of operations; D: the saddle connection diagram; w: a word to be made maximal.

1: **procedure** $\textsc{Delete}(xs, D, w)$
2:     **for** $x \in xs$ **do**
3:         **while** D has a deletable x structure **do**
4:             delete its saddle connection from D
5:             $w \leftarrow x \cdot w$
6:     **end for**

---

first need to previously delete the c structure because the inverse operation deleting the b_0 structure does not create any b_2 structures. Note that this deletion order results in the maximal word.

Algorithms 1 and 2 are *determinate*, meaning that the same input saddle connection diagrams always produce the same maximal words. This means that the semantics of these algorithms are functions. However, they are not *deterministic* because the same deletable structure may appear in more than one place at times. Therefore, we have the following well-definedness.

Lemma 2.6 (Determination of Algorithms 1 and 2). *Algorithm* 1 *computes a function, and Algorithm* 2 *computes a function.*

In procedure Delete, each iteration of the "for" loop (lines 2–6) sets x to be an element in xs from the beginning to the end. The inner "while" loop deletes as many x structures as possible. Because the number of elements in xs and the number of x structures in D are finite, procedure Delete(xs, D, w) is terminating for any xs and D. This implies the following statement.

Lemma 2.7 (Termination of Delete). *For any list of operations xs, any saddle connection diagram D, and any word w, the procedure call* Delete(xs, D, w) *is terminating.*

This lemma leads to a statement regarding the guarantee of the termination of Algorithms 1 and 2.

Lemma 2.8 (Termination of Algorithm1 and Algorithm2). *For any saddle connection diagram D on a bounded domain, the procedure call* Algorithm1(D) *is terminating. For any saddle connection diagram D on an unbounded domain, the procedure call* Algorithm2(D) *is terminating.*

The proofs for the termination of these algorithms are trivial but tedious, and we simply mention their outline. For any inputs, Algorithm 1 is terminating because each iteration of its "while" loop decrements the number of b_2, b_0, and c structures in D, and Delete is terminating. For any inputs, Algorithm 2 is terminating because Algorithm 1 is terminating and Delete is terminating. Algorithms 1 and 2 return one of the maximal words for any input. In other words, we have the following correctness.

Lemma 2.9 (Correctness of Algorithm1 and Algorithm2). *For any saddle connection diagram D on a bounded domain, the procedure call* Algorithm1(D) *returns a unique maximal word. For any saddle connection diagram D on an unbounded domain, the procedure call* Algorithm2(D) *returns a unique maximal word.*

Proof. As is mentioned above, after deleting as many b_2 structures as possible because the inverse operation deleting the b_0 structure does not create any

b_2 structures, we first need to delete the c structure. This implies that ALGORITHM1(D) (respectively, ALGORITHM2(D)) makes a word in the standard form given in Lemma 2.4 (respectively, Lemma 2.5). □

In summary, we can characterize the two algorithms by the following lemmas. Denote by $\mathcal{H}_{\mathrm{bd,str}}(n)$ the set of structurally stable Hamiltonian flows in $\mathcal{H}_{\mathrm{bd}}(n)$, and by $\mathcal{H}_{\mathrm{ubd,str}}(n)$ the set of structurally stable Hamiltonian flows in $\mathcal{H}_{\mathrm{ubd}}(n)$. The algorithms previously given imply the following statements.

Lemma 2.10. *There is a recursive injective function from the set of topological equivalence classes of* $\mathcal{H}_{\mathrm{bd,str}}(n)$ *to the set of maximal words consisting of the three operations* b_0, b_2, *and* c.

Lemma 2.11. *There is a recursive injective function from the set of topological equivalence classes of* $\mathcal{H}_{\mathrm{ubd,str}}(n)$ *to the set of maximal words consisting of the five operations* a_0, a_2, b_0, b_2, *and* c.

2.6. *Transitions of structurally stable Hamiltonian flows*

To facilitate reasoning and the analysis of the algorithms, we list all generic transitions among structurally stable Hamiltonian flows.

For a word $w = w_1 w_2 \cdots w_n$ and a non-negative integer $k \leq n$, we say that $w_{i_1} w_{i_2} \cdots w_{i_k}$ is a subword of w if $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. As mentioned after Lemma 2.3, all generic transitions between structurally stable Hamiltonian flows in $\mathcal{H}(n)$ are either p-unstable or h-unstable.

Recall from [10] that there are 17 operations D_0, $D_{0,s}$, D_2, $D_{2,s}$, E_0, $E_{0,s}$, E_2, $E_{2,s}$, Φ_0, $\Phi_{0,s}$, Λ, Δ_1, $M_{1,i}$, $M_{1,o}$, $M_{1,s}$, $\Psi(2)$, $\Xi(2n-1)$, $\Xi_s(2n-1)$, shown in Fig. 12, the first 14 operations of which add one genus element, and the other 3 of which add more than one genus element, with the counts given in parentheses. These operations are described as follows:

$(D_0, D_{0,s})$	The operation D_0 (respectively, $D_{0,s}$) adds both a saddle and its self-connected separatrix to a self-connected separatrix of a ∂-saddle (respectively, non-self-connected separatrix).
$(D_2, D_{2,s})$	The operation D_2 (respectively, $D_{2,s}$) adds a boundary consisting of two ∂-saddles and of two self-connected separatrices to a self-connected separatrix of a ∂-saddle (respectively, non-self-connected separatrix).
$(E_0, E_{0,s})$	The operation E_0 (respectively, $E_{0,s}$) adds both a saddle and its self-connected separatrix to a self-connected separatrix of a saddle on (respectively, outside of) the boundary of the set of ss-orbits, as shown in Fig. 12.

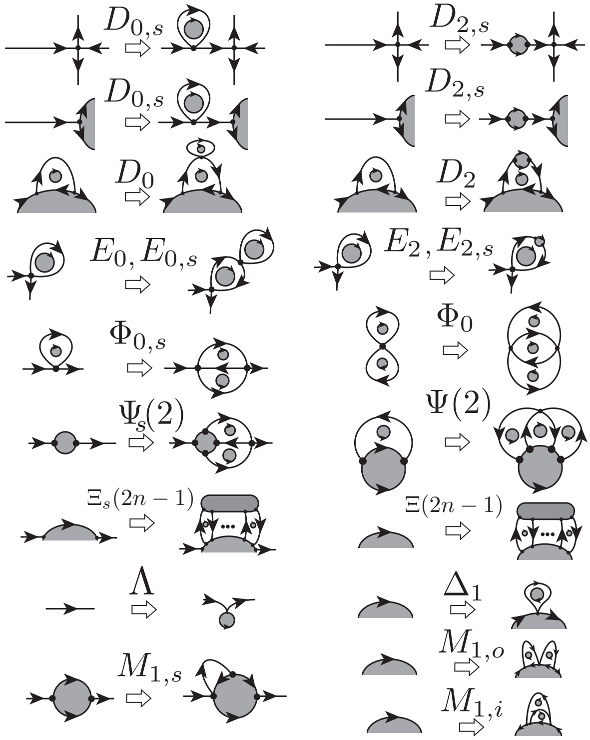


Fig. 12. The 17 generic transition operations.

($E_2, E_{2,s}$) The operation E_2 (respectively, $E_{2,s}$) adds a boundary consisting of two ∂-saddles and of two self-connected separatrices to a self-connected separatrix of a saddle on (respectively, outside of) the boundary of the set of ss-orbits, as shown in Fig. 12.

(Φ_0, $\Phi_{0,s}$) The operation Φ_0 (respectively, $\Phi_{0,s}$) splits a saddle into two saddles and two non-self-connected separatrices whose union encloses a genus element, as shown in Fig. 12.

($\Psi(2)$) The operation $\Psi(2)$ adds two boundary components, two ∂-saddles, and a saddle with three separatrices among them, as shown in Fig. 12.

($\Xi(2n-1)$, $\Xi_s(2n-1)$) The operations $\Xi(2n-1)$ and $\Xi_s(2n-1)$ add any positive odd numbers $2n-1$ of boundary components and separatrices, as shown in Fig. 12.

(Δ_1) The operation Δ_1 attaches a pinching with a separatrix from and to it enclosing a boundary, as shown in Fig. 12.

($M_{1,i}, M_{1,o}$, $M_{1,s}$) The operations $M_{1,i}$, $M_{1,o}$, and $M_{1,s}$ add a separatrix to a boundary one of whose end points is attached to a ∂-saddle, as shown in Fig. 12.

Because all local transition rules are illustrated in Fig. 12, by calculating all possible combinations of saddle connections, we can list all transitions among maximal patterns of the same length as follows.

Lemma 2.12 ([10, Proposition 6.1]). *An h-unstable Hamiltonian flow has exactly one unstable saddle connection* (*as shown in Fig.* 13) *which is generated by a subword in Table* 1. *A p-unstable Hamiltonian flow has exactly one unstable saddle connection* (*as in Fig.* 13) *which is generated by a subword in Table* 1.

This lemma says that the complete list of generic transitions among structurally stable Hamiltonian flows is given in Fig. 13 up to mirror image and time-reversion as a directed graph on the sphere.

Let $\mathcal{H}(n)$ be either the set of flows in $\mathcal{H}_{\mathrm{bd}}(n)$ satisfying condition $(A1)$ or the set of flows in $\mathcal{H}_{\mathrm{ubd}}(n)$ satisfying condition $(A1)$.

Recall that the set of p-unstable or h-unstable Hamiltonian flows forms an open dense subset of the set of non-structurally–stable Hamiltonian flows. Let $\mathcal{H}_{\mathrm{bd},1}(n)$ (respectively, $\mathcal{H}_{\mathrm{ubd},1}(n)$) denote the set of p-unstable or h-unstable Hamiltonian flows on a bounded (respectively, an unbounded) domain. The structures D_0, $D_{0,s}$, D_2, $D_{2,s}$, E_0, $E_{0,s}$, E_2, $E_{2,s}$, Φ_0, $\Phi_{0,s}$, Λ, Δ_1, $M_{1,i}$, $M_{1,o}$, $M_{1,s}$, $\Psi(2)$, $\Xi(2n-1)$, and $\Xi_s(2n-1)$ are *deletable* if they are innermost, as shown in Fig. 12. In Algorithm 1, replacing "b_0, b_2, or c" by "b_0, b_2, c, D_0, $D_{0,s}$, D_2, $D_{2,s}$, E_0, $E_{0,s}$, E_2, $E_{2,s}$, Φ_0, $\Phi_{0,s}$, Λ, Δ_1, $M_{1,i}$, $M_{1,o}$, $M_{1,s}$, $\Psi(2)$, $\Xi(2n-1)$, or $\Xi_s(2n-1)$" and replacing $[b_2, b_0, c]$

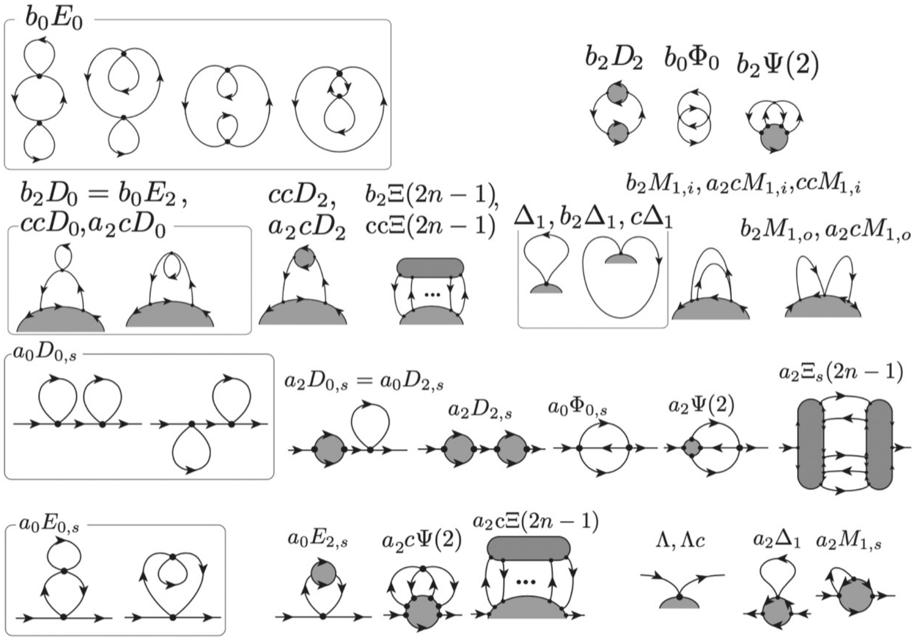


Fig. 13. List of the generic transitions among structurally stable Hamiltonian flows.

Table 1. List of subwords corresponding to p-unstable or h-unstable flows which can be applied under the given preconditions and the postconditions of subwords.

No.	Transition rule	Precondition	Postcondition
1	b_0E_0	b_0b_0	Same as the precondition
	$b_0\Phi_0$	b_0b_0	Same as the precondition
	$b_2D_0 = b_0E_2$	b_0b_2	Same as the precondition
	b_2D_2	b_2b_2	Same as the precondition
	$b_2\Psi(2)$	$b_0b_2c, b_2b_0c, b_2cb_0$	Same as the precondition
	$b_2\Xi(2n-1)$	$c^kb_2c^{2n-2-k}$	Same as the precondition
	ccD_0	cb_0c, ccb_0	Same as the precondition
	ccD_2	ccb_2, cb_2c	Same as the precondition
	$cc\Xi(2n-1)$	$c^kb_2c^{2n-k}$	Same as the precondition
2	$b_2M_{1,i}$	b_0b_2, b_2b_0	b_2c
	$b_2M_{1,o}$	b_0b_2, b_2b_0	b_2c
	$b_2\Delta_1$	b_0b_2, b_2b_0	b_2c
	$ccM_{1,i}$	cb_0c, ccb_0	ccc
	$c\Delta_1$	cb_0	cc
	Δ_1	b_0	b_2
3	$a_0D_{0,s}$	a_0a_0	Same as the precondition
	$a_0\Phi_{0,s}$	a_0a_0	Same as the precondition
	$a_2D_{0,s} = a_0D_{2,s}$	a_2a_0	Same as the precondition
	$a_2D_{2,s}$	a_2a_2	Same as the precondition
	$a_2\Psi(2)$	a_2a_0c	Same as the precondition
	$a_2\Xi_s(2n-1)$	$a_2a_2c^{2n-2}$	Same as the precondition
4	$a_0E_{0,s}$	a_0a_0	a_0b_0
	$a_0E_{2,s}$	a_2a_0	a_0b_2
	a_2cD_0	a_2a_0c	a_2cb_0
	a_2cD_2	a_2a_2c	a_2cb_2
	$a_2c\Psi(2)$	a_2cb_0c, a_2ccb_0	a_2a_0cc
	$a_2c\Xi(2n-1)$	$a_2a_2c^{2n-1}$	$a_2c^kb_2c^{2n-1-k}$
5	Λ	a_0	a_2
	$a_2cM_{1,i}$	a_2a_0c	a_2cc
	$a_2cM_{1,o}$	a_2cb_0	a_2cc
	$a_2M_{1,s}$	a_2a_0	a_2c
	$a_2\Delta_1$	a_2a_0	a_2c
	Λc	a_0b_2	a_2c

by [b_0, b_2, c, D_0, $D_{0,s}$, D_2, $D_{2,s}$, E_0, $E_{0,s}$, E_2, $E_{2,s}$, Φ_0, $\Phi_{0,s}$, Λ, Δ_1, $M_{1,i}$, $M_{1,o}$, $M_{1,s}$, $\Psi(2)$, $\Xi(2n-1)$, $\Xi_s(2n-1)$], summarizing previous lemmas, we obtain the following statements.

Proposition 2.13. *There is a recursive injective function from the set of topological equivalence classes of $\mathcal{H}_{\mathrm{bd},1}(n)$ to the set of maximal words consisting of the three operations b_0, b_2, and c except for one operation D_0, D_2, E_0, E_2, Φ_0, Λ, Δ_1, $M_{1,i}$, $M_{1,o}$, $\Psi(2)$, or $\Xi(2n-1)$.*

Proposition 2.14. *There is a recursive injective functions from the set of topological equivalence classes of $\mathcal{H}_{\mathrm{ubd},1}(n)$ to the set of maximal words consisting of the five operations a_0, a_2, b_0, b_2, and c except for D_0, $D_{0,s}$, D_2, $D_{2,s}$, E_0, $E_{0,s}$, E_2, $E_{2,s}$, Φ_0, $\Phi_{0,s}$, Λ, Δ_1, $M_{1,i}$, $M_{1,o}$, $M_{1,s}$, $\Psi(2)$, $\Xi(2n-1)$, or $\Xi_s(2n-1)$.*

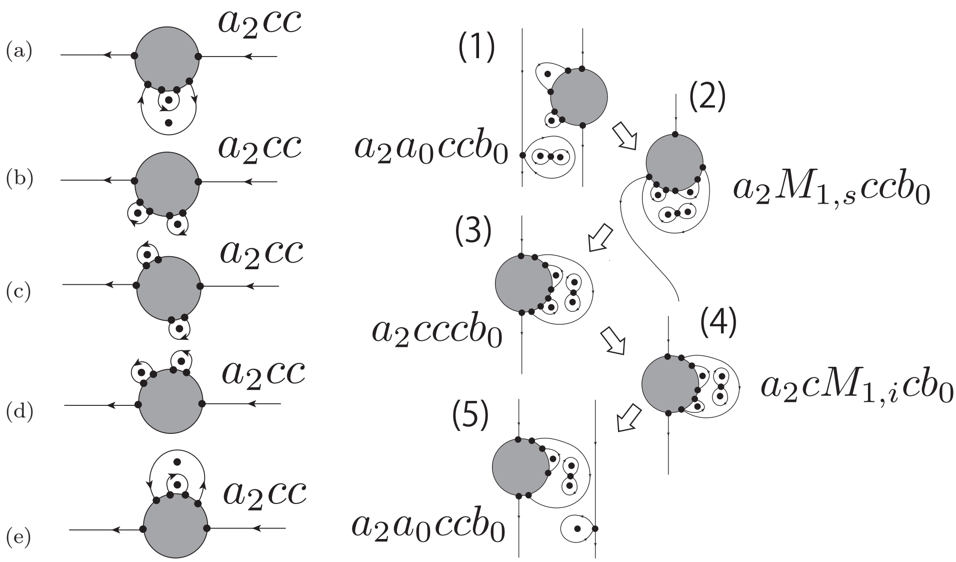


Fig. 14. Left: a maximal word a_2cc is a set of the five Hamiltonian flows (a)–(e); right: three Hamiltonian flows (1), (3), (5) and the two intermediate Hamiltonian flows (2), (4).

These propositions are not explicitly stated but are implicitly given in [10]. We have the following example. Applying a_2 once and c twice to a uniform flow, we obtain the five flows in Fig. 14(a)–14(e).

Namely, the word a_2cc, which is a sequence of operations, is a set of the five flows. Stable Hamiltonian flows are changed by large perturbations. Fig. 14(1)–(5) shows examples of transitions of structurally stable Hamiltonian flows (1), (3), (5) via unstable ones (2), (4). There is just one direct transition from (1) to (3), which must go through (2). Using word representation, we can easily check whether transitions of flows by a numerical simulation are *not incorrect* from a topological point of view. For instance, the transition of flows from (1) to (2) via a single unstable flow *may* occur because the word $a_2a_0ccb_0$ of (1) can be transformed to the word a_2cccb_0 of (3) by a single transition rule.

2.7. *Creations and annihilations of structurally stable Hamiltonian flows*

In Sec. 2.6, we deal with transitions without creations and annihilations of structurally stable Hamiltonian flows. However, creations and annihilations do appear in time evaluations of Hamiltonian flows. The application of (creation) operation x to a word w adds the letter x to word w. For a creation operation x, application of the annihilation operation denoted by x^{-1} removes the letter x from word w. Creation and annihilation operations are applicable only under certain conditions.

Data analysis of time evaluations has been widely studied for its applications in engineering and medical science. Topological data analysis (TDA) is rapidly spreading as a new method for investigations in such areas. For example, to compare certain materials, a typical TDA method measures distances. Similarly, with topological flow data analysis, measuring topological distances between flows seems to be an effective tool for comparing fluid phenomena. For this reason, in the following section we will define topological distance for topological equivalence classes. There are two conflicting requirements on this measure: precision and efficiency. Because of the trade-off, it is important to define several meaningful distances with different levels of scaling precision that can be processed with different levels of efficiency.

3. Vector Representation for Topological Flow

For some applications, we may require a smaller data representation that is smaller than word representation because the computational resources are limited or because information is required for the analysis. The natural number of genus elements n_{all} is a more abstract representation, in which case creation and annihilation of genus elements are represented by increments and decrements of one. The distance between two non-negative integer representations is the difference between the two non-negative integers. Obviously, creation is always possible, whereas annihilation is possible only when $n_{\mathrm{all}} \geq 1$.

The vectors of the counts of a_0, a_2, b_0, b_2, and c, i.e., $(n_{a_0}, n_{a_2}, n_{b_0}, n_{b_2}, n_c)$, are another representation. Here, n_x represents the number of x operations. The vector representation is a refinement of the single natural number representation in the sense that $n_{\mathrm{all}} = n_{a_0} + n_{a_2} + n_{b_0} + n_{b_2} + n_c$. The sum of all of the vector elements of the vectors must be greater than or equal to one. Vector representation is useful when the number of genus elements frequently changes. In the vectors of size one, the count of a_0, a_2, b_0, or b_2 is one, but the count of c must be zero. The creation or annihilation of a genus corresponds to an increment or decrement, respectively, of an element in the vector, respectively. We list all forbidden creations and annihilations.

Proposition 3.1. *Let $n(v) := (n_{a_0}, n_{a_2}, n_{b_0}, n_{b_2}, n_c) \in (\mathbb{Z}_{\geq 0})^5$ be a non-zero vector of the counts of operations a_0, a_2, b_0, b_2, and c. Denote by $\leq$ the product partial order on $(\mathbb{Z}_{\geq 0} \sqcup \{\infty\})^5$. Creation is possible in any case except the following*:

- $(c1)$ *When* $n_{a_2} = n_{b_2} = 0$ (i.e., $n(v) \leq (\infty, 0, \infty, 0, \infty)$), *operation c is not applicable.*
- $(c2)$ *When* $n_{a_0} = n_{a_2} = 0$ (i.e., $n(v) \leq (0, 0, \infty, \infty, \infty)$), *neither operation a_0 nor operation a_2 is applicable.*
- $(c3)$ *When* $n_{a_0} = n_c = 0$ *and* $n_{a_2} \neq 0$ (i.e., $(0, 1, 0, 0, 0) \leq n(v) \leq (0, \infty, \infty, \infty, 0)$), *neither operation b_0 nor operation b_2 is applicable.*

Annihilation is possible in any case except the following:

($a1$) *When* $n_c > 0$, $n_{a_2} = 1$, *and* $n_{b_2} = 0$ (i.e., $(0,1,0,0,1) \leq n(v) \leq (\infty,1,\infty,0,\infty)$), *the annihilation operation* a_2^{-1} *is not applicable.*
($a2$) *When* $n_c > 0$, $n_{a_2} = 0$, *and* $n_{b_2} = 1$ (i.e., $(0,0,0,1,1) \leq n(v) \leq (\infty,0,\infty,1,\infty)$), *the annihilation operation* b_2^{-1} *is not applicable.*
($a3$) *When* $n_{b_0} > 0$, $n_{a_0} = 1$, *and* $n_c = 0$ (i.e., $(1,0,1,0,0) \leq n(v) \leq (1,\infty,\infty,\infty,0)$), *the annihilation operation* a_0^{-1} *is not applicable.*
($a4$) *When* $n_{b_2} > 0$, $n_{a_0} = 1$, *and* $n_c = 0$ (i.e., $(1,0,0,1,0) \leq n(v) \leq (1,\infty,\infty,\infty,0)$), *the annihilation operation* a_0^{-1} *is not applicable.*
($a5$) *When* $n_{a_2} > 0$, $n_{b_0} > 0$, $n_{a_0} = 0$, *and* $n_c = 1$ (i.e., $(0,1,1,0,1) \leq n(v) \leq (0,\infty,\infty,\infty,1)$), *the annihilation operation* c^{-1} *is not applicable.*
($a6$) *When* $n_{a_2} > 0$, $n_{b_2} > 0$, $n_{a_0} = 0$, *and* $n_c = 1$ (i.e., $(0,1,0,1,1) \leq n(v) \leq (0,\infty,\infty,\infty,1)$), *the annihilation operation* c^{-1} *is not applicable.*

Proof. Notice that all creations are listed in Fig. 8 and that each creation is denoted by a_0, a_2, b_0, b_2, or c. Because the existence of inner boundaries is required to apply operation c, the exception for case ($c1$) follows. Operations a_0 and a_2 can be applied to unbounded orbits, and so the exception for case ($c2$) follows. Operations b_0 and b_2 can be applied to bounded orbits, and so the exception for case ($c3$) follows. Because operation c must be applied to a boundary, the exceptions for cases ($a1$) and ($a2$) follow. Operations b_0 and b_2 must be applied to bounded domains, and so the exceptions for cases ($a3$)–($a6$) follow. □

We can naively define the distance between two vector representations as the sum of all of the elements of the difference between the two vectors. However, this naive distance is not a lower bound on the graph geodesic distance (i.e., the path length) on the transition graph of the flows, and so in Sec. 5 we will define a distance more adapted to the transition operation.

4. Transition Diagrams

For any generic Hamiltonian flow with a few heteroclinic orbits, we list all generic transitions of Hamiltonian flows and construct a complete transition graph to automatically check the correctness as demonstrated in Sec. 2.5.

A *subword* of a given word is the given word with zero or more letters left out. In other words, a word $P = v_{p_1}v_{p_2}\cdots v_{p_k}$ is a subword of a word $V = v_1v_2\cdots v_n$ if $1 \leq p_1 < p_2 < \cdots < p_k \leq n$. We consider the following word search problem: given a pair of maximal words P and W, to determine whether P is a subword of W. We call the maximal word P for a word search problem a *pattern*. Patterns are compared by the inclusion of the sets of all maximal words that the patterns have as a subword. A pattern P is called *maximal* if there is no pattern that is greater than

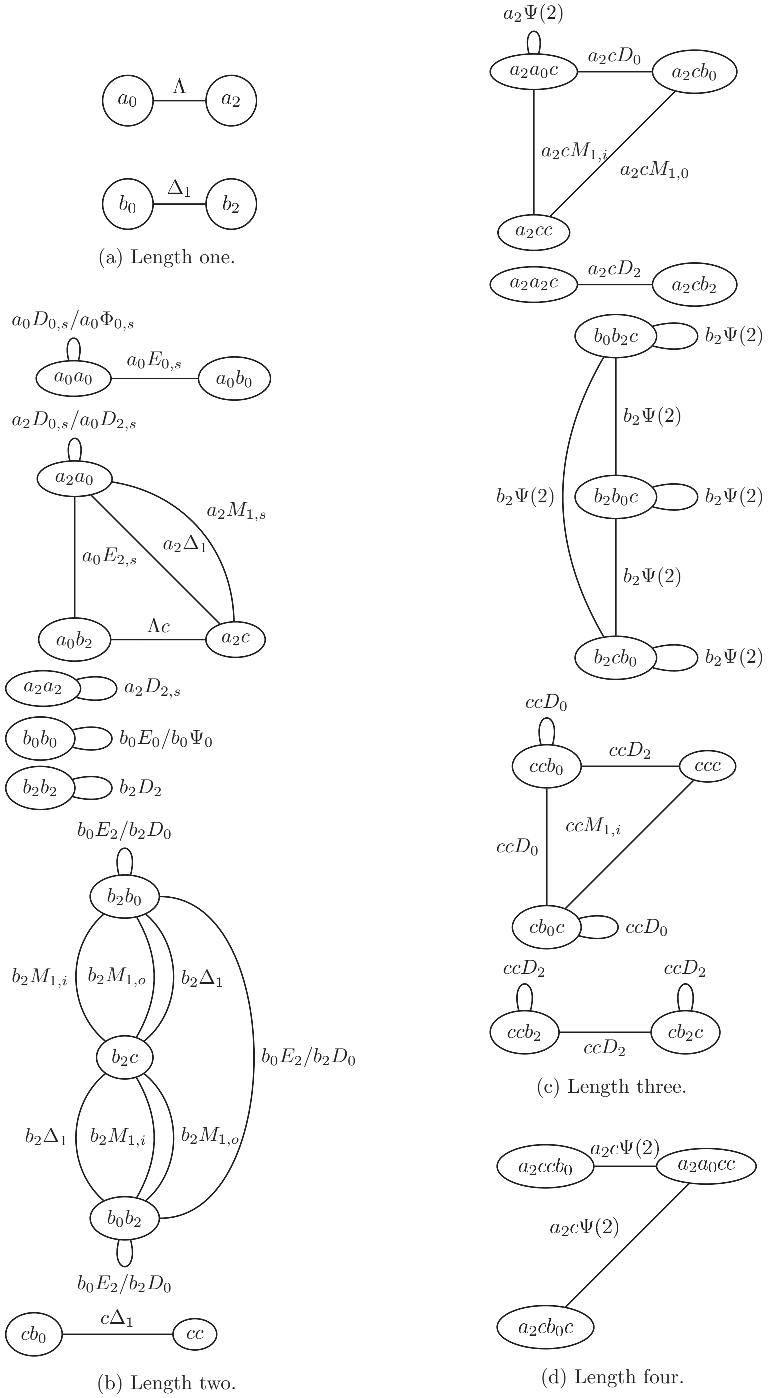
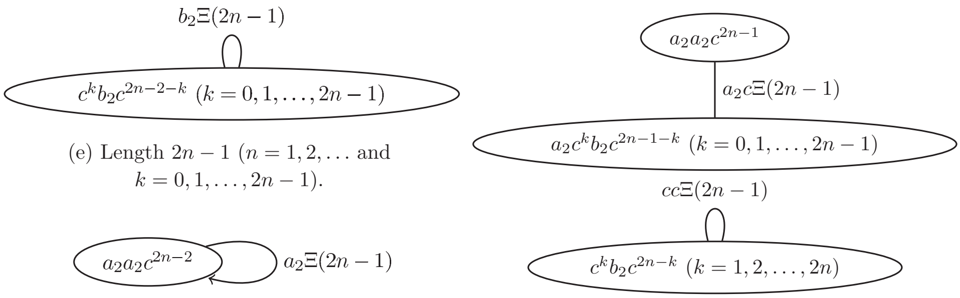


Fig. 15. Complete transition graphs for all Hamiltonian surface flow maximal patterns of each fixed length (1–4).



(e) Length $2n-1$ ($n = 1, 2, \ldots$ and $k = 0, 1, \ldots, 2n-1$).

(f) Length $2n$ ($n = 1, 2, \ldots$).

(g) Length $2n+1$ ($n = 1, 2, \ldots$).

Fig. 16. Complete transition graphs for all Hamiltonian surface flow maximal patterns of each variable length.

P. Figures 15 and 16 show the essential transition graphs for maximal patterns of the same length.

Proposition 4.1. *The complete transition graphs for maximal patterns of the same length are shown in Figs.* 15 *and* 16.

Proof. All local transition rules are illustrated in Fig. 12, and the list of all transitions among maximal patterns of the same length are illustrated in Fig. 13 and described in Table 1. Therefore, we can easily check that the transition graphs shown in Figs. 15 and 16 are complete. □

Each edge is associated with the name of a transition operation. A relation $V \to_X W$ is a transition from a maximal word that has V as a subword to another maximal word that has W as a subword by the operation X. A relation $V \to_X W$ holds if and only if there is an edge with the operation X between the patterns P and Q, where the pattern $P = v_{p_1} v_{p_2} \cdots v_{p_k}$ is a subword of the word $V = v_1 v_2 \cdots v_n$, the pattern $Q = w_{q_1} w_{q_2} \cdots w_{q_k}$ is a subword of the word $W = w_1 w_2 \cdots w_n$, and the maximal word of the word which is obtained by replacing v_{p_i} in V with w_{q_i} for $i = 1, 2, \ldots, k$ is W. For example, we have $a_2 a_0 a_0 \to_{a_0 E_{2,s}} a_0 a_0 b_2$ because replacing $a_2 a_0$ with $a_0 b_2$ in $a_2 a_0 a_0$ becomes $a_0 b_2 a_0$, and its maximal word is $a_0 a_0 b_2$.

We define a transition graph of maximal words of the same length of structurally stable Hamiltonian flows as the graph whose vertices are maximal words and whose edges connect maximal words if there are transitions between them by operations. Notice that the list of transitions among structurally stable Hamiltonian flows with maximal words of the same length is finite, and so the transition graph can be algorithmically computed. In particular, when for readability we omit loops and edges between maximal patterns having the same vector representations, we obtain complete transition graphs for all maximal words of length three (Fig. 17) and length four (Figs. 18 and 19).

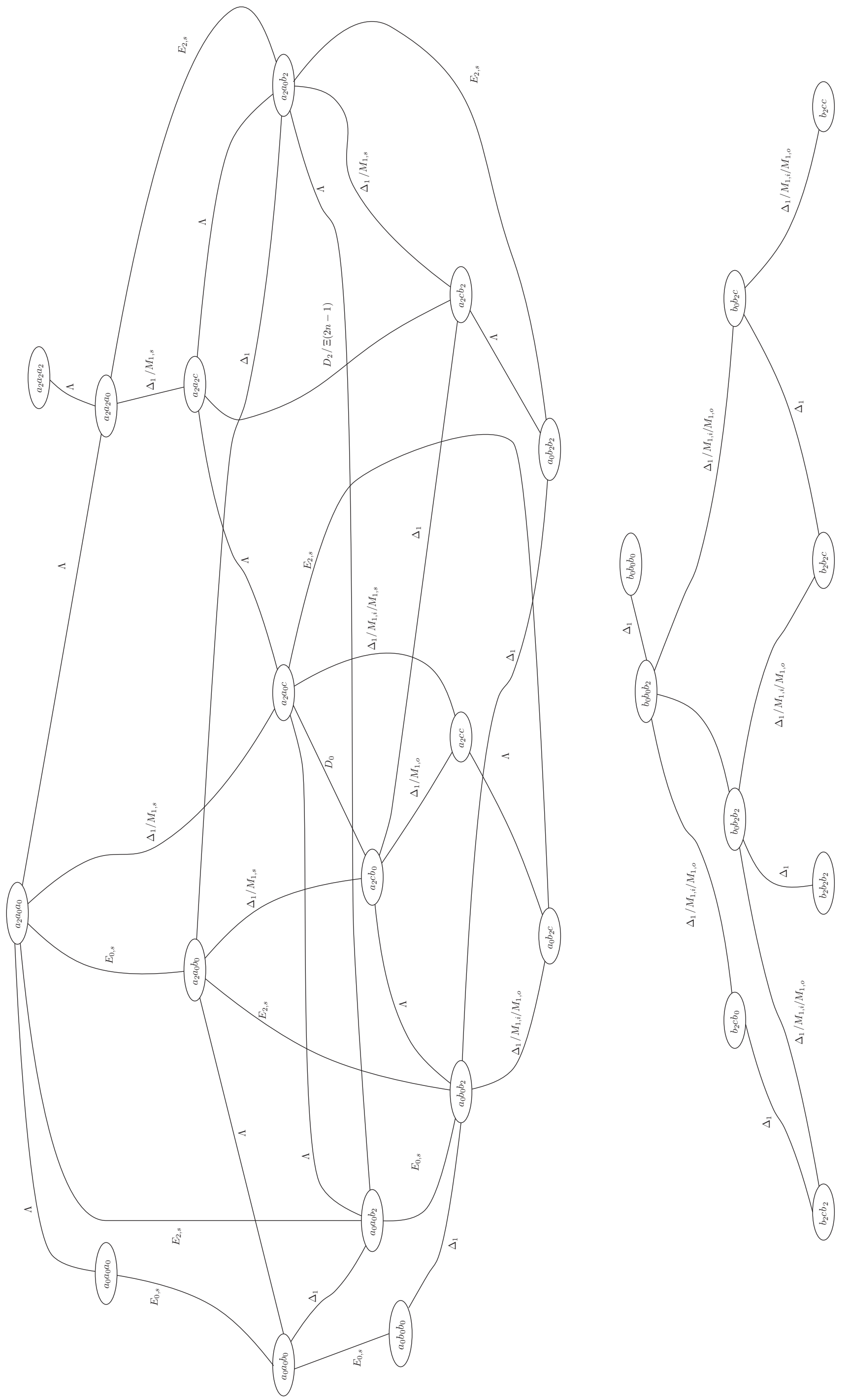


Fig. 17. Complete transition graph of Hamiltonian surface flows of maximal word length three.

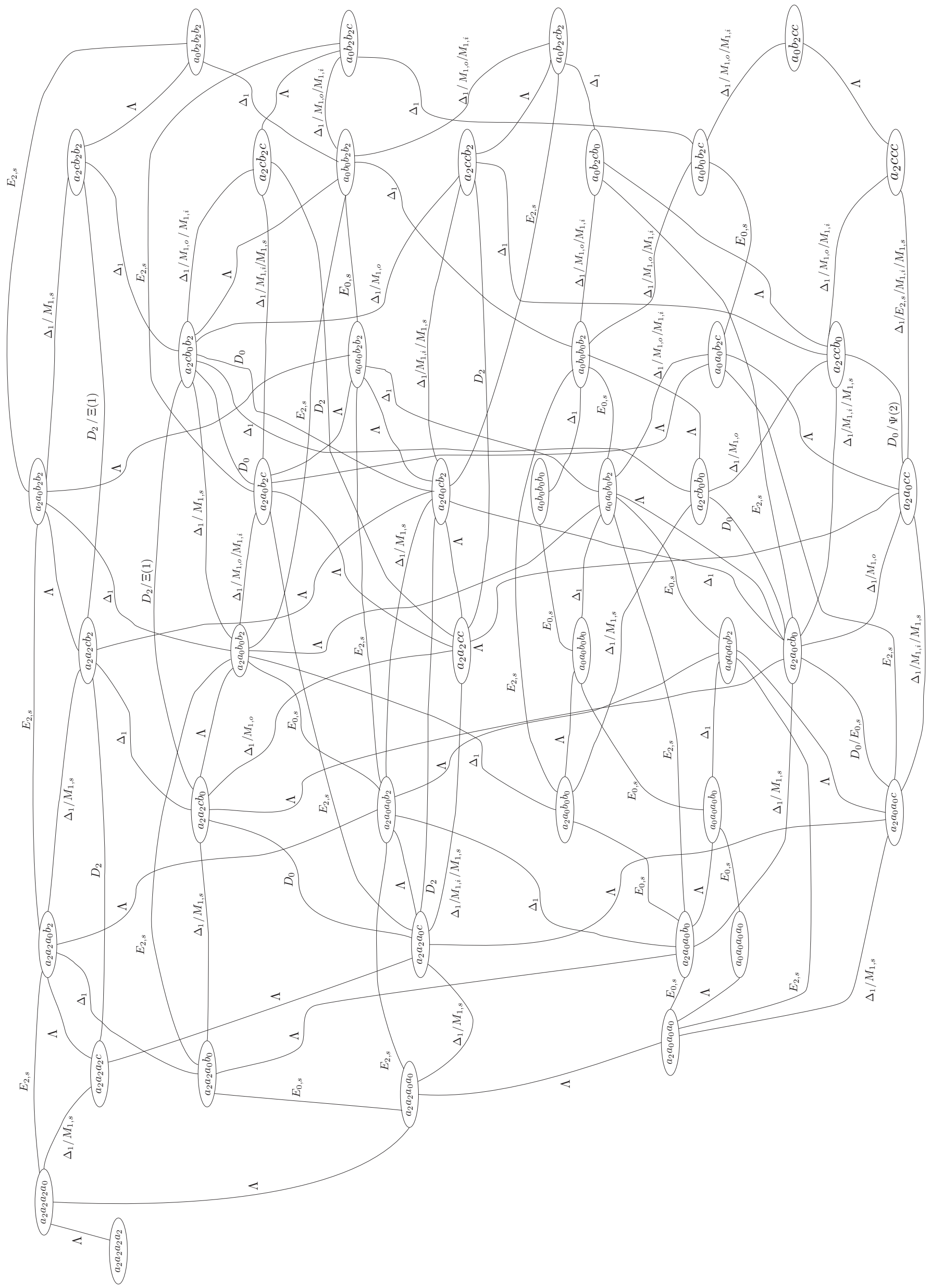


Fig. 18. First half of complete transition graph of Hamiltonian surface flows of maximal word length four.

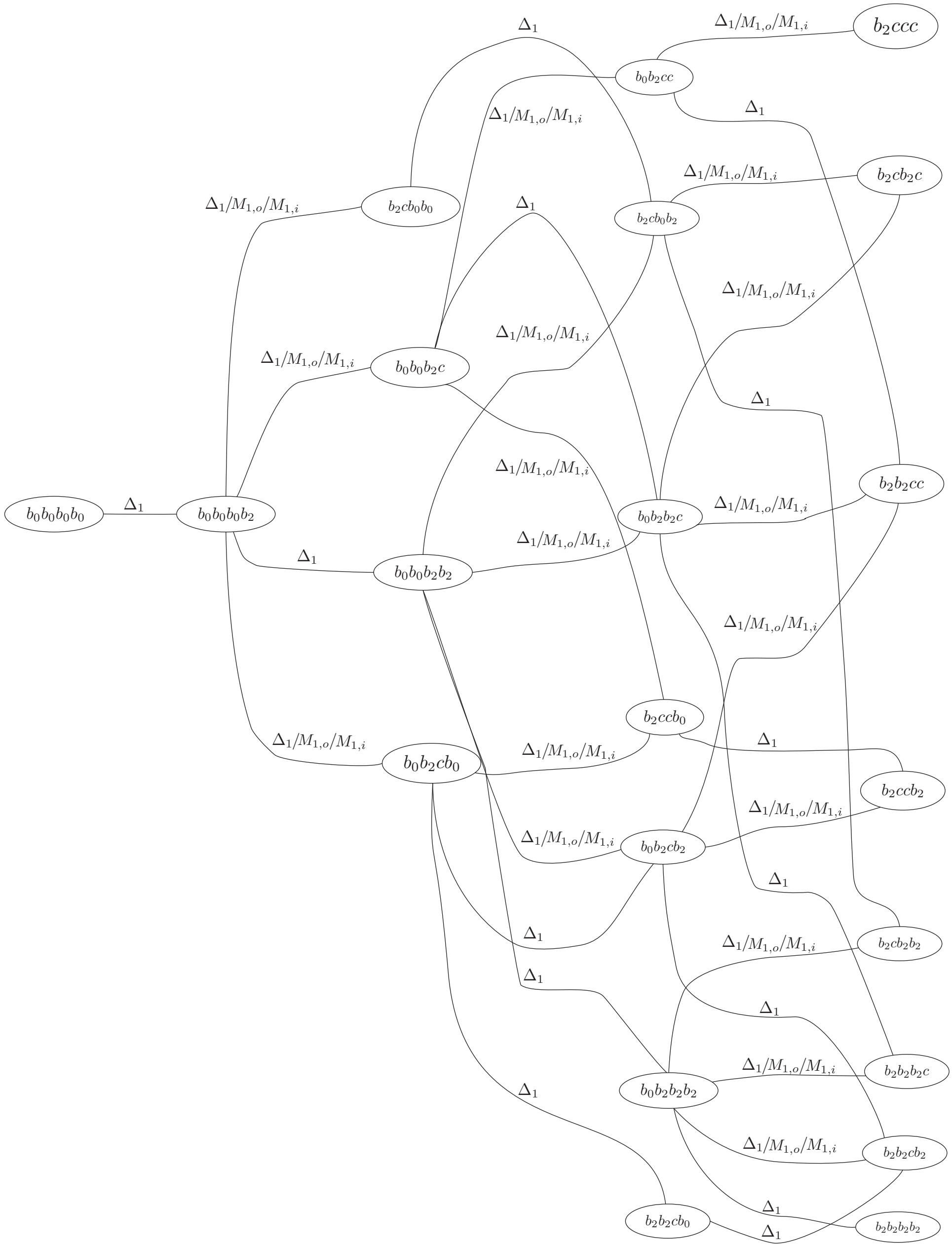


Fig. 19. Second half of complete transition graph of Hamiltonian surface flows of maximal word length four.

Now, we give an example of an analysis that uses a transition graph. In the right half of Fig. 14, the word of (1) (respectively, (5)) is $a_2a_0ccb_0$, and the word of (3) is a_2cccb_0. Using the transition graph of structurally stable Hamiltonian flows and comparing the tree structures of the flows, we can estimate the unique intermediate Hamiltonian flow (2) (respectively, (4)) of (1) and (3) (respectively, (3) and (5)) under an assumption that the transition occurs along the shortest path on the transition graph. In particular, disregarding the common letter c, we can find the edge from (1) to (3) (respectively, from (3) to (5)) in the transition graph of length four in Figs. 18 and 19.

5. Analysis Using Transition Diagrams

Let the L_1 *distance divided by two* ($L_1/2$ distance, in short) $d_1(v, w)$ of two vectors $v = (n_1, n_2, n_3, n_4, n_5)$ and $w = (m_1, m_2, m_3, m_4, m_5)$ be the sum $\sum_{i=1}^{5} |n_i - m_i|/2$, and let the *transition distance* of two vectors be the minimum number of applications of transition rules that will transform one to the other.

By reducing word representations to vector representations, Table 2 can be obtained from Table 1. Table 2 lists all the transition rules among maximal patterns of the same length that change at least one operation. To apply a transition rule to a given vector v, each element of v must be greater than or equal to the one in the corresponding precondition. When applying a transition rule, we subtract the diff vector from the elements of the given vector. Then, the postcondition must be satisfied. An overline in a diff vector represents a negative sign. For example, $\overline{1}$ means -1. Each transition rule can be applied inversely; for this, the precondition and the postcondition are swapped and the sign of each element of the diff vector is flipped.

Let the *weak transition distance* between two vectors be the minimum number of applications of vector transformations that will transform one to the other. We denote the weak transition distance and the transition distance between two vectors v_1 and v_2 by $d_2(v_1, v_2)$ and $d_3(v_1, v_2)$, respectively. For any v_1 and v_2, we have $d_1(v_1, v_2) \leq d_2(v_1, v_2) \leq d_3(v_1, v_2)$.

Table 3 shows an example of a flow transition. Each row contains the maximal word representation, its vector representation, the $L_1/2$ distance from the original flow $a_2a_2a_2$, the weak transition distance from the original flow $a_2a_2a_2$, and the transition rule to be applied next. This example transition is highlighted in Fig. 20, which shows part of the transition graph for maximal word length three. Whereas the $L_1/2$ distances to $a_2a_2a_0$ and a_2a_2c are the same because only the last letter has changed from the original word, the weak transition distance changes from one to two because at least two transitions Λ and either Δ_1 or $M_{1,s}$ are necessary for the original flow to become the flow a_2a_2c. The non-negative integer representation 3 remains unchanged during the entire process, which is depicted as the first 5 pictures in Fig. 21.

Fig. 20. A connected part of a transition graph of Hamiltonian surface flows of maximal word length three.

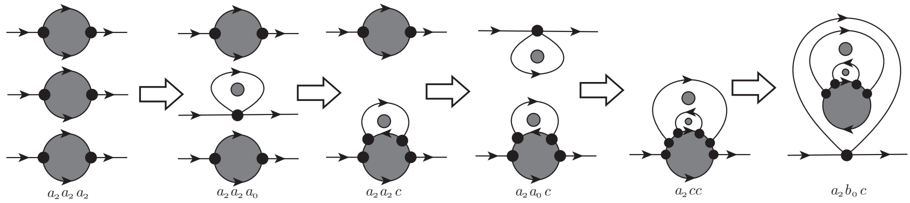


Fig. 21. Transition path of length three in Fig. 20.

Using the transition graph of Hamiltonian surface flows, the problem of finding a lower bound on the topological transition distance can be reduced to an integer programming problem. Therefore, problems of computational fluid dynamics can be reduced to optimization problems. In particular, the relaxation of an integer program for obtaining the transition distance between two vectors $n(v) := (n_{a_0}, n_{a_2}, n_{b_0}, n_{b_2}, n_c) \in (\mathbb{Z}_{\geq 0})^5$ and $n(v') := (n'_{a_0}, n'_{a_2}, n'_{b_0}, n'_{b_2}, n'_c) \in (\mathbb{Z}_{\geq 0})^5$ is as follows:

$$\begin{aligned}
&\min \sum_{i=1}^{7} x_i \\
&\text{s.t. } n_{a_0} - x_1 + x_2 + x_3 - x_4 &&= n'_{a_0} \\
&\qquad n_{a_2} - x_2 - x_3 - x_5 &&= n'_{a_2} \\
&\qquad n_{b_0} + x_1 - x_6 + x_7 &&= n'_{b_0} \\
&\qquad n_{b_2} + x_3 + x_5 + x_6 &&= n'_{b_2} \\
&\qquad n_c - x_3 + x_4 - x_7 &&= n'_c.
\end{aligned}$$

Table 2. List of transition rules that can be applied under the preconditions and the postconditions of vectors grouped by the vector differences ($i \geq 1$).

No.	Transition rule	Precondition	Diff	Postcondition
1	a_2cD_0	$\geq (1,1,0,0,1)$	$(1,0,\overline{1},0,0)$	$\geq (0,1,1,0,1)$
	$a_0E_{0,s}$	$\geq (2,0,0,0,0)$	$(1,0,\overline{1},0,0)$	$\geq (1,0,1,0,0)$
	$a_2c\Psi(2)$	$\geq (1,1,0,0,2)$	$(1,0,\overline{1},0,0)$	$\geq (0,1,1,0,2)$
2	Λ	$\geq (0,1,0,0,0)$	$(\overline{1},1,0,0,0)$	$\geq (1,0,0,0,0)$
3	Λc	$\geq (0,1,0,0,1)$	$(\overline{1},1,0,\overline{1},1)$	$\geq (1,0,0,1,0)$
4	Δ_1	$\geq (0,0,1,0,0)$	$(0,0,1,\overline{1},0)$	$\geq (0,0,0,1,0)$
5	$b_2M_{1,o}$	$\geq (0,0,1,1,0)$	$(1,0,0,0,\overline{1})$	$\geq (0,0,0,1,1)$
	$a_2cM_{1,i}$	$\geq (1,1,0,0,1)$	$(1,0,0,0,\overline{1})$	$\geq (0,1,0,0,2)$
	$a_2M_{1,s}$	$\geq (1,1,0,0,0)$	$(1,0,0,0,\overline{1})$	$\geq (0,1,0,0,1)$
	$a_2\Delta_1$	$\geq (1,1,0,0,0)$	$(1,0,0,0,\overline{1})$	$\geq (0,1,0,0,1)$
6	a_2cD_2	$\geq (0,2,0,0,1)$	$(0,1,0,\overline{1},0)$	$\geq (0,1,0,1,1)$
	$a_0E_{2,s}$	$\geq (1,1,0,0,0)$	$(0,1,0,\overline{1},0)$	$\geq (1,0,0,1,0)$
	$a_2c\Xi(2i-1)$	$\geq (0,2,0,0,2i-1)$	$(0,1,0,\overline{1},0)$	$\geq (0,1,0,1,2i-1)$
7	$b_2\Delta_1$	$\geq (0,0,1,1,0)$	$(0,0,1,0,\overline{1})$	$\geq (0,0,0,1,1)$
	$c\Delta_1$	$\geq (0,0,1,0,1)$	$(0,0,1,0,\overline{1})$	$\geq (0,0,0,0,2)$
	$a_2cM_{1,o}$	$\geq (0,1,1,0,1)$	$(0,0,1,0,\overline{1})$	$\geq (0,1,0,0,2)$
	$b_2M_{1,i}$	$\geq (0,0,1,1,0)$	$(0,0,1,0,\overline{1})$	$\geq (0,0,0,1,1)$
	$ccM_{1,i}$	$\geq (0,0,1,0,2)$	$(0,0,1,0,\overline{1})$	$\geq (0,0,0,0,3)$

Table 3. An example transition and the change of its measures. Here, Vector rep. (respectively, $L_1/2$ dist., Weak trans. dist., Trans. rule) stands for Vector representation (respectively, $L_1/2$ distance, Weak transition distance, Transition rule).

Maximal word	Vector rep.	$L_1/2$ dist.	Weak trans. dist.	Trans. rule
$a_2a_2a_2$	$(0, 3, 0, 0, 0)$	0	0	Λ
$a_2a_2a_0$	$(1, 2, 0, 0, 0)$	1	1	Δ_1
a_2a_2c	$(0, 2, 0, 0, 1)$	1	2	$E_{2,s}$
a_2a_0c	$(1, 1, 0, 0, 1)$	2	3	$M_{1,i}t$
a_2cc	$(0, 1, 0, 0, 2)$	2	4	

Here, x_i is the number of applications of transition rule no. i, and the problem is to minimize the total number of applications of transition rules.

The transition rule $a_0a_0 \to a_0b_0$ is referred to by the name $E_{0,s}$.

6. Conclusion

We have proposed a method for constructing complete transition graphs of generic Hamiltonian flows with a few heteroclinic orbits. The transition among the vertices in the graphs is a necessary condition for having structurally stable Hamiltonian flow changes. The check is computationally lightweight.

To determine the usefulness of the proposed graphs, we plan to conduct experiments using real orbit data and to estimate and improve the performance and memory usage of the analysis by computer simulations. Our approach does not target orbits with higher numbers of heteroclinic orbits. More precisely, the transitions in Fig. 16(a)–16(c) for n greater than one are not considered. The usefulness of the study toward this end depends on how often the applications require analysis of the transitions that are less likely to occur in practice. The existence of a transition edge between vertices in our graphs does not necessarily mean that the transition is possible for all pairs of orbits in both vertices; it only means that there is at least one pair of orbits in both vertices that can undergo the transition. The classification of transitions of generic Hamiltonian flows whose word length is greater than four has been studied [10], and the analysis of these transitions is reserved for future work.

Acknowledgments

This work is partially supported by Nanzan University Pache Research Subsidy I-A-2 for the 2020 academic year and is partially supported by JST PRESTO Grant Number JPMJPR16E and JSPS Kakenhi Grant Numbers 20K03583, 18H01136.

References

[1] S. Kh. Aranson, G. R. Belitsky and E. V. Zhuzhoma, *Introduction to the Qualitative Theory of Dynamical Systems on Surfaces*, Translations of Mathematical Monographs, Vol. 153 (American Mathematical Society, 1996).

[2] H. Aref and M. Brøns, On stagnation points and streamline topology in vortex flows, *J. Fluid Mech.* **370** (1998) 1–27.

[3] M. Cobo, C. Gutierrez and J. Llibre, Flows without wandering points on compact connected surfaces, *Trans. Amer. Math. Soc.* **362**(9) (2010) 4569–4580.

[4] A. T. Fomenko and H. Zieschang, A topological invariant and a criterion for the equivalence of integrable Hamiltonian systems with two degrees of freedom, *Izv. Akad. Nauk SSSR Ser. Mat.* **54**(3) (1990) 546–575 (in Russian); *Math. USSR-Izv.* **36**(3) (1991) 567–596.

[5] R. Kidambi and P. K. Newton, Streamline topologies for integrable vortex motion on a sphere, *Phys. D Nonlinear Phenom.* **140**(1–2) (2000) 95–125.

[6] T. Ma and S. Wang, *Geometric Theory of Incompressible Flows with Applications to Fluid Dynamics*, Mathematical Surveys and Monographs, Vol. 119 (American Mathematical Society, 2005).

[7] M. L. Minsky, *Computation: Finite and Infinite Machines* (Prentice-Hall, Inc., USA, 1967).

[8] H. K. Moffatt, The topology of scalar fields in 2D and 3D turbulence, in *IUTAM Symp. Geometry and Statistics of Turbulence* (Springer, Dordrecht, 2001), pp. 13–22.

[9] T. Sakajo, Y. Sawamura and T. Yokoyama, Unique encoding for streamline topologies of incompressible and inviscid flows in multiply connected domains, *Fluid Dynam. Res.* **46**(3) (2014) 031411.

[10] T. Sakajo and T. Yokoyama, ‘Transitions between streamline topologies of structurally stable Hamiltonian flows in multiply connected domains, *Phys. D Nonlinear Phenom.* **307** (2015) 22–41.

[11] T. Yokoyama and T. Sakajo, Word representation of streamline topologies for structurally stable vortex flows in multiply connected domains, *Proc. R. Soc. A Math. Phys. Eng. Sci.* **469**(2150) (2013) 20120558.