

PAPER • OPEN ACCESS

Infinitesimal deformations of Poisson bi-vectors using the Kontsevich graph calculus

To cite this article: Ricardo Buring *et al* 2018 *J. Phys.: Conf. Ser.* **965** 012010

View the [article online](#) for updates and enhancements.

Related content

- [Differential Topology and Geometry with Applications to Physics: Integration on manifolds](#)
E Nahmad-Achar
- [The Foundations of Electric Circuit Theory: Mathematical introduction](#)
N R S Harsha, A Prakash and D P Kothari
- [Do the Kontsevich tetrahedral flows preserve or destroy the space of Poisson bi-vectors?](#)
Anass Bouisaghouane and Arthemy V Kiselev



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Infinitesimal deformations of Poisson bi-vectors using the Kontsevich graph calculus

Ricardo Buring¹, Arthemy V Kiselev² and Nina Rutten²

¹ Institut für Mathematik, Johannes Gutenberg–Universität, Staudingerweg 9, D-55128 Mainz, Germany

² Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, P.O. Box 407, 9700 AK Groningen, The Netherlands

E-mail: rburing@uni-mainz.de, A.V.Kiselev@rug.nl

Abstract. Let \mathcal{P} be a Poisson structure on a finite-dimensional affine real manifold. Can \mathcal{P} be deformed in such a way that it stays Poisson? The language of Kontsevich graphs provides a universal approach – with respect to all affine Poisson manifolds – to finding a class of solutions to this deformation problem. For that reasoning, several types of graphs are needed. In this paper we outline the algorithms to generate those graphs. The graphs that encode deformations are classified by the number of internal vertices k ; for $k \leq 4$ we present all solutions of the deformation problem. For $k \geq 5$, first reproducing the pentagon-wheel picture suggested at $k = 6$ by Kontsevich and Willwacher, we construct the heptagon-wheel cocycle that yields a new unique solution without 2-loops and tadpoles at $k = 8$.

Introduction. This paper contains a set of algorithms to generate the Kontsevich graphs that encode polydifferential operators – in particular multi-vectors – on Poisson manifolds. We report a result of implementing such algorithms in the problem of finding symmetries of Poisson structures. Namely, continuing the line of reasoning from [1, 2], we find all the solutions of this deformation problem that are expressed by the Kontsevich graphs with at most four internal vertices. Next, we present one six-vertex solution (based on the previous work by Kontsevich [10] and Willwacher [13]). Finally, we find a heptagon-wheel eight-vertex graph which, after the orientation of its edges, gives a new universal Kontsevich flow. We refer to [8, 9] for motivations, to [2, 4] for an exposition of basic theory, and to [6] and [5] for more details about the pentagon-wheel (5+1)-vertex and heptagon-wheel (7+1)-vertex solutions respectively. Let us remark that all the algorithms outlined here can be used without modification in the course of constructing all k -vertex Kontsevich graph solutions with higher $k \geq 5$ in the deformation problem under study.

Basic concept. We work with real vector spaces generated by finite graphs of the following two types: (1) k -vertex non-oriented graphs, without multiple edges nor tadpoles, endowed with a wedge ordering of edges, e.g., $E = e_1 \wedge \cdots \wedge e_{2k-2}$; (2) oriented graphs on k internal vertices and n sinks such that every internal vertex is a tail of two edges with a given ordering Left \prec Right. Every connected component of a non-oriented graph γ is fully encoded by an ordering E



on the set of adjacency relations for its vertices.¹ Every such oriented graph is given by the list of ordered pairs of directed edges. An edge swap $e_i \wedge e_j = -e_j \wedge e_i$ and the reversal Left \leftrightarrow Right of those edges' order in the tail vertex implies the change of sign in front of the graph at hand.²

Example 1. The sum γ_5 of two 6-vertex 10-edge graphs,

$$\begin{aligned} \gamma_5 = & 12^{(I)} \wedge 23^{(II)} \wedge 34^{(III)} \wedge 45^{(IV)} \wedge 51^{(V)} \wedge 16^{(VI)} \wedge 26^{(VII)} \wedge 36^{(VIII)} \wedge 46^{(IX)} \wedge 56^{(X)} \\ & + \frac{5}{2} \cdot 12^{(I)} \wedge 23^{(II)} \wedge 34^{(III)} \wedge 41^{(IV)} \wedge 45^{(V)} \wedge 15^{(VI)} \wedge 56^{(VII)} \wedge 36^{(VIII)} \wedge 26^{(IX)} \wedge 13^{(X)}, \end{aligned}$$

is drawn in Theorem 7 on p. 10 below.

Example 2. The sum $\mathcal{Q}_{1;\frac{6}{2}}$ of three oriented 8-edge graphs on $k = 4$ internal vertices and $n = 2$ sinks (enumerated using 0 and 1, see the notation on p. 4),

$$\mathcal{Q}_{1;\frac{6}{2}} = 2\ 4\ 1\ 0\ 1\ 2\ 4\ 2\ 5\ 2\ 3 - 3(2\ 4\ 1\ 0\ 3\ 1\ 4\ 2\ 5\ 2\ 3 + 2\ 4\ 1\ 0\ 3\ 4\ 5\ 1\ 2\ 2\ 4)$$

is obtained from the non-oriented tetrahedron graph $\gamma_3 = 12^{(I)} \wedge 13^{(II)} \wedge 14^{(III)} \wedge 23^{(IV)} \wedge 24^{(V)} \wedge 34^{(VI)}$ on four vertices and six edges by taking all the admissible edge orientations (see Theorem 4 and Remark 1).

I.1. Let γ_1 and γ_2 be connected non-oriented graphs. The definition of insertion $\gamma_1 \circ_i \gamma_2$ of the entire graph γ_1 into vertices of γ_2 and the construction of Lie bracket $[\cdot, \cdot]$ of graphs and differential d in the non-oriented graph complex, referring to a sign convention, are as follows (cf. [8] and [7, 11, 12]); these definitions apply to sums of graphs by linearity.

Definition 1. The insertion $\gamma_1 \circ_i \gamma_2$ of a k_1 -vertex graph γ_1 with ordered set of edges $E(\gamma_1)$ into a graph γ_2 with $\#E(\gamma_2)$ edges on k_2 vertices is the sum of k_2 graphs on $k_1 + k_2 - 1$ vertices and $\#E(\gamma_1) + \#E(\gamma_2)$ edges. Topologically, the sum $\gamma_1 \circ_i \gamma_2 = \sum (\gamma_1 \rightarrow v \text{ in } \gamma_2)$ consists of all the graphs in which a vertex v from γ is replaced by the entire graph γ_1 and the edges touching v in γ_2 are re-attached to the vertices of γ_1 in all possible ways.³ By convention, in every new term the edge ordering is $E(\gamma_1) \wedge E(\gamma_2)$.

To simplify sums of graphs, first eliminate the zero graphs. Now suppose that in a sum, two non-oriented graphs, say α and β , are isomorphic (topologically, i.e. regardless of the respective vertex labellings and edge orderings $E(\alpha)$ and $E(\beta)$). By using that isomorphism, which establishes a 1-1 correspondence between the edges, extract the sign from the equation $E(\alpha) = \pm E(\beta)$. If “+”, then $\alpha = \beta$; else $\alpha = -\beta$. Collecting similar terms is now elementary.

Lemma 1. The bi-linear graded skew-symmetric operation,

$$[\gamma_1, \gamma_2] = \gamma_1 \circ_i \gamma_2 - (-)^{\#E(\gamma_1) \cdot \#E(\gamma_2)} \gamma_2 \circ_i \gamma_1,$$

¹ The edges are antipermutable so that a graph which equals minus itself –under a symmetry that induces a parity-odd permutation of edges– is proclaimed to be equal to zero. In particular (view $\bullet\text{---}\bullet\text{---}\bullet$), every graph possessing a symmetry which swaps an odd number of edge pairs is a zero graph. For example, the 4-wheel $12 \wedge 13 \wedge 14 \wedge 15 \wedge 23 \wedge 25 \wedge 34 \wedge 45 = I \wedge \dots \wedge VIII$ or the 2ℓ -wheel at any $\ell > 1$ is such; here, the reflection symmetry is $I \rightleftharpoons III$, $V \rightleftharpoons VII$, and $VI \rightleftharpoons VIII$.

² An oriented graph equals minus itself, hence it is a zero graph if there is a permutation of labels for its internal vertices such that the adjacency tables for the two vertex labellings coincide but the two realisations of the same graph differ by the ordering of outgoing edges at an odd number of internal vertices (see Example 3 below).

³ Let the enumeration of vertices in every such term in the sum start running over the enumerated vertices in γ_2 until v is reached. Now the enumeration counts the vertices in the graph γ_1 and then it resumes with the remaining vertices (if any) that go after v in γ_2 .

is a Lie bracket on the vector space \mathfrak{G} of non-oriented graphs.⁴

Lemma 2. The operator $d(\text{graph}) = [\bullet\!\!\!\bullet, \text{graph}]$ is a differential: $d^2 = 0$.

In effect, the mapping d blows up every vertex v in its argument in such a way that whenever the number of adjacent vertices $N(v) \geq 2$ is sufficient, each end of the inserted edge $\bullet\!\!\!\bullet$ is connected with the rest of the graph by at least one edge.

Summarising, the real vector space \mathfrak{G} of non-oriented graphs is a differential graded Lie algebra (dgLa) with Lie bracket $[\cdot, \cdot]$ and differential $d = [\bullet\!\!\!\bullet, \cdot]$. The graphs γ_5 and γ_3 from Examples 1 and 2 are d -cocycles. Neither is exact, hence marking a nontrivial cohomology class in the non-oriented graph complex.

Theorem 3 ([7, Th. 5.5]). *At every $\ell \in \mathbb{N}$ in the connected graph complex there is a d -cocycle on $2\ell + 1$ vertices and $4\ell + 2$ edges. Such cocycle contains the $(2\ell + 1)$ -wheel in which, by definition, the axis vertex is connected with every other vertex by a spoke so that each of those 2ℓ vertices is adjacent to the axis and two neighbours; the cocycle marked by the $(2\ell + 1)$ -wheel graph can contain other $(2\ell + 1, 4\ell + 2)$ -graphs (see Example 1 and [5]).*

I.2. The oriented graphs under study are built over n sinks from k wedges $\xleftarrow{i_\alpha} \bullet \xrightarrow{j_\alpha}$ (here $\xleftarrow{i_\alpha} \prec \xrightarrow{j_\alpha}$) so that every edge is decorated with its own summation index which runs from 1 to the dimension of a given affine Poisson manifold $(\mathcal{N}, \mathcal{P})$. Each edge \xrightarrow{i} encodes the derivation $\partial/\partial x^i$ of the arrowhead object with respect to a local coordinate x^i on \mathcal{N} . By placing an α th copy $P^{i_\alpha j_\alpha}(\mathbf{x})$ of the Poisson bi-vector \mathcal{P} in the wedge top ($1 \leq \alpha \leq k$), by taking the product of contents of the $n + k$ vertices (and evaluating all objects at a point $\mathbf{x} \in \mathcal{N}$), and summing over all indices, we realise a polydifferential operator in n arguments; the operator coefficients are differential-polynomial in \mathcal{P} . Totally skew-symmetric operators of differential order one in each argument are well-defined n -vectors on the affine manifolds \mathcal{N} .

The space of multi-vectors G encoded by oriented graphs is equipped with a graded Lie algebra structure, namely the Schouten bracket $[[\cdot, \cdot]]$. Its realisation in terms of oriented graphs is shown in [2, Remark 4]. Recall that by definition the bi-vectors \mathcal{P} at hand are Poisson by satisfying the Jacobi identity $[[\mathcal{P}, \mathcal{P}]] = 0$. The Poisson differential $\partial_{\mathcal{P}} = [[\mathcal{P}, \cdot]]$ now endows the space of multi-vectors on \mathcal{N} with the differential graded Lie algebra (dgLa) structure. The cohomology groups produced by the two dgLa structures introduced so far are correlated by the edge orientation mapping $\text{O}\vec{\cdot}$.

Theorem 4 ([8] and [12, App. K]). *Let $\gamma \in \ker d$ be a cocycle on k vertices and $2k - 2$ edges in the non-oriented graph complex. Denote by $\{\Gamma\} \subset G$ the subspace spanned by all those bi-vector graphs Γ which are obtained from (each connected component in) γ by adding to it two edges to the new sink vertices and then by taking the sum of graphs with all the admissible orientations of the old $2k - 2$ edges (so that a set of Kontsevich graphs built of k wedges is produced). Then in that subspace $\{\Gamma\}$ there is a sum of graphs that encodes a nonzero Poisson cocycle $Q(\mathcal{P}) \in \ker \partial_{\mathcal{P}}$.*

Consequently, to find some cocycle $Q(\mathcal{P})$ in the Poisson complex on any affine Poisson manifold it suffices to find a cocycle in the non-oriented graph complex and then consider the sum of graphs which are produced by the orientation mapping $\text{O}\vec{\cdot}$. On the other hand, to list all the $\partial_{\mathcal{P}}$ -cocycles $Q(\mathcal{P})$ encoded by the bi-vector graphs made of k wedges $\leftarrow\bullet\rightarrow$, one must generate all the relevant oriented graphs and solve the equation $\partial_{\mathcal{P}}(Q) \doteq 0$ via $[[\mathcal{P}, \mathcal{P}]] = 0$, that is, solve

⁴ The postulated precedence or antecedence of the wedge product of edges from γ_1 with respect to the edges from γ_2 in every graph within $\gamma_1 \circ_i \gamma_2$ produce the operations \circ_i which coincide with \circ , respectively, differ from Definition 1 by the sign factor $(-)^{\#E(\gamma_1) \cdot \#E(\gamma_2)}$. The same applies to the Lie bracket of graphs $[\gamma_1, \gamma_2]$ if the operation $\gamma_1 \circ_i \gamma_2$ is the insertion of γ_2 into γ_1 (as in [11]). Anyway, the notion of d -cocycles which we presently recall is well defined and insensitive to such sign ambiguity.

graphically the factorisation problem $[[\mathcal{P}, Q(\mathcal{P})]] = \diamond(\mathcal{P}, [[\mathcal{P}, \mathcal{P}]])$ in which the cocycle condition in the left-hand side holds by virtue of the Jacobi identity in the right. Such construction of some and classification (at a fixed $k > 0$) of all universal infinitesimal symmetries of Poisson brackets are the problems which we explore in this paper.

Remark 1. To the best of our knowledge [10], in a bi-vector graph $Q(\mathcal{P}) = \text{Or}\bar{\gamma}$, at every internal vertex which is the tail of two oriented edges towards other internal vertices, the edge ordering Left \prec Right is inherited from a chosen wedge product $E(\gamma)$ of edges in the non-oriented graph γ . How are the new edges towards the sinks ordered, either between themselves at a vertex or with respect to two other oriented edges, coming from γ and issued from different vertices in $Q(\mathcal{P})$? Our findings in [6] will help us to verify the order preservation claim and assess answers to this question.

1. The Kontsevich graph calculus

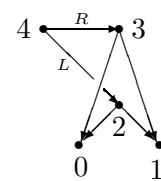
Definition 2. Let us consider a class of oriented graphs on $n+k$ vertices labelled $0, \dots, n+k-1$ such that the consecutively ordered vertices $0, \dots, n-1$ are sinks, and each of the internal vertices $n, \dots, n+k-1$ is a source for two edges. For every internal vertex, the two outgoing edges are ordered using $L \prec R$: the preceding edge is labelled L (Left) and the other is R (Right). An oriented graph on n sinks and k internal vertices is a *Kontsevich graph* of type (n, k) .

For the purpose of defining a graph normal form, we now consider a Kontsevich graph Γ together with a *sign* $s \in \{0, \pm 1\}$, denoted by concatenation of the symbols: $s\Gamma$.

Notation (Encoding of the Kontsevich graphs). The format to store a signed graph $s\Gamma$ for a Kontsevich graph Γ is the integer number $n > 0$, the integer $k \geq 0$, the sign s , followed by the (possibly empty, when $k = 0$) list of k ordered pairs of targets for edges issued from the internal vertices $n, \dots, n+k-1$, respectively. The full format is then $(n, k, s; \text{list of ordered pairs})$.

Definition 3 (Normal form of a Kontsevich graph). The list of targets in the encoding of a graph Γ can be considered as a $2k$ -digit integer written in base- $(n+k)$ notation. By running over the entire group $S_k \times (\mathbb{Z}_2)^k$, and by this over all the different re-labellings of Γ , we obtain many different integers written in base- $(n+k)$. The *absolute value* $|\Gamma|$ of Γ is the re-labelling of Γ such that its list of targets is *minimal* as a nonnegative base- $(n+k)$ integer. For a signed graph $s\Gamma$, the *normal form* is the signed graph $t|\Gamma|$ which represents the same polydifferential operator as $s\Gamma$. Here we let $t = 0$ if the graph is zero (see Example 3 below).

Example 3 (Zero Kontsevich graph). Consider the graph with the encoding $2\ 3\ 1\ 0\ 1\ 0\ 1\ 2\ 3$. The swap of vertices $2 \rightleftharpoons 3$ is a symmetry of this graph, yet it also swaps the ordered edges $(4 \rightarrow 2) \prec (4 \rightarrow 3)$, producing a minus sign. Equal to minus itself, this Kontsevich graph is zero.



Notation. Every Kontsevich graph Γ on n sinks (or every sum Γ of such graphs) yields the sum $\text{Alt } \Gamma$ of Kontsevich graphs which is totally skew-symmetric with respect to the n sinks content s_1, \dots, s_n . Indeed, let

$$(\text{Alt } \Gamma)(s_1, \dots, s_n) = \sum_{\sigma \in \mathcal{S}_n} (-)^\sigma \Gamma(s_{\sigma(1)}, \dots, s_{\sigma(n)}). \tag{1}$$

Due to skew-symmetrisation, the sum of graphs $\text{Alt } \Gamma$ can contain zero graphs or repetitions.

Example 4 (The Jacobiator). The left-hand side of the Jacobi identity is a skew sum of Kontsevich graphs (e.g. it is obtained by skew-symmetrizing the first term)

$$\begin{array}{c} \boxed{\bullet \bullet} \\ \downarrow \downarrow \downarrow \\ 1 \quad 2 \quad 3 \end{array}
 :=
 \begin{array}{c} \bullet \\ \swarrow \downarrow \searrow \\ i \quad j \quad k \\ \downarrow \downarrow \downarrow \\ 1 \quad 2 \quad 3 \end{array}
 -
 \begin{array}{c} \bullet \\ \swarrow \downarrow \searrow \\ i \quad j \quad k \\ \downarrow \downarrow \downarrow \\ 1 \quad 2 \quad 3 \end{array}
 -
 \begin{array}{c} \bullet \\ \swarrow \downarrow \searrow \\ i \quad j \quad k \\ \downarrow \downarrow \downarrow \\ 1 \quad 2 \quad 3 \end{array}
 . \tag{2}$$

The default ordering of edges is the one which we see.

Definition 4 (Leibniz graph). A *Leibniz graph* is a graph whose vertices are either sinks, or the sources for two arrows, or the Jacobiator (which is a source for three arrows). There must be at least one Jacobiator vertex. The three arrows originating from a Jacobiator vertex must land on three distinct vertices. Each edge falling on a Jacobiator works by the Leibniz rule on the two internal vertices in it.

Example 5. The Jacobiator itself is a Leibniz graph (on one tri-valent internal vertex).

Definition 5 (Normal form of a Leibniz graph with one Jacobiator). Let Γ be a Leibniz graph with one Jacobiator vertex Jac . From (2) we see that expansion of Jac into a sum of three Kontsevich graphs means adding one new edge $w \rightarrow v$ (namely joining the internal vertices w and v within the Jacobiator). Now, from Γ construct three Kontsevich graphs by expanding Jac using (2) and letting the edges which fall on Jac in Γ be directed only to v in every new graph. Next, for each Kontsevich graph find the relabelling τ which brings it to its normal form and re-express the edge $w \rightarrow v$ using τ . Finally, out of the three normal forms of three graphs pick the minimal one. By definition, the *normal form* of the Leibniz graph Γ is the pair: normal form of Kontsevich graph, that edge $\tau(w) \rightarrow \tau(v)$.

We say that a sum of Leibniz graphs is a *skew Leibniz graph* $\text{Alt } \Gamma$ if it is produced from a given Leibniz graph Γ by alternation using formula (1).

Definition 6 (Normal form of a skew Leibniz graph with one Jacobiator). Likewise, the normal form of a skew Leibniz graph $\text{Alt } \Gamma$ is the minimum of the normal forms of Leibniz graphs (specifically, of the graph but not edge encodings) which are obtained from Γ by running over the group of permutations of the sinks content.

Lemma 5 ([3]). In order to show that a sum S of weighted skew-symmetric Kontsevich graphs vanishes for all Poisson structures \mathcal{P} , it suffices to express S as a sum of skew Leibniz graphs: $S = \diamond(\mathcal{P}, \text{Jac}(\mathcal{P}))$.

1.1. Formulation of the problem

Let $\mathcal{P} \mapsto \mathcal{P} + \varepsilon \mathcal{Q}(\mathcal{P}) + \bar{o}(\varepsilon)$ be a deformation of bi-vectors that preserves their property to be Poisson at least infinitesimally on all affine manifolds: $\llbracket \mathcal{P} + \varepsilon \mathcal{Q} + \bar{o}(\varepsilon), \mathcal{P} + \varepsilon \mathcal{Q} + \bar{o}(\varepsilon) \rrbracket = \bar{o}(\varepsilon)$. Expanding and equating the first order terms, we obtain the equation $\llbracket \mathcal{P}, \mathcal{Q}(\mathcal{P}) \rrbracket \doteq 0$ via $\llbracket \mathcal{P}, \mathcal{P} \rrbracket = 0$. The language of Kontsevich graphs allows one to convert this infinite analytic problem within a given set-up $(\mathcal{N}^n, \mathcal{P})$ in dimension n into a set of finite combinatorial problems whose solutions are universal for all Poisson geometries in all dimensions $n < \infty$.

Our first task in this paper is to find the space of flows $\dot{\mathcal{P}} = \mathcal{Q}(\mathcal{P})$ which are encoded by the Kontsevich graphs on a fixed number of internal vertices k , for $1 \leq k \leq 4$. Specifically, we solve the graph equation

$$\llbracket \mathcal{P}, \mathcal{Q}(\mathcal{P}) \rrbracket = \diamond(\mathcal{P}, \text{Jac}(\mathcal{P})) \quad (3)$$

for the Kontsevich bi-vector graphs $\mathcal{Q}(\mathcal{P})$ and Leibniz graphs \diamond . We then factor out the Poisson-trivial and improper solutions, that is, we quotient out all bi-vector graphs that can be written in the form $\mathcal{Q}(\mathcal{P}) = \llbracket \mathcal{P}, X \rrbracket + \nabla(\mathcal{P}, \text{Jac}(\mathcal{P}))$, where X is a Kontsevich one-vector graph and ∇ is a Leibniz bi-vector graph. (The bi-vectors $\llbracket \mathcal{P}, X \rrbracket$ make $\llbracket \mathcal{P}, \mathcal{Q}(\mathcal{P}) \rrbracket$ vanish since $\llbracket \mathcal{P}, \cdot \rrbracket$ is a differential. The improper graphs $\nabla(\mathcal{P}, \text{Jac}(\mathcal{P}))$ vanish identically at all Poisson bi-vectors \mathcal{P} on every affine manifold.

Before solving factorisation problem (3) with respect to the operator \diamond , we must generate –e.g., iteratively as described below– an ansatz for expansion of the right-hand side using skew Leibniz graphs with undetermined coefficients.

1.2. How to generate Leibniz graphs iteratively

The first step is to construct a layer of skew Leibniz graphs, that is, all skew Leibniz graphs which produce at least one graph in the input (in the course of expansion of skew Leibniz graphs using formula (1) and then in the course of expansion of every Leibniz graph at hand to a sum of Kontsevich graphs). For a given Kontsevich graph in the input S , one such Leibniz graph can be constructed by contracting an edge between two internal vertices so that the new vertex with three outgoing edges becomes the Jacobiator vertex. Note that these Leibniz graphs, which are designed to reproduce S , may also produce extra Kontsevich graphs that are not given in the input. Clearly, if the set of Kontsevich graphs in S coincides with the set of such graphs obtained by expansion of all the Leibniz graphs in the ansatz at hand, then we are done: the extra graphs, not present in S , are known to all cancel. Yet it could very well be that it is not possible to express S using only the Leibniz graphs from the set accumulated so far. Then we proceed by constructing the next layer of skew Leibniz graphs that reproduce at least one of the extra Kontsevich graphs (which were not present in S but which are produced by the graphs in the previously constructed layer(s) of Leibniz graphs). In this way we proceed iteratively until no new Leibniz graphs are found; of course, the overall number of skew Leibniz graphs on a fixed number of internal vertices and sinks is bounded from above so that the algorithm always terminates. Note that the Leibniz graphs obtained in this way are the only ones that can in principle be involved in the vanishing mechanism for S .

Notation. Let v be a graph vertex. Denote by $N(v)$ the set of neighbours of v , by $H(v)$ the (possibly empty) set of arrowheads of oriented edges issued from the vertex v , and by $T(v)$ the (possibly empty) set of tails for oriented edges pointing at v . For example, $\#N(\bullet) = 2$, $\#H(\bullet) = 2$, and $T(\bullet) = \emptyset$ for the top \bullet of the wedge graph $\leftarrow \bullet \rightarrow$.

Algorithm Consider a skew-symmetric sum S_0 of oriented Kontsevich graphs with real coefficients. Let $S_{\text{total}} := S_0$ and create an empty table L . We now describe the i th iteration of the algorithm ($i \geq 1$).

Loop \circ Run over all Kontsevich graphs Γ in S_{i-1} : for each internal vertex v in a graph Γ , run over all vertices $w \in T(v)$ in the set of tails of oriented edges pointing at v such that $v \notin T(w)$ and $H(v) \cap H(w) = \emptyset$ for the sets of targets of oriented edges issued from v and w . Replace the edge $w \rightarrow v$ connecting w to v by Jacobiator (2), that is, by a single vertex Jac with three outgoing edges and such that $T(\text{Jac}) = (T(v) \setminus w) \cup T(w)$ and $H(\text{Jac}) = H(v) \cup (H(w) \setminus v) =: \{a, b, c\}$.

Because we shall always expand the skew Leibniz graphs in what follows, we do not actually contract the edge $w \rightarrow v$ (to obtain a Leibniz graph explicitly) in this algorithm but instead we continue working with the original Kontsevich graphs containing the distinct vertices v and w .

For every edge that points at w , redirect it to v . Sum over the three cyclic permutations that provide three possible ways to attach the three outgoing edges for v and w (excluding $w \rightarrow v$) – now seen as the outgoing edges of the Jacobiator – to the target vertices a , b , and c depending on w and v . Skew-symmetrise⁵ each of these three graphs with respect to the sinks by applying formula (1).

For every marked edge $w \rightarrow v$ indicating the internal edge in the Jacobiator vertex in a graph, replace each sum of the Kontsevich graphs which is skew with respect to the sink content by using the normal form of the respective skew Leibniz graph, see Definition 6. If this skew Leibniz graph is not contained in L , apply the Leibniz rule(s) for all the derivations acting on the Jacobiator vertex Jac. Otherwise speaking, sum over all possible ways to attach the incoming edges of the target v in the marked edge $w \rightarrow v$ to its source w and target. To each Kontsevich

⁵ This algorithm can be modified so that it works for an input which is not skew, namely, by replacing skew Leibniz graphs by ordinary Leibniz graphs (that is, by skipping the skew-symmetrisation). For example, this strategy has been used in [3, 4] to show that the Kontsevich star product $\star \bmod \bar{o}(\hbar^4)$ is associative: although the associator $(f \star g) \star h - f \star (g \star h) = \diamond(\mathcal{P}, \text{Jac}(\mathcal{P}))$ is not skew, it does vanish for every Poisson structure \mathcal{P} .

graph resulting from a skew Leibniz graph at hand assign the same undetermined coefficient, and add all these weighted Kontsevich graphs to the sum S_i . Further, add a row to the table L , that new row containing the normal form of this skew Leibniz graph (with its coefficient that has been made common to the Kontsevich graphs).

By now, the new sum of Kontsevich graphs S_i is fully composed. Having thus finished the current iteration over all graphs Γ in the set S_{i-1} , redefine the algebraic sum of weighted graphs S_{total} by subtracting from it the newly formed sum S_i . Collect similar terms in S_{total} and reduce this sum of Kontsevich graphs modulo their skew-symmetry under swaps $L \rightleftharpoons R$ of the edge ordering in every internal vertex, so that all zero graphs (see Example 3) also get eliminated. ○ end loop

Increment i by 1 and repeat the iteration until the set of weighted (and skew) Leibniz graphs L stabilizes. Finally, solve – with respect to the coefficients of skew Leibniz graphs – the linear algebraic system obtained from the graph equation $S_{\text{total}} = 0$ for the sum of Kontsevich graphs which has been produced from its initial value S_0 by running the iterations of the above algorithm.

Example 6. For the skew sum of Kontsevich graphs in the right-hand side of (2), the algorithm would produce just one skew Leibniz graph: namely, the Jacobiator itself.

Example 7 (The 3-wheel). For the Kontsevich tetrahedral flow $\dot{\mathcal{P}} = \mathcal{Q}_{1.6/2}(\mathcal{P})$ on the spaces of Poisson bi-vectors \mathcal{P} , see [8, 9] and [1, 2], building a sufficient set of skew Leibniz graphs in the r.-h. s. of factorisation problem (3) requires two iterations of the above algorithm: 11 Leibniz graphs are produced at the first step and 50 more are added by the second step, making 61 in total. One of the two known solutions of this factorisation problem [2] then consists of 8 skew Leibniz graphs (expanding to 27 Leibniz graphs). In turn, as soon as all the Leibniz rules acting on the Jacobiators are processed and every Jacobiator vertex is expanded via (2), the right-hand side $\diamond(\mathcal{P}, \text{Jac}(\mathcal{P}))$ equals the sum of 39 Kontsevich graphs which are assembled into the 9 totally skew-symmetric terms in the left-hand side $\llbracket \mathcal{P}, \mathcal{Q}_{1.6/2} \rrbracket$.

Example 8 (The 5-wheel). Consider the factorisation problem $\llbracket \mathcal{P}, \text{Or}(\gamma_5) \rrbracket = \diamond(\mathcal{P}, \text{Jac}(\mathcal{P}))$ for the pentagon-wheel deformation $\dot{\mathcal{P}} = \text{Or}(\gamma_5)(\mathcal{P})$ of Poisson bi-vectors \mathcal{P} , see [10, 13] and [6]. The ninety skew Kontsevich graphs encoding the bi-vector $\text{Or}(\gamma_5)$ are obtained by taking all the admissible orientations of two $(5+1)$ -vertex graphs γ_5 , one of which is the pentagon wheel with five spokes, the other graph complementing the former to a cocycle in the non-oriented graph complex. By running the iterations of the above algorithm for self-expanding construction of the Leibniz tri-vector graphs in this factorisation problem, we achieve stabilisation of the number of such graphs after the seventh iteration, see Table 1 below.

Table 1. The number of skew Leibniz graphs produced iteratively for $\llbracket \mathcal{P}, \text{Or}(\gamma_5) \rrbracket$.

No. iteration i	1	2	3	4	5	6	7	8
# of graphs	1609	14937	41122	54279	56409	56594	56600	56600
of them new	all	+13328	+26185	+13157	+2130	+185	+6	none

2. Generating the Kontsevich multi-vector graphs

Let us return to problem (3): it is the ansatz for bi-vector Kontsevich graphs $\mathcal{Q}(\mathcal{P})$ with k internal vertices, as well as the Kontsevich 1-vectors \mathcal{X} with $k-1$ internal vertices (to detect trivial terms $\mathcal{Q}(\mathcal{P}) = \llbracket \mathcal{P}, \mathcal{X} \rrbracket$) which must be generated at a given k . (At $1 \leq k \leq 4$, one can still

expand with respect to *all* the Leibniz graphs in the r.-h.s. of (3), not employing the iterative algorithm from §1.2. So, a generator of the Kontsevich (and Leibniz) tri-vectors will also be described presently.)

The Kontsevich graphs corresponding to n -vectors are those graphs with n sinks (each containing the respective argument of n -vector) in which exactly one arrow comes into each sink, so that the order of the differential operator encoded by an n -vector graph equals one w.r.t. each argument, and which are totally skew-symmetric in their n arguments. Let us explain how one can economically obtain the set of one-vectors and skew-symmetric bi- and tri-vectors with k internal vertices in three steps (including graphs with eyes $\bullet \rightleftharpoons \bullet$ but excluding graphs with tadpoles). This approach can easily be extended to the construction of n -vectors with any $n \geq 1$.

2.1. *One-vectors*

Each one-vector under study is encoded by a Kontsevich graph with one sink. Since the sink has one incoming arrow, there is an internal vertex as the tail of this incoming arrow. The target of another edge issued from this internal vertex can be any internal vertex other than itself.

Step 1. Generate all Kontsevich graphs on $k - 1$ internal vertices and one sink (i.e. graphs including those with eyes yet excluding those with tadpoles, and not necessarily of differential order one with respect to the sink content).

Step 2. For every such graph with $k - 1$ internal vertices, add the new sink and make it a target of the old sink, which itself becomes the k th internal vertex. Now run over the $k - 1$ internal vertices excluding the old sink and – via the Leibniz rule – make every such internal vertex the second target of the old sink.

2.2. *Bi-vectors*

There are two cases in the construction of bi-vectors encoded by the Kontsevich graphs. At all $k \geq 1$ the first variant is referred to those graphs with an internal vertex that has both sinks as targets.

Variant 1: Step 1. Generate all k -vertex graphs on $k - 1$ internal vertices and one sink.

Variant 1: Step 2. For every such graph, add two new sinks and proclaim them as targets of the old sink.

Note that the obtained graphs are skew-symmetric.

The second variant produces those graphs which contain two internal vertices such that one has the first sink as target and the other has the second sink as target. The second target of either such internal vertex can be any internal vertex other than itself. Note that for $k = 1$ only the first variant applies.

Variant 2: Step 1. Generate all k -vertex Kontsevich graphs on $k - 2$ internal vertices and two sinks. These sinks now become the $(k - 1)$ th and k th internal vertices.

Variant 2: Step 2. For every such graph, add two new sinks, make the first new sink a target of the first old sink and make the second new sink a target of the second old sink. Now run over the $k - 1$ internal vertices excluding the first old sink, each time proclaiming an internal vertex the second target of the first old sink. Simultaneously, run over the $k - 1$ internal vertices excluding the second old sink and likewise, declare an internal vertex to be the second target of the second old sink.

Variant 2: Step 3. Skew-symmetrise each graph with respect to the content of two sinks using (1).

2.3. *Tri-vectors*

For $k \geq 3$, there exist two variants of tri-vectors. The first variant at all $k \geq 2$ yields those Kontsevich graphs with two internal vertices such that one has two of the three sinks as its

targets while another internal vertex has the third sink as one of its targets. The second target of this last vertex can be any internal vertex other than itself. The second variant contains those graphs with three internal vertices such that the first one has the first sink as a target, the second one has the second sink as a target, and the third one has the third sink as a target. For each of these three internal vertices with a sink as target, the second target can be any internal vertex other than itself.

Variant 1: Step 1. Generate all k -vertex Kontsevich graphs on $k - 2$ internal vertices and two sinks.

Variant 1: Step 2. For every such graph, add three new sinks, make the first two new sinks the targets of the first old sink and make the third new sink a target of the second old sink. Now run over the $k - 1$ internal vertices excluding the second old sink and every time declare an internal vertex the second target of the second old sink.

Variant 1: Step 3. Skew-symmetrise all graphs at hand by applying formula (1) to each of them.

Note that for $k = 1$ there are no tri-vectors encoded by Kontsevich graphs and also note that for $k = 2$ only the first variant applies.

Variant 2: Step 1. Generate all Kontsevich graphs on $k - 3$ internal vertices and three sinks.

Variant 2: Step 2. For every such graph, add three new sinks, make the first new sink a target of the first old sink, make the second new sink a target of the second old sink and make the third new sink a target of the third old sink. Now run over the $k - 1$ internal vertices excluding the first old sink and declare every such internal vertex the second target of the first old sink. Independently, run over the $k - 1$ internal vertices excluding the second old sink and declare each internal vertex to be the second target of the second old sink. Likewise, run over the $k - 1$ internal vertices excluding the third old sink and declare each internal vertex to be the second target of the third old sink.

Variant 2: Step 3 Skew-symmetrise all the graphs at hand using (1).

2.4. *Non-iterative generator of the Leibniz n -vector graphs*

The following algorithm generates all Leibniz graphs with a prescribed number of internal vertices and sinks. Note that not only multi-vectors, but also all graphs of arbitrary differential order with respect to the sinks can be generated this way.

Step 1: Generate all Kontsevich graphs of prescribed type on $k - 1$ internal vertices and n sinks, e.g., all n -vectors.

Step 2: Run through the set of these Kontsevich graphs and in each of them, run through the set of its internal vertices v . For every vertex v do the following: re-enumerate the internal vertices so that this vertex is enumerated by $k - 1$. This vertex already targets two vertices, i and j , where $i < j < k - 1$. Proclaim the last, $(k - 1)$ th vertex to be the placeholder of the Jacobiator (see (2)), so we must still add the third arrow. Let a new index ℓ run up to $i - 1$ starting at n if only the n -vectors are produced.⁶ For every admissible value of ℓ , generate a new graph where the ℓ th vertex is proclaimed the third target of the Jacobiator vertex $k - 1$. (Restricting ℓ by $< i$, we reduce the number of possible repetitions in the set of Leibniz graphs. Indeed, for every triple $\ell < i < j$, the same Leibniz graph in which the Jacobiator stands on those three vertices would be produced from the three Kontsevich graphs: namely, those in which the $(k - 1)$ th vertex targets at the ℓ th and i th, at the ℓ th and j th, and at the i th and j th vertices. In these three cases it is the j th, i th, and ℓ th vertex, respectively, which would be appointed by the algorithm as the Jacobiator's third target.)

We use this algorithm to generate the Leibniz tri- and bi-vector graphs: to establish Theorem 6, we list all possible terms in the right-hand side of factorisation problem (3) at $k \leq 4$

⁶ If we want to generate not only n -vectors but all graphs of arbitrary differential orders, then we let ℓ start at 0 (so that the sinks are included).

and then we filter out the improper bi-vectors in the found solutions $\mathcal{Q}(\mathcal{P})$.

Remark 2. There are at least 265,495 Leibniz graphs on 3 sinks and 6 internal vertices of which one is the Jacobiator vertex. When compared with Table 1 on p. 7, this estimate suggests why at large $k \gtrsim 5$, the breadth-first-search iterative algorithm from §1.2 generates a smaller number of the Leibniz tri-vector graphs, namely, only the ones which can in principle be involved in the factorisation under study.

3. Main result

Theorem 6 ($k \leq 4$). *The few-vertex solutions of problem (3) are these (note that disconnected Kontsevich graphs in $\mathcal{Q}(\mathcal{P})$ are allowed!):*

- $k = 1$: The dilation $\dot{\mathcal{P}} = \mathcal{P}$ is a unique, nontrivial and proper solution.
- $k = 2$: No solutions exist (in particular, neither trivial nor improper).
- $k = 3$: There are no solutions: neither Poisson-trivial nor Leibniz bi-vectors.
- $k = 4$: A unique nontrivial and proper solution is the Kontsevich tetrahedral flow $\mathcal{Q}_{1:\frac{6}{2}}(\mathcal{P})$ from Example 2 (see [8, 9] and [1, 2]). There is a one-dimensional space of Poisson trivial (still proper) solutions $\llbracket \mathcal{P}, X \rrbracket$; the Kontsevich 1-vector X on three internal vertices is drawn in [2, App. F]. Intersecting with the former by $\{0\}$, there is a three-dimensional space of improper (still Poisson-nontrivial) solutions of the form $\nabla(\mathcal{P}, \text{Jac}(\mathcal{P}))$.

None of the solutions $\mathcal{Q}(\mathcal{P})$ known so far contains any 2-cycles (or “eyes” $\bullet \rightleftharpoons \bullet$).⁷

We now report a classification of Poisson bi-vector symmetries $\dot{\mathcal{P}} = \mathcal{Q}(\mathcal{P})$ which are given by those Kontsevich graphs $\mathcal{Q} = \text{Or}(\gamma)$ on k internal vertices that can be obtained at $5 \leq k \leq 9$ by orienting k -vertex connected graphs γ without double edges. By construction, this extra assumption keeps only those Kontsevich graphs which may not contain eyes.

We first find such graphs γ that satisfy $d(\gamma) = 0$, then we exclude the coboundaries $\gamma = d(\gamma')$ for some graphs γ' on $k - 1$ vertices and $2k - 3$ edges.

Theorem 7 ($5 \leq k \leq 9$). *Consider the vector space of non-oriented connected graphs on k vertices and $2k - 2$ edges, without tadpoles and without multiple edges. All nontrivial d-cocycles for $5 \leq k \leq 9$ are exhausted by the following ones:*

- $k = 5, 7, 9$: No solutions.
- $k = 6$: A unique solution⁸ is given by the Kontsevich–Willwacher pentagon-wheel cocycle (see Example 1). The established factorisation $\llbracket \mathcal{P}, \text{Or}(\gamma_5) \rrbracket = \diamond(\mathcal{P}, \text{Jac}(\mathcal{P}))$ will be addressed in a separate paper (see [6]).
- $k = 8$: The only solution γ_7 consists of the heptagon-wheel and 45 other graphs (see Table 2, in which the coefficient of heptagon graph is **1** in bold, and [5]).

$$\gamma_5 = \text{pentagon-wheel} + \frac{5}{2} \text{rectangle}.$$

Remark 3. The wheel graphs are built of triangles. The differential d cannot produce any triangle since multiple edges are not allowed. Therefore, all wheel cocycles are nontrivial. Note also that every wheel graph with 2ℓ spokes is invariant under a mirror reflection with respect to a diagonal consisting of two edges attached to the centre. Hence there exists an edge permutation that swaps $2\ell - 1$ pairs of edges. By footnote 1 such graph equals zero.

Acknowledgments

A.V.Kiselev thanks the Organising committee of the international conference ISQS’25 on integrable systems and quantum symmetries (6–10 June 2017 in ČVUT Prague, Czech Republic)

⁷ Finding solutions $\mathcal{Q}(\mathcal{P})$ with tadpoles or extra sinks – with fixed arguments – is a separate problem.

⁸ There are only 12 admissible graphs to build cocycles from; of these 12, as many as 6 are zero graphs. This count shows to what extent the number of graphs decreases if one restricts to only the flows $\mathcal{Q} = \text{Or}(\gamma)$ obtained from cocycles $\gamma \in \ker(d)$ in the non-oriented graph complex.

Table 2. The heptagon-wheel graph cocycle γ_7 .

Graph encoding	Coeff.	Graph encoding	Coeff.
16 17 18 23 25 28 34 38 46 48 57 58 68 78	1	12 13 18 25 26 37 38 45 46 47 56 57 68 78	-7
12 14 18 23 27 35 37 46 48 57 58 67 68 78	-21/8	12 14 16 23 25 36 37 45 48 57 58 67 68 78	77/8
13 14 18 23 25 28 37 46 48 56 57 67 68 78	-77/4	13 16 17 24 25 26 35 37 45 48 58 67 68 78	-7
12 13 15 24 27 35 36 46 48 57 58 67 68 78	-35/8	14 15 17 23 26 28 37 38 46 48 56 57 68 78	49/4
12 13 18 24 26 37 38 46 47 56 57 58 68 78	49/8	12 16 18 27 28 34 36 38 46 47 56 57 58 78	-147/8
14 17 18 23 25 26 35 37 46 48 56 58 67 78	77/8	12 15 16 27 28 35 36 38 45 46 47 57 68 78	-21/8
12 13 18 26 27 35 38 45 46 47 56 57 68 78	-105/8	12 14 18 23 27 35 36 45 46 57 58 67 68 78	-35/8
12 14 18 23 27 36 38 46 48 56 57 58 67 78	7/8	14 15 16 23 26 28 37 38 46 48 57 58 67 78	-49/4
12 14 15 23 27 35 36 46 48 57 58 67 68 78	35/8	12 15 18 23 28 34 37 46 48 56 57 67 68 78	105/8
12 13 14 27 28 36 38 46 47 56 57 58 68 78	-49/8	12 14 17 23 26 37 38 46 48 56 57 58 68 78	-49/8
12 13 18 25 27 34 36 47 48 56 58 67 68 78	35/4	12 16 18 25 27 35 36 37 45 46 48 57 68 78	49/16
12 13 14 25 26 36 38 45 47 57 58 67 68 78	-119/16	12 13 18 25 27 35 36 46 47 48 56 57 68 78	7
12 13 15 24 28 36 38 47 48 56 57 67 68 78	49/8	12 14 18 25 28 34 36 38 47 57 58 67 68 78	-7
12 13 14 23 28 37 46 48 56 57 58 67 68 78	77/4	12 16 18 25 27 35 36 37 45 46 48 58 67 78	-77/16
12 15 17 25 26 35 36 38 45 47 48 67 68 78	-49/8	12 14 18 23 27 35 38 46 47 57 58 67 68 78	77/4
13 15 18 24 26 28 37 38 46 47 56 57 68 78	-49/4	12 14 15 23 27 36 38 46 48 57 58 67 68 78	35/2
13 14 18 25 26 28 36 38 47 48 56 57 67 78	-49/4	12 13 18 25 27 34 36 46 48 57 58 67 68 78	-105/8
12 14 18 23 28 35 37 46 48 56 57 67 68 78	-7	12 15 16 25 27 35 36 38 46 47 48 57 68 78	-7
12 14 18 23 28 36 38 46 47 56 57 58 67 78	-7	12 13 16 25 28 34 37 47 48 57 58 67 68 78	-147/16
12 15 16 25 27 35 36 38 46 47 48 58 67 78	49/8	12 13 17 25 26 35 37 45 46 48 58 67 68 78	-77/4
12 14 18 23 28 36 37 46 47 56 57 58 68 78	49/8	12 14 17 23 27 35 38 46 48 57 58 67 68 78	-49/8
12 13 15 26 27 35 36 45 47 48 58 67 68 78	-7	12 13 15 26 28 35 37 45 46 47 58 67 68 78	-7/4
12 13 18 24 28 35 38 46 47 57 58 67 68 78	7	12 14 18 23 26 36 38 47 48 56 57 58 67 78	-7

for a warm atmosphere during the meeting. The authors are grateful to M. Kontsevich and T. Willwacher for helpful discussion. We also thank Center for Information Technology of the University of Groningen for providing access to Peregrine high performance computing cluster. This research was supported in part by JBI RUG project 106552 (Groningen, The Netherlands). A part of this research was done while R. Buring and A. V. Kiselev were visiting at the IHÉS (Bures-sur-Yvette, France) and A. V. Kiselev was visiting at the MPIM (Bonn, Germany).

Appendix A. How the orientation mapping Or is calculated

The algorithm lists all ways in which a given non-oriented graph can be oriented in such a way that it becomes a Kontsevich graph on two sinks. It consists of two steps:

- (i) choosing the source(s) of the two arrows pointing at the first and second sink, respectively;
- (ii) orienting the edges between the internal vertices in all admissible ways, so that only Kontsevich graphs are obtained.

Step 1. Enumerate the k vertices of a given non-oriented, connected graph using $2, \dots, k+1$. They become the internal vertices of the oriented graph. Now add the two sinks to the non-oriented graph, the sinks enumerated using 0 and 1. Let a and b be a non-strictly ordered ($a \leq b$) pair of internal vertices in the graph. Extend the graph by oriented edges $a \rightarrow 0$ and $b \rightarrow 1$ from vertices a and b to the sinks 0 and 1, respectively.

Remark 4. The choice of such a base pair, that is, the vertex or vertices from which two arrows are issued to the sinks, is an external input in the orientation procedure. Let us agree that if, at any step of the algorithm, a contradiction is achieved so that a graph at hand cannot be of Kontsevich type, the oriented graph draft is discarded; one proceeds with the next options in that loop, or if the former loop is finished, with the next level-up loops, or – having returned to

the choice of base vertices – with the next base. In other words, we do not exclude in principle a possibility to have no admissible orientations for a particular choice of the base for a given non-oriented graph.

Notation. Let v be an internal vertex. Recalling from p. 6 the notation for the set $N(v)$ of neighbours of v , the (initially empty) set $H(v)$ of arrowheads of oriented edges issued from the vertex v , and the (initially empty) set $T(v)$ of tails for oriented edges pointing at v , we now put by definition $F(v) := N(v) \setminus (H(v) \cup T(v))$. In other words, $F(v)$ is the subset of neighbours connected with v by a non-oriented edge.

Step 2.1. Inambiguous orientation of (some) edges. Here we use that every internal vertex of a Kontsevich graph should be the tail of exactly two outgoing arrows. We run over the set of all internal vertices v . For every vertex such that the number of elements $\#H(v) = 2$, proclaim $T(v) := N(v) \setminus H(v)$, whence $F(v) = \emptyset$. If for a vertex v we have that $\#H(v) = 1$ and $\#F(v) = 1$, then include $F(v) \hookrightarrow H(v)$, that is, convert a unique non-oriented edge touching v into an outgoing edge issued from this vertex. If $\#H(v) = 0$ and $\#F(v) = 2$, also include $F(v) \hookrightarrow H(v)$, effectively making both non-oriented edges outgoing from v .

Repeat the three parts of Step 2.1 while any of the sets $F(v)$, $T(v)$, or $S(v)$ is modified for at least one internal vertex v unless a contradiction is revealed. Summarising, Step 2.1 amounts to finding the edge orientations which are implied by the choice of the base pair a, b and by all the orientations of edges fixed earlier.

Step 2.2. Fixing the orientation of (some) remaining edges. Choose an internal vertex v such that $H(v) < 2$ and such that $H(v) \neq \emptyset$ or $T(v) \neq \emptyset$, that is, choose a vertex that is not yet equipped with two outgoing edges and that is attached to an oriented edge. If $\#H(v) = 1$, then run over the non-empty set $F(v)$: for every vertex w in $F(v)$, include $\{w\} \hookrightarrow H(v)$ and start over at Step 2.1. Otherwise, i.e. if $H(v) = \emptyset$, run over all ordered pairs (u, v) of vertices in the set $F(v)$: for every such pair, make $H(v) := \{u, w\}$ and start over at Step 2.1.

By realising Steps 1 and 2 we accumulate the sum of fully oriented Kontsevich graphs.

References

- [1] Bouisaghouane A and Kiselev A V 2017 Do the Kontsevich tetrahedral flows preserve or destroy the space of Poisson bi-vectors? *J. Phys.: Conf. Ser.* **804** 012008 (*Preprint arXiv:1609.06677* [q-alg])
- [2] Bouisaghouane A, Buring R and Kiselev A 2017 The Kontsevich tetrahedral flow revisited *J. Geom. Phys.* **119** 272–85 (*Preprint arXiv:1608.01710* [q-alg])
- [3] Buring R and Kiselev A V 2017 On the Kontsevich \star -product associativity mechanism *PEPAN Letters* **14:2** 403–7 (*Preprint arXiv:1602.09036* [q-alg])
- [4] Buring R and Kiselev A V 2017 Software modules and computer-assisted proof schemes in the Kontsevich deformation quantization *Preprint IHÉS/M/17/05*
- [5] Buring R, Kiselev A V and Rutten N J 2017 The heptagon-wheel cocycle in the Kontsevich graph complex *J. Nonlin. Math. Phys.*
- [6] Buring R, Kiselev A V and Rutten N J 2017 The Kontsevich–Willwacher pentagon-wheel symmetry of Poisson structures
- [7] Dolgushev V A, Rogers C L and Willwacher T H 2015 Kontsevich’s graph complex, GRT, and the deformation complex of the sheaf of polyvector fields *Ann. Math.* **182:3** 855–943 (*Preprint arXiv:1211.4230* [math.KT])
- [8] Kontsevich M 1997 Formality conjecture *Deformation theory and symplectic geometry* June 17–21 1996 Ascona *Math. Phys. Stud.* vol 20 ed D Sternheimer et al (Dordrecht: Kluwer Acad. Publ.) pp 139–56
- [9] Kontsevich M 2017 Derived Grothendieck–Teichmüller group and graph complexes [after T. Willwacher] *Séminaire Bourbaki* 1126
- [10] Kontsevich M 2017 Private communication
- [11] Khoroshkin A, Willwacher T and Živković M 2017 Differentials on graph complexes *Adv. Math.* **307** 1184–214 (*Preprint arXiv:1411.2369* [q-alg])
- [12] Willwacher T 2015 M. Kontsevich’s graph complex and the Grothendieck–Teichmüller Lie algebra *Invent. Math.* **200:3** 671–760 (*Preprint arXiv:1009.1654* [q-alg])
- [13] Willwacher T 2017 Private communication