Algebraic tools and symbolic package for the study of order conditions for sampling the invariant measure of ergodic Stochastic Differential Equations

Eugen Bronasco
Supervisor: Gilles Vilmart, Adrien Laurent

Master Thesis
University of Geneva

Feb. 2020 – Jan. 2021

In this master thesis, we describe the algebraic structures related to the Butcher series, study order conditions for sampling the invariant measure of ergodic Stochastic Differential Equations, and introduce a new symbolic package PyTreeHopf developed on the basis of SymPy to perform automatic operations on exotic trees. The first part of the master thesis is devoted to the algebraic framework for the study of Butcher series. We introduce a new unique homomorphism between Butcher and Substitution groups and study its relation to the Hopf and pre-Lie algebra homomorphisms. The second part of the master thesis is focused on the theory of exotic Butcher series that describes the order conditions for invariant measure of ergodic Stochastic Differential Equations. We study the order conditions for Runge-Kutta methods and prove that certain order conditions can be expressed in terms of conditions for lower orders. We also generate order conditions for Runge-Kutta methods of order 4 using the new symbolic package. The third part describes several algorithms for exotic trees and the structure of the symbolic package PyTreeHopf. The package can be used to implement algorithms that involve operations on trees. The implemented operations include the grafting of trees, composition coproduct, substitution coproduct, and several operations discussed in the thesis.

Contents

1 Introduction

In this Master thesis, we study the structures related to the concept of a Butcher series (B-series) that is defined in Section 2. Butcher series were first introduced by John Butcher in the 1960s as a way to study Runge-Kutta methods which is a widely used class of numerical methods. Butcher series presented a general way of finding order conditions for Runge-Kutta methods. The work of Butcher on this topic was spread across multiple papers. The paper titled *An algebraic theory of integration methods* (1972) [5] introduces what is now called the Butcher group.

Later, the work of Butcher was noticed by Ernst Hairer and Gerhard Wanner who polished the theory and introduced both Butcher series and the term Butcher group in 1974 [17]. A modern exposition of the theory can be found in [15] Ch.III and a detailed history in [6], [16], [26].

The goal of this Master thesis is to present the algebraic framework used for the study of Butcher series (B-series) in Section 3, describe a generalization called exotic B-series used to find the order conditions for invariant measure of ergodic stochastic differential equations in Section 5 and 6, and develop a symbolic package to perform the combinatorial computations that occur in this context which is discussed in Section 7.

Runge-Kutta methods is a widely used class of numerical methods $X_{n+1} = \Psi(X_n, h, \xi)$ of the form

$$\begin{aligned} Y_i &= X_n + h\sum_{j=1}^{s} a_{ij} f(Y_j) + \sum_{k=1}^{l} d_i^{(k)} \sqrt{h}\xi_n^{(k)},\ i = 1, \dots, s, \\ X_{n+1} &= X_n + h\sum_{i=1}^{s} b_i f(Y_i) + \sqrt{h}\xi_n^{(1)}, \end{aligned} \tag{1}$$

where $a_{ij}, b_i, d_i^{(k)}$ are the coefficients defining the Runge-Kutta method, and $\xi_n^{(k)} \sim \mathcal{N}(0, I_d)$ are independent normally distributed random vectors. For simplicity, let us assume that $l = 1$ and there is only one random vector. It turns out that Runge-Kutta methods of this form are exotic B-series. This allows us to compute order conditions for invariant measure of ergodic SDE for Runge-Kutta methods.

During the work on this thesis, several new results were obtained. In Section 4, we introduce a new unique group homomorphism between the Butcher group and the Substitution group [10], show its relation to a corresponding Hopf algebra homomorphism, and describe its dual which is a unique homomorphism from the free pre-Lie algebra. In Section 6, we reformulate the results from [20], formulate an algorithm to generate order conditions for invariant measure of ergodic SDE, and prove new results related to the order conditions.

In Section 7, we describe the new Python symbolic package based on SymPy, which performs the combinatorial operations on rooted exotic trees automatically. This package allowed us to implement the products in pre-Lie algebras on trees, coproducts in combinatorial Hopf algebras on trees, and the algorithms used to generate the order conditions for invariant measure on ergodic SDE. This package was used to generate order conditions for invariant measure of ergodic SDE for order 4 which was not done before due to the huge number of computations.

2 Definition of B-Series

In this thesis, we study a class of numerical integrators called **Butcher series** methods, which are characterized by the fact that they can be expressed in the following form

$$B(a, hf, y_0) := \sum_{\tau \in \mathcal{T}} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau) F_f(\tau)(y_0),$$

where

$\mathcal{T}$ is the set of isomorphism classes of all rooted trees including the empty tree,

$a : \mathcal{T} \to \mathbb{R}$ is a functional describing the coefficients of the **B-series**,

h is a stepsize of the numerical integrator,

f is a vector field, i.e. $\dot{y} = f(y)$,

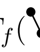
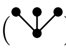
y_0 is an initial value, i.e. $y(0) = y_0$,

$|\tau|$ is the number of vertices in the tree τ,

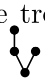
$\sigma(\tau)$ is the size of automorphism group of the tree τ,

$F_f(\tau)$ is the elementary differential operator correspoing to the tree τ.

We need to go into more detail into how the funtion F_f works. The idea of associating rooted trees to elementary differential operators is not new. It was studied by Cayley in 1857 [8] and then rediscovered many times. A few examples should make the association clear.

$F_f(\bullet) = f,$ $F_f($$) = f''(f, f),$ $F_f($$) = f'''(f, f, f),$

$F_f($$) = f'f,$ $F_f($$) = f'f'f,$ $\ldots$

To make it formal, we need to introduce new notations. Let $\tau \in \mathcal{T}$ be a rooted tree and let $\mathcal{F}$ be a set of all forests $\{\tau_1 \cdots \tau_n \,|\, n \in \mathbb{N}, \tau_i \in \mathcal{T}, i = 1, \ldots, n\}$. Then $\tau \in \mathcal{T}$ can be put together by taking a forest $\tau_1 \cdots \tau_n \in \mathcal{F}$ and connecting all the roots in the forest to a new vertex. This forms the tree τ with the new vertex as a root. This can be written as $\tau = [\tau_1 \cdots \tau_n]$. For example,  $= [$ $\bullet]$. Then F_f is recursively defined as

$$F_f([\tau_1, \ldots, \tau_n]) := f^{(n)}(F_f(\tau_1), \ldots, F_f(\tau_n)),$$
$$F_f(\bullet) := f,$$

with $F_f(\emptyset) = y_0$.

The power of B-series comes from the following two facts

- A B-series $B(a, hf, y_0)$ is fully determined by the functional $a : \mathcal{T} \to \mathbb{R}$ with $a(\emptyset) = 1$ which means that we can use combinatorics of trees to study them,
- The exact solution is also a B-series with $a(\tau) = \frac{1}{\tau!}$ where $\tau!$ is the factorial of a tree which is defined as

$$\tau! := |\tau| \cdot \prod_{i=1}^{n} \tau_i!,$$

  where $\tau = [\tau_1, \ldots, \tau_n]$. This means that a B-series has order p if for all trees τ with $|\tau| \leq p$ we have $a(\tau) = \frac{1}{\tau!}$. One can find more details in the book [15] in Chapter III.1.1.

3 Algebraic framework for the study of B-series

In this section, we look at the algebraic structure that arises once we introduce binary operations on B-series. The content of this section is based on [11] and [7].

3.1 Composition Group

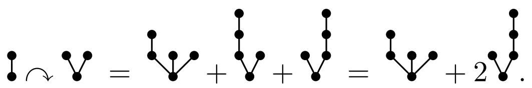
A natural next step is to look at composition of B-series. It turns out that a composition of two B-series is also a B-series. Due to the fact that the B-series $B(a, hf, y_0)$ and $B(b, hf, y_0)$ are determined by the functionals a and b, the composition of B-series determines an operation on these functionals. The operation is called the **composition law**. Thus,

$$B(b, hf, B(a, hf, y_0)) = B(a \cdot b, hf, y_0).$$

To describe the $\cdot$ operation we have to look at the composition of two elementary differential operators $F_f(\tau_1)$ and $F_f(\tau_2)$ for two trees $\tau_1, \tau_2 \in \mathcal{T}$. First, we define the map F_f and the functional a on $\mathcal{F}$. For $\tau \in \mathcal{F}$ we have

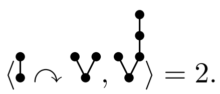
$$F_f(\tau) = F_f(\tau_1) \cdots F_f(\tau_k) \text{ and } a(\tau) = a(\tau_1) \cdots a(\tau_k)$$

where $\tau = \tau_1 \cdots \tau_k$. Composition of elementary differential operators $F_f(\tau_1)$ and $F_f(\tau_2)$ is equal to $F_f(\tau_1 \curvearrowright \tau_2)$ where $\curvearrowright$ is an operation on rooted trees where the root of τ_1 is connected to the vertices of τ_2 in all possible ways. For example,


(2)

This operation can be naturally extended to forests such that every root of the first forest is connected to vertices of the second forest in all possible ways. This operation is called the **grafting of forests**.

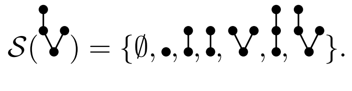
Let us also introduce an inner product such that the elements of $\mathcal{F}$ are orthonormal. For example, based on the equation (2), we have



Then, using the composition of forests and the inner product, we can write a formula for the composition law on the functionals on forests:

$$(a \cdot b)(\tau) := \sum_{\tau_1 \in \mathcal{F}, \tau_2 \in \mathcal{T}} \langle \tau, \tau_1 \curvearrowright \tau_2 \rangle \, \frac{\sigma(\tau)}{\sigma(\tau_1)\sigma(\tau_2)} \, a(\tau_1) b(\tau_2). \tag{3}$$

To make things easier, we can rewrite the formula in purely combinatorial terms. Instead of taking the sum over all possible forests and trees, let us take only the forests and trees for which $\langle \tau, \tau_1 \curvearrowright \tau_2 \rangle \neq 0$. We need to find such pairs of (τ_1, τ_2) that τ can be obtained by connecting the roots of τ_1 to some vertices of τ_2. Let $\mathcal{S}(\tau)$ denote a multiset of all **ordered rooted subtrees** of $\tau \in \mathcal{T}$. The subtrees are **ordered** because we consider all vertices of τ as different even if the corresponding rooted subtrees are the same. The subtrees are **rooted** because the nonempty subtrees must contain the root. For example,


(4)

Let $\tau \setminus s$ for $s \in \mathcal{S}(\tau)$ be the **complement** of s in τ, i.e. a forest obtained by removing the ordered rooted subtree from τ. For example, $[[\bullet]\bullet] \setminus \bullet = [\bullet]\,\bullet$. Notice that the two ordered rooted subtrees $[\bullet]$ in (4) have different complements $[\bullet]$ and $\bullet\,\bullet$. The formula for the composition law can be rewriten in the following way: ([15] Ch.III.1.4)

$$(a \cdot b)(\tau) := \sum_{s \in \mathcal{S}(\tau)} a(\tau \setminus s) b(s).$$

It can be checked that composition law is an associative product on $G_C := \{a : \mathcal{T} \to \mathbb{R} \mid a(\emptyset) = 1\ \}$. Identity in G_C is defined as $e(\tau) = 0$ for all $\tau \neq \emptyset$. Moreover, it can be shown that every function $a \in G_C$ has an inverse. To show it, we first see that for $\tau = \emptyset$, $(a \cdot b)(\tau) = a(\emptyset)b(\emptyset) = 1$. For $\tau \neq \emptyset$, we have

$$(a \cdot b)(\tau) = a(\tau) + b(\tau) + \sum_{\substack{s \in \mathcal{S} \\ s \neq \emptyset, \tau}} a(\tau \setminus s) b(s).$$

We see that $|s| < |\tau|$ and $|\tau \setminus s| < |\tau|$, and, therefore, we can define the inverse recursively as

$$a^{-1}(\tau) = -a(\tau) - \sum_{\substack{s \in \mathcal{S}(\tau) \\ s \neq \emptyset, \tau}} a(\tau \setminus s) a^{-1}(s).$$

It follows that G_C is a group. Let $a, b \in G_C$, let us compute the values of $a \cdot b$ for trees up to order 3.

$$\begin{aligned}
(a \cdot b)(\emptyset) &= a(\emptyset)b(\emptyset), \\
(a \cdot b)(\bullet) &= a(\emptyset)b(\bullet) + a(\bullet)b(\emptyset), \\
(a \cdot b)([\bullet]) &= a(\emptyset)b([\bullet]) + a(\bullet)b(\bullet) + a([\bullet])b(\emptyset), \\
(a \cdot b)([\bullet\bullet]) &= a(\emptyset)b([\bullet\bullet]) + 2a(\bullet)b([\bullet]) + a(\bullet)^2 b(\bullet) + a([\bullet\bullet])b(\emptyset), \\
(a \cdot b)([[\bullet]]) &= a(\emptyset)b([[\bullet]]) + a(\bullet)b([\bullet]) + a([\bullet])b(\bullet) + a([[\bullet]])b(\emptyset).
\end{aligned}$$

3.2 Connes-Kreimer Hopf algebra

The formula for the composition law gives us a hint that there is more structure than can be seen at first glance as noted by [2]. It hints at a coalgebra structure on rooted trees with the coproduct being defined as

$$\Delta_{CK}(\tau) := \sum_{s \in \mathcal{S}(\tau)} (\tau \setminus s) \otimes s,$$

and a counit $\epsilon : \mathcal{T} \to \mathbb{R}$ with $\epsilon(\emptyset) = 1$ and $\epsilon(\tau) = 0$ for $\tau \neq \emptyset$. Together with the concatenation of rooted trees and the unit $\emptyset$, it forms a graded connected bialgebra H_{CK}. We notice that

$$(a \cdot b)(\tau) = \times \circ (a \otimes b) \circ \Delta_{CK}(\tau),$$

where $\times$ is the product in $\mathbb{R}$.

Let $\sqcup$ denote the concatenation product. A **bialgebra** is an algebra and coalgebra in which the product and coproduct respect each other, in our case it can be written as

$$\Delta_{CK}(a \sqcup b) = \Delta_{CK}(a) \sqcup \Delta_{CK}(b),$$

where $(a_1 \otimes a_2) \sqcup (b_1 \otimes b_2) = (a_1 \sqcup b_1) \otimes (a_2 \sqcup b_2)$.

$H_{CK} = \bigoplus_{k=0}^{\infty} H_{CK}^k$ is graded by the number of vertices in a tree. The product and coproduct respect the grading, i.e.

$$\Delta_{CK}(H_{CK}^n) \subset \sum_{k=0}^{n} H_{CK}^k \otimes H_{CK}^{n-k} \text{ and } H_{CK}^n \sqcup H_{CK}^m \subset H_{CK}^{n+m}.$$

An **antipode** is a map $S : H_{CK} \to H_{CK}$ which is the inverse of identity in the group of Hopf algebra endomorphisms with the convolution product. That is

$$\sqcup \circ (\mathrm{id} \otimes S) \circ \Delta_{CK} = u_{CK} \circ \epsilon_{CK}.$$

It is known that a graded connected bialgebra has an antipode which makes it a **Hopf algebra**. We can find the formula for the antipode the same way we found the formula for the inverse of an element in G_C. The antipode has the following formula

$$S_{CK}(\tau) = -\tau - \sum_{\substack{s \in \mathcal{S}(\tau) \\ s \neq \emptyset, \tau}} (\tau \setminus s) S_{CK}(s).$$

Such Hopf algebra is called **Connes-Kreimer Hopf algebra** introduced in 1998 [12]. Notice the similarity between the convolution product of two endomorphisms of H_{CK} and the composition law of two B-series. Using $a(\tau_1 \sqcup \tau_2) = a(\tau_1)a(\tau_2)$, we have $a^{-1} = a \circ S$.

3.3 Substitution Group

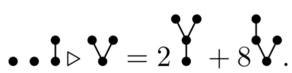
There is another way to combine two B-series. A composition of B-series corresponds to the substitution of the initial value y_0 with another B-series. A similar operation can be done with the vector field hf. It turns out that the result of such an operation is again a B-series. The **substitution law** is defined the following way

$$B(b, B(a, hf, \cdot), y_0) = B(a \star b, hf, y_0).$$

Again, we need to look at how the elementary differentials interact, however, now we need to express $F_{\frac{1}{h}B(a,hf,\cdot)}(\gamma)$ in terms of F_f or look at the operation on forests that would correspond to it. Notice that this requires $a(\emptyset) = 0$ and let us take $a(\bullet) = 1$ for simplicity, i.e. $B(a, hf, y_0) = hf + \cdots$. Let $\mathcal{T}' = \mathcal{T} \setminus \{\emptyset\}$, then such operation on forests looks the following way

$$\tau_1 \triangleright \tau_2 = \sum_{\substack{\tau \in \mathcal{T}' \\ \tau_1 \subset \tau,\ \tau/\tau_1 = \tau_2}} M(\tau_1, \tau_2, \tau)\ \tau,$$

where $M(\tau_1, \tau_2, \tau)$ is the number of ways to insert the forest τ_1 into the tree τ_2 to get τ. Such operation is called **insertion** of forests. Intuitively, it can be understood as a sum over all the possible ways in which the nodes in τ_2 can be substituted by the trees from τ_1. For example,



Then, using the insertion of forests and the inner product, we can write a formula for the substitution law on the functionals on forests.

$$(a \star b)(\tau) := \sum_{\tau_1,\tau_2 \in \mathcal{F}} \langle \tau, \tau_1 \triangleright \tau_2 \rangle \, \frac{\sigma(\tau)}{\sigma(\tau_1)\sigma(\tau_2)} \, a(\tau_1)b(\tau_2). \tag{5}$$

This formula can be rewritten using combinatorics. Let $\mathcal{P}(\tau)$ denote all the possible covering subforests. A covering subforest can be obtained by removing a certain subset of edges. Let $p \in \mathcal{P}(\tau)$, then τ/p is a graph obtained by shrinking every tree of p into a node. Then the substitution law can be written as [10]

$$(a \star b)(\tau) := \sum_{p \in \mathcal{P}(\tau)} a(p)b(\tau/p).$$

Let $G_S^* := \{a : T \to \mathbb{R} \mid a(\emptyset) = 0,\ a(\bullet) = 1\}$, then together with the substitution law, G_S^* is a group with identity $\delta_\bullet$ where the existence of inverses can be shown using the same argument as in the previous subsection and $\delta_\bullet(\bullet) = 1$ and $\delta_\bullet(\tau) = 0$ for all $\tau \neq \bullet$. Let $a, b \in G_S^*$, let us compute the values of $a \star b$ for trees up to order 3.

$$\begin{aligned}
(a \star b)(\emptyset) &= a(\emptyset)b(\emptyset), \\
(a \star b)(\bullet) &= a(\bullet)b(\bullet), \\
(a \star b)\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}\right) &= a\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}\right)b(\bullet) + a(\bullet)^2 b\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}\right), \\
(a \star b)\left(\begin{smallmatrix}\bullet\ \bullet\\ \vee\\ \bullet\end{smallmatrix}\right) &= a\left(\begin{smallmatrix}\bullet\ \bullet\\ \vee\\ \bullet\end{smallmatrix}\right)b(\bullet) + 2a(\bullet)a\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}\right)b\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}\right) + a(\bullet)^3 b\left(\begin{smallmatrix}\bullet\ \bullet\\ \vee\\ \bullet\end{smallmatrix}\right), \\
(a \star b)\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\\ \mid\\ \bullet\end{smallmatrix}\right) &= a\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\\ \mid\\ \bullet\end{smallmatrix}\right)b(\bullet) + 2a(\bullet)a\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}\right)b\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}\right) + a(\bullet)^3 b\left(\begin{smallmatrix}\bullet\\ \mid\\ \bullet\\ \mid\\ \bullet\end{smallmatrix}\right).
\end{aligned}$$

3.4 Calaque, Ebrahimi-Fard & Manchon Hopf algebra

We notice that similar to the group G_C, the group G_S^* has a Hopf algebra associated to it. This Hopf algebra is called the **Calaque, Ebrahimi-Fard & Manchon Hopf algebra** H_{CEM} and was introduced in 2011 [7]. The product of this algebra is the concatenation product $\sqcup$ and, therefore, the group G_S^* is a group of algebra morphisms, $a : H_{CEM} \to \mathbb{R}$. Recall that $a(\bullet) = 1$. So in order for G_S^* to be a group of algebra morphisms, the tree $\bullet$ needs to be the unit of the algebra. Thus, in H_{CEM}, we have $\begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}\ \bullet = \begin{smallmatrix}\bullet\\ \mid\\ \bullet\end{smallmatrix}$. A coalgebra structure on H_{CEM} is again hinted at by the substitution law discussed above. The coproduct is defined in the following way:

$$\Delta_{CEM}(\tau) := \sum_{p \in \mathcal{P}(\tau)} p \otimes (\tau/p),$$

with the counit being the $\delta_\bullet$. It can be shown that $(H_{CEM}, \sqcup, \bullet, \Delta_{CEM}, \delta_\bullet)$ is a bialgebra. Moreover, it is a connected bialgebra graded by the number of edges in the tree. This implies that there is an antipode of the form

$$S_{CEM}(\tau) := -\tau - \sum_{p \in \mathcal{P}(\tau)} pS(\tau/p).$$

and the inverse of an element $a \in G_S^*$ has the form $a^{-1} = a \circ S_{CEM}$.

There is an alternative formula for the coproduct that will be useful in Section 4. Notice that to every covering subforest $p \in \mathcal{P}(\tau)$ there corresponds a subset of edges $\bar{p}$ that was removed to obtain the covering subforest. Thus, the coproduct can be written as

$$\Delta_{CEM}(\tau) := \sum_{p \in \mathcal{P}} (\tau \setminus \bar{p}) \otimes \bar{p}.$$

where $\bar{p}$ is both the set of edges that is removed to obtain p, and the tree obtained by connecting the edges in $\bar{p}$ such that two edges are connected by a vertex if the corresponding ends of the edges are in the same tree of p.

3.5 Pre-Lie algebras

Definition 1. A **pre-Lie algebra** is a vector space A together with a map $\cdot : A \otimes A \to A$ satisfying the relation

$$(x \cdot y) \cdot z - x \cdot (y \cdot z) = (y \cdot x) \cdot z - y \cdot (x \cdot z) \quad \text{for all } x, y, z \in A.$$

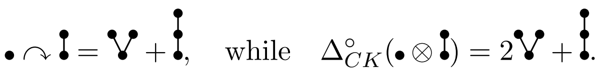
Notice that associative algebras are pre-Lie, but not all pre-Lie algebras are associative. The vector space of rooted trees in $\mathcal{T}$ together with the grafting of trees forms a pre-Lie algebra $\mathcal{A}_{\mathcal{T}}$. It was shown in [9] that this pre-Lie algebra is a free pre-Lie algebra with one generator. This means that it has the universal property. Thus, given another pre-Lie algebra B, there is a homomorphism ψ from $\mathcal{A}_{\mathcal{T}}$ to B that is fully determined by $\psi(\bullet)$.

The homomorphism is fully determined by its value at $\bullet$ because every tree can be written as a grafting product of $\bullet$ with itself. For example,  $= \bullet \curvearrowright (\bullet \curvearrowright \bullet) - (\bullet \curvearrowright \bullet) \curvearrowright \bullet$. This can be proven by induction on the number of vertices and the number of edges going to the root.

Let $A_C := H^{\circ}_{CK}$ and $A_S := H^{\circ}_{CEM}$ be the dual algebras with products the duals of the respective coproducts. They are defined over the vector space of forests in $\mathcal{F}$ with A_S not containing the empty forest. Denote the products as Δ°_{CK} and Δ°_{CEM}, respectively. Then

$$\Delta^{\circ}_{CK}(\tau_1 \otimes \tau_2) := \sum_{\substack{\tau \in T,\ V(\tau) = V_1 \sqcup V_2 \\ \tau_1 = \tau_{|V_1}, \tau_2 = \tau_{|V_2} \\ \text{root} \in V_2}} \tau$$

is a sum that lists all the trees that can be split into a rooted subtree τ_2 with remainder being equal to τ_1. Notice that $\Delta^{\circ}_{CK} \neq \curvearrowright$, for example,



However, these products are equal under a different normalization. To change the normalization, we use the automorphism $A_{\sigma}(\tau) = \sigma(\tau)\tau$. So we have

$$\curvearrowright = A_{\sigma}^{-1} \circ \Delta^{\circ}_{CK} \circ (A_{\sigma} \otimes A_{\sigma}).$$

A similar situation is true for the A_S algebra. We have

$$\triangleright = A_{\sigma}^{-1} \circ \Delta^{\circ}_{CEM} \circ (A_{\sigma} \otimes A_{\sigma}).$$

Recall the first formulas (3) and (5) of composition and substitution laws discussed in this thesis. They are defined using the duality discussed above. The coefficients $\frac{\sigma(\tau)}{\sigma(\tau_1)\sigma(\tau_2)}$ have their origin in the identities from above. Notice that $\mathcal{A}_{\mathcal{T}}$ is a subalgebra of $\mathcal{A}_C$.

4 The new group homomorphism between G_C and G_S^*

In this section, we search for a group homomorphism $\Phi^* : G_C \to G_S^*$. Thus, it needs to satisfy the following identity:

$$\Phi^*(a \cdot b) = \Phi^*(a) \star \Phi^*(b),$$

where $\Phi^*(a) = a \circ \Phi$ with $\Phi : H_{CEM} \to H_{CK}$.

Proposition 1. *$\Phi^* : G_C \to G_S^*$ is a group homomorphism if and only if Φ is a coalgebra homomorphism, $\Phi : H_{CEM} \to H_{CK}$.*

Proof. Let $\Phi^* : G_C \to G_S^*$ be a group homomorphism. Then for all $a, b \in G_C$ we have

$$\begin{aligned} \Phi^*(a \cdot b) &= \Phi^*(a) \star \Phi^*(b) \\ (a \cdot b) \circ \Phi &= (a \circ \Phi) \star (b \circ \Phi) \\ m \circ (a \otimes b) \circ \Delta_{CK} \circ \Phi &= m \circ (a \otimes b) \circ (\Phi \otimes \Phi) \circ \Delta_{CEM} \\ \Delta_{CK} \circ \Phi &= (\Phi \otimes \Phi) \circ \Delta_{CEM} \end{aligned}$$

Moreover, $\Phi^*(\mathrm{id}_C) = \mathrm{id}_S$ implies that $\Phi(\bullet) = \emptyset$ and, therefore, $\epsilon_{CEM} = \epsilon_{CK} \circ \Phi$.
This implies that Φ is a coalgebra homomorphism. The converse is also true. □

Thus, let us focus our attention on finding a coalgebra homomorphism $\Phi : H_{CEM} \to H_{CK}$. To simplify the matters, let $\Phi(\tau_1 \sqcup \tau_2) = \Phi(\tau_1) \sqcup \Phi(\tau_2)$. This implies that $\Phi : H_{CEM} \to H_{CK}$ is a Hopf algebra homomorphism.

To find such a Hopf algebra homomorphism we need to find a value $\Phi(\tau)$ for every rooted tree $\tau \in H_{CEM}$. We already know from the proof that $\Phi(\bullet) = \emptyset$.

For all the other $\tau \in H_{CEM}$, we will use the following relation

$$\Delta_{CK} \circ \Phi(\tau) = (\Phi \otimes \Phi) \circ \Delta_{CEM}(\tau).$$

Substitute the coproducts with the respective formulas

$$\begin{aligned} &\emptyset \otimes \Phi(\tau) + \Phi(\tau) \otimes \emptyset + \sum_{\substack{s \in \mathcal{S}(\Phi(\tau)) \\ s \neq \emptyset, \Phi(\tau)}} (\Phi(\tau) \setminus s) \otimes s = \\ &\qquad = \Phi(\bullet) \otimes \Phi(\tau) + \Phi(\tau) \otimes \Phi(\bullet) + \sum_{\substack{p \in \mathcal{P}(\tau) \\ p \neq \emptyset, \tau}} \Phi(\tau \setminus \bar{p}) \otimes \Phi(\bar{p}), \end{aligned}$$

use the fact that $\Phi(\bullet) = \emptyset$ to get

$$\begin{aligned} &\emptyset \otimes \Phi(\tau) + \Phi(\tau) \otimes \emptyset + \sum_{\substack{s \in \mathcal{S}(\Phi(\tau)) \\ s \neq \emptyset, \Phi(\tau)}} (\Phi(\tau) \setminus s) \otimes s = \\ &\qquad = \emptyset \otimes \Phi(\tau) + \Phi(\tau) \otimes \emptyset + \sum_{\substack{p \in \mathcal{P}(\tau) \\ p \neq \emptyset, \tau}} \Phi(\tau \setminus \bar{p}) \otimes \Phi(\bar{p}), \end{aligned}$$

and after the cancelations we get

$$\sum_{\substack{s \in \mathcal{S}(\Phi(\tau)) \\ s \neq \emptyset, \Phi(\tau)}} (\Phi(\tau) \setminus s) \otimes s = \sum_{\substack{p \in \mathcal{P}(\tau) \\ p \neq \emptyset, \tau}} \Phi(\tau \setminus \bar{p}) \otimes \Phi(\bar{p}). \tag{6}$$

We can write it as $\bar{\Delta}_{CK} \circ \Phi(\tau) = (\Phi \otimes \Phi) \circ \bar{\Delta}_{CEM}$ where $\bar{\Delta}_{CK}$ and $\bar{\Delta}_{CEM}$ denote the middle terms of the respective coproducts. An important property of $\bar{\Delta}_{CK}(\tau)$ and $\bar{\Delta}_{CEM}(\tau)$ is that their terms are the tensor products of trees that have less vertices than the tree τ. This implies that the RHS of (6) can be computed recursively and, therefore, RHS is known. What remains is to find such $\Phi(\tau)$ that the middle terms of its CK coproduct are equal to the RHS. A few simple computations for this new homomorphism Φ show that

Notice that the map Φ is not injective. It was also noticed that for all trees τ with $|\tau| \geq 3$ the RHS of (6) should contain a term of the form

$$(\sum_{e \in E(\tau)} \Phi(\tau \setminus e)) \otimes \Phi\left(\begin{smallmatrix}\bullet\\|\\\bullet\end{smallmatrix}\right) = (\sum_{e \in E(\tau)} \Phi(\tau \setminus e)) \otimes \bullet,$$

and the LHS of (6) should also contain a term $\gamma \otimes \bullet$ where γ is a linear combination of forests such that $\Phi(\tau) = [\gamma]$. This implies that

$$\Phi(\tau) = [\sum_{e \in E(\tau)} \Phi(\tau \setminus e))] = \sum_{e \in E(\tau)} [\Phi(\tau \setminus e)].$$

Therefore, if such homomorphism exists, it must be unique. We now prove that this formula actually works.

Lemma 1. *Let $\overline{\Delta}^*_{CK}(\tau)$ correspond to the coproduct without the $\tau \otimes \emptyset$ term. Then*

$$\overline{\Delta}^*_{CK}([\tau_1 \cdots \tau_n]) = (id_{CK} \otimes [\cdot]) \circ \Delta_{CK}(\tau_1 \cdots \tau_n).$$

Proof. We use the formula for $\bar{\Delta}^*_{CK}$.

$$\begin{aligned}
\overline{\Delta}^*_{CK}([\tau_1 \cdots \tau_n]) &= \sum_{\substack{s \in S([\tau_1 \cdots \tau_n]) \\ s \neq \emptyset}} ([\tau_1 \cdots \tau_n] \setminus s) \otimes s = \\
&= \sum_{\substack{s \in S([\tau_1 \cdots \tau_n]) \\ \text{root} \in s}} ([\tau_1 \cdots \tau_n] \setminus s) \otimes s = \\
&= \sum_{s \in S(\tau_1 \cdots \tau_n)} (\tau_1 \cdots \tau_n \setminus s) \otimes [s] = \\
&= (\mathrm{id}_{CK} \otimes [\cdot]) \circ \sum_{s \in S(\tau_1 \cdots \tau_n)} (\tau_1 \cdots \tau_n \setminus s) \otimes s = \\
&= (\mathrm{id}_{CK} \otimes [\cdot]) \circ \Delta_{CK}(\tau_1 \cdots \tau_n).
\end{aligned}$$

This finishes our proof. $\square$

The next proposition presents a recursive formula for a coalgebra homomorphism $\Phi : H_{CEM} \to H_{CK}$.

Proposition 2. *Let $\Phi : H_{CEM} \to H_{CK}$ be defined as*

$$\Phi(\tau) := \sum_{e \in E(\tau)} [\Phi(\tau \setminus e)],$$

where $E(\tau)$ are the edges of τ. The operation $\tau \setminus e$ for an edge e corresponds to cutting the edge e from the tree τ which creates a forest with two trees.

Then Φ is the unique coalgebra homomorphism.

Proof. We first check that

and it turns out to be true. Now we have to prove that the relation

$$\overline{\Delta}^*_{CK} \circ \Phi(\tau) = (\Phi \otimes \Phi) \circ \overline{\Delta}^*_{CEM}(\tau)$$

is true for all $\tau \in H_{CEM}$. Take $\tau \in H_{CEM}$ and assume that the relation is true for all trees ν with $|\nu| < |\tau|$.

$$\overline{\Delta}^*_{CK} \circ \Phi(\tau) = \overline{\Delta}^*_{CK}\big(\sum_{e \in E(\tau)} [\Phi(\tau \setminus e)]\big) = \sum_{e \in E(\tau)} \overline{\Delta}^*_{CK}([\Phi(\tau \setminus e)]) =$$

(apply Lemma 1)

$$= \sum_{e \in E(\tau)} (\mathrm{id}_{CK} \otimes [\cdot]) \circ \Delta_{CK} \circ \Phi(\tau \setminus e) =$$

(apply inductive assumption)

$$= \sum_{e \in E(\tau)} (\mathrm{id}_{CK} \otimes [\cdot]) \circ (\Phi \otimes \Phi) \circ \Delta_{CEM}(\tau \setminus e) =$$

$$= \sum_{e \in E(\tau)} (\Phi \otimes [\cdot] \circ \Phi) \circ \Big(\sum_{p \in P(\tau \setminus e)} ((\tau \setminus e) \setminus \bar{p}) \otimes \bar{p} \Big) =$$

$$= \sum_{\substack{e \in E(\tau) \\ p \in P(\tau \setminus e)}} \Big(\Phi((\tau \setminus e) \setminus \bar{p}) \otimes [\Phi(\bar{p})] \Big) =$$

(note that $\{(\tau \setminus e) \setminus \bar{p} \,:\, p \in P(\tau \setminus e)\} = \{\tau \setminus \bar{p} \,:\, p \in P(\tau), e \in \bar{p}\}$)

$$= \sum_{\substack{e \in E(\tau) \\ p \in P(\tau), e \in \bar{p}}} \Big(\Phi(\tau \setminus \bar{p}) \otimes [\Phi(\bar{p} \setminus e)] \Big) =$$

(we revert the order in which we choose e and p)

$$= \sum_{\substack{p \in P(\tau) \\ p \neq \emptyset}} \sum_{e \in E(p)} \Big(\Phi(\tau \setminus \bar{p}) \otimes [\Phi(\bar{p} \setminus e)] \Big) =$$

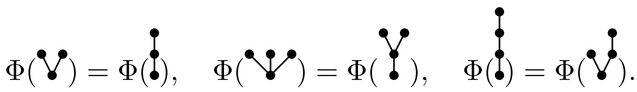
$$= \sum_{\substack{p \in P(\tau) \\ p \neq \emptyset}} \Phi(\tau \setminus \bar{p}) \otimes \Big(\sum_{e \in E(\tau)} [\Phi(\bar{p} \setminus e)] \Big) =$$

$$= \sum_{\substack{p \in P(\tau) \\ p \neq \emptyset}} \Phi(\tau \setminus \bar{p}) \otimes \Phi(\bar{p}) =$$

$$= (\Phi \otimes \Phi) \circ \Big(\sum_{\substack{p \in P(\tau) \\ p \neq \emptyset}} (\tau \setminus \bar{p}) \otimes \bar{p} \Big) = (\Phi \otimes \Phi) \circ \overline{\Delta}^*_{CEM}(\tau).$$

Thus, the relation is true for all $\tau \in H_{CEM}$ by induction which proves that Φ is a coalgebra homomorphism. $\square$

We notice an interesting property in the values we have computed. It seems that if a tree can be transformed into another tree by changing the root then these two trees have the same value under Φ. For example,



This property can be proven using the formula for Φ that we have found in the previous proposition.

Proposition 3. *If τ can be transformed into $\hat{\tau}$ by changing the root along an edge (i.e. such that there exists an edge $e \in E(\tau)$ such that $o(e)$ is root in τ and $t(e)$ is root in $\hat{\tau}$) then $\Phi(\tau) = \Phi(\hat{\tau})$.*

Proof. Let $\tau^{\hat{e}}$ denote the tree τ with the root changed along the edge $\hat{e}$. Take a tree τ and assume that $\Phi(\nu) = \Phi(\nu^{\hat{e}})$ for all ν and $\hat{e} \in E(\nu)$ with $o(\hat{e}) =$ root such that $|\nu| < |\tau|$. Notice that for $e \in E(\tau)$ we have $\tau \setminus e = \tau_1 \sqcup \tau_2$, i.e. $\tau \setminus e$ is a forest with two trees. Let the root be always in the left one. We can assume that as the concatenation of trees is commutative. Choose $\hat{e} \in E(\tau)$ such that $o(\hat{e}) =$ root. Then

$$\Phi(\tau) = \sum_{e \in E(\tau)} [\Phi(\tau \setminus e)] = [\Phi(\tau \setminus \hat{e})] + \sum_{\substack{e \in E(\tau) \\ e \neq \hat{e}}} [\Phi(\tau \setminus e)] =$$

$$= [\Phi(\tau \setminus \hat{e})] + \sum_{\substack{e \in E(\tau) \\ e \neq \hat{e}}} [\Phi(\tau_1) \sqcup \Phi(\tau_2)] =$$

$$= [\Phi(\tau \setminus \hat{e})] + \sum_{\substack{e \in E(\tau) \\ e \neq \hat{e}}} [\Phi(\tau_1^{\hat{e}}) \sqcup \Phi(\tau_2)] =$$

$$= [\Phi(\tau^{\hat{e}} \setminus \hat{e})] + \sum_{\substack{e \in E(\tau^{\hat{e}}) \\ e \neq \hat{e}}} [\Phi(\tau_1^{\hat{e}}) \sqcup \Phi(\tau_2)] =$$

$$= [\Phi(\tau^{\hat{e}} \setminus \hat{e})] + \sum_{\substack{e \in E(\tau^{\hat{e}}) \\ e \neq \hat{e}}} [\Phi(\tau^{\hat{e}} \setminus e)] =$$

$$= \sum_{e \in E(\tau^{\hat{e}})} [\Phi(\tau^{\hat{e}} \setminus e)] = \Phi(\tau^{\hat{e}}).$$

Thus, the proposition is proved by induction. □

Let τ_1 and τ_2 be two trees that give the same tree after we forget their roots. Then we can write $\tau_1 \sim_r \tau_2$ and due to the last proposition we see that if $\tau_1 \sim_r \tau_2$ then $\Phi(\tau_1) = \Phi(\tau_2)$. Let $\hat{H}_{CEM} := H_{CEM}/\sim_r$ be the CEM Hopf algebra on non-rooted trees. The operations in this algebra have the same definition as the original definitions because the original definitions do not depend on the existence of the root. Thus, $\hat{\Phi} : \hat{H}_{CEM} \to H_{CK}$ is a Hopf algebra homomorphism defined the following way. Let τ be a non-rooted tree and $v \in V(\tau)$ be a vertex of τ. Then $r_v(\tau)$ is a rooted tree obtained by choosing the vertex v as the root of τ. Then $\hat{\Phi}(\tau) = \Phi(r_v(\tau))$ where v is any vertex in τ. The map $\hat{\Phi}$ is well-defined. Now let us prove two technical lemmas that will be used to show that $\hat{\Phi}$ is injective.

Lemma 2. *Let τ be a labeled non-rooted tree and v a vertex of τ. We denote by $\tau \setminus v$ the graph obtained by removing v and all edges connected to it from τ. Let $L(\tau)$ be the set containing all leaves of τ (i.e all vertices of degree one) and let $\gamma \to_v \tau$ denote the grafting of tree γ to vertex v of τ. Then*

$$\{\tau\} = \bigcap_{v \in L(\tau)} \{v \to_u (\tau \setminus v) \mid u \in V(\tau \setminus v)\}.$$

Proof. It is trivial to show that $\{\tau\} \subset \bigcap_{v \in L(\tau)} \{v \to_u (\tau \setminus v) \mid u \in V(\tau \setminus v)\}$ as each term of the intersection contains τ because $v \in L(\tau)$ implies $\exists!(v, v^e) \in E(\tau)$ and, therefore, $\tau = v \to_{v^e} (\tau \setminus v)$.

It remains to show that the intersection cannot contain any other trees. Take $v_0 \in L(\tau)$ with $(v_0, v_0^e) \in E(\tau)$. Then for all $\gamma \in \{v_0 \to_u (\tau \setminus v_0) \mid u \in V(\tau \setminus v_0)\}$ such that $\gamma \neq \tau$ we have $(v_0, x) \in E(\gamma)$ with $x \neq v_0^e$ and $deg(v_0) = 1$.

Then for all $v \neq v_0$ and all $\nu \in \{v \to_u (\tau \setminus v) \mid u \in V(\tau \setminus v)\}$ either $deg(v_0) \neq 2$ when v is connected to v_0, or $(v_0, x) \notin E(\nu)$ as $deg(v_0) = 1$ and $(v_0, v_0^e) \in E(\nu)$. In either case $\gamma \neq \nu$. This implies that $\gamma \neq \tau$ is not in the intersection. □

Lemma 3. *Let τ_1 and τ_2 be non-rooted trees. Then*

$$\tau_1 = \tau_2 \iff \sum_{e \in E(\tau_1)} \tau_1 \setminus e = \sum_{e \in E(\tau_2)} \tau_2 \setminus e.$$

Proof. It can be checked to be true for γ with $|\gamma| < 5$. We will use induction on the number of vertices. Assume that the statement is true for all γ with $|\gamma| < n$ and take τ_1 and τ_2 with $|\tau_1| = |\tau_2| = n$. Then the direction ($\Longrightarrow$) is trivial.

Let τ_1 and τ_2 be such that $\sum_{e\in E(\tau_1)} \tau_1 \setminus e = \sum_{e\in E(\tau_2)} \tau_2 \setminus e$. We have

$$\sum_{e\in E(\tau_1)} \tau_1 \setminus e = \sum_{e\in E(\tau_2)} \tau_2 \setminus e$$
$$\Updownarrow$$
$$\sum_{\substack{e\in E(\tau_1)\\ e\neq(v,v^e)\\ v\in L(\tau_1)}} \tau_1 \setminus e + \sum_{v\in L(\tau_1)} \{v\} \sqcup (\tau_1 \setminus v) = \sum_{\substack{e\in E(\tau_2)\\ e\neq(v,v^e)\\ v\in L(\tau_2)}} \tau_2 \setminus e + \sum_{v\in L(\tau_2)} \{v\} \sqcup (\tau_2 \setminus v)$$
$$\Downarrow$$
$$\sum_{v\in L(\tau_1)} \{v\} \sqcup (\tau_1 \setminus v) = \sum_{v\in L(\tau_2)} \{v\} \sqcup (\tau_2 \setminus v).$$

Recall that $\tau_1 = \tau_2$ if and only if there is an isomorphism between τ_1 and τ_2 given a labelling of both trees. So label both τ_1 and τ_2.

$\sum_{v\in L(\tau_1)}\{v\} \sqcup (\tau_1 \setminus v) = \sum_{v\in L(\tau_2)}\{v\} \sqcup (\tau_2 \setminus v)$ implies that for every $v \in L(\tau_1)$ there exists a $u \in L(\tau_2)$ such that $(\tau_1 \setminus v) \cong (\tau_2 \setminus u)$ and vice versa. But then

$$\{\tau_1\} = \bigcap_{v\in L(\tau_1)} \{v \to_u (\tau_1 \setminus v) \mid u \in V(\tau_1 \setminus v)\} \cong_e$$
$$\cong_e \bigcap_{v\in L(\tau_2)} \{v \to_u (\tau_2 \setminus v) \mid u \in V(\tau_2 \setminus v)\} = \{\tau_2\},$$

where $\cong_e$ denotes the fact that the two sets are isomorphisc and every element in the first set is isomorphic to an element in the second set. Thus, $\tau_1 = \tau_2$ and the lemma is proved. □

Proposition 4. *The homomorphism $\hat{\Phi} : \hat{H}_{CEM} \to H_{CK}$ is injective.*

Proof. We have seen that $\hat{\Phi}$ is injective on trees γ with $|\gamma| < 5$. We will use induction on the number of vertices. Assume that the statement is true for all γ with $|\gamma| < n$ and take τ_1 and τ_2 with $|\tau_1| = |\tau_2| = n$ such that $\hat{\Phi}(\tau_1) = \hat{\Phi}(\tau_2)$. Then we have

$$\hat{\Phi}(\tau_1) = \sum_{e\in E(\tau_1)} [\hat{\Phi}(\tau_1 \setminus e)] = \sum_{e\in E(\tau_2)} [\hat{\Phi}(\tau_2 \setminus e)] = \hat{\Phi}(\tau_2).$$

Use the fact that $[\cdot]$ is injective. We get

$$\sum_{e\in E(\tau_1)} \hat{\Phi}(\tau_1 \setminus e) = \sum_{e\in E(\tau_2)} \hat{\Phi}(\tau_2 \setminus e).$$

By assumption, $\hat{\Phi}$ is injective on γ with $|\gamma| < |\tau_1| = |\tau_2|$ so

$$\sum_{e\in E(\tau_1)} \tau_1 \setminus e = \sum_{e\in E(\tau_2)} \tau_2 \setminus e.$$

Now apply the previous lemma and we get $\tau_1 = \tau_2$ which proves the injectivity of $\hat{\Phi}$ for trees with any number of vertices. □

4.1 The dual of the homomorphism

Recall the discussion of pre-Lie algebras from Section 3.5. The Hopf algebra homomorphism $\Phi : H_{CEM} \to H_{CK}$ is the dual of the unique algebra homomorphism $\Psi : \mathcal{A}_C \to \mathcal{A}_S$. The algebra homomorphism Ψ is uniquely defined on $\mathcal{A}_{\mathcal{T}} \subset \mathcal{A}_C$ by sending $\bullet$ to $[\bullet]$ and uniquely defined on the rest of $\mathcal{A}_C$ by $\Psi(\tau_1\tau_2) = \Psi(\tau_1)\Psi(\tau_2)$. Let $]\cdot[: \mathcal{T} \to \mathcal{F}$ be the function that removes the root of a tree. Take $t : (\mathcal{T} \sqcup \mathcal{T}) \cup \mathcal{T} \to \mathcal{T} \otimes \mathcal{T}$ be defined the following way

$$\begin{aligned} t(\tau_1 \sqcup \tau_2) &= \tau_1 \otimes \tau_2 + \tau_2 \otimes \tau_1, \text{ if } \tau_1 \neq \tau_2, \\ t(\tau \sqcup \tau) &= \tau \otimes \tau, \\ t(\tau) &= \bullet \otimes \tau + \tau \otimes \bullet. \end{aligned}$$

Let $\tau \in \mathcal{F}$ be a forest with more than two trees then $t(\tau) = 0$.

Proposition 5. *Let $\Psi = \Phi^\circ : A_C \to A_S$. Then*

$$\Psi = A_\sigma \circ \curvearrowright \circ (A_\sigma^{-1} \otimes A_\sigma^{-1}) \circ t \circ \Psi \circ]\cdot[.$$

Proof. The formula for the $\Phi : H_{CEM} \to H_{CK}$ can be written as $\Phi = [\cdot] \circ \Phi \circ \big(\sum_{e \in E(\tau)} \tau \backslash e \big)$.

Recall that $\tau \setminus e = \tau_1^e \sqcup \tau_2^e$ for some τ_1^e and τ_2^e. Then the dual of $\sum_{e \in E(\tau)} \tau \setminus e$ is $\Delta_{CK}^\circ \circ t$. It is clear that the dual of $[\cdot]$ is the map that removes the root denoted as $]\cdot[$. $\square$

The dual of $\hat{\Phi}$ is denoted by $\hat{\Psi}$. From the definiton of $\hat{\Phi}$ we have $\hat{\Psi} = r^\circ \circ \Phi^\circ = r^\circ \circ \Psi$. Let τ be a non-rooted tree then $r(\tau)$ is the set of all possible rooted trees that can be obtained by choosing a vertex or τ as a root. Then $\hat{\Psi}(\tau) = \frac{1}{|r(\tau)|}\Psi(\tau)$.

Let us try to understand what the formula for Ψ actually means. Let us see what it does. Denote by Ψ_σ the map defined by $\Psi = A_\sigma \circ \Psi_\sigma$. Take $\tau = [\tau_1 \sqcup \tau_2]$ and assume $\Psi(\tau_1) \neq \Psi(\tau_2)$. Then

$$\begin{aligned} \Psi(\tau) &= A_\sigma \circ \curvearrowright \circ (A_\sigma^{-1} \otimes A_\sigma^{-1}) \circ t \circ \Psi \circ]\tau[\\ \Psi(\tau) &= A_\sigma \circ \curvearrowright \circ (A_\sigma^{-1} \otimes A_\sigma^{-1}) \circ t(\Psi(\tau_1) \sqcup \Psi(\tau_2)) \\ \Psi(\tau) &= A_\sigma \circ \curvearrowright \circ (A_\sigma^{-1} \otimes A_\sigma^{-1})(\Psi(\tau_1) \otimes \Psi(\tau_2) + \Psi(\tau_2) \otimes \Psi(\tau_1)) \\ \Psi(\tau) &= A_\sigma \circ \curvearrowright (\Psi_\sigma(\tau_1) \otimes \Psi_\sigma(\tau_2) + \Psi_\sigma(\tau_2) \otimes \Psi_\sigma(\tau_1)) \\ \Psi(\tau) &= A_\sigma(\Psi_\sigma(\tau_1) \curvearrowright \Psi_\sigma(\tau_2) + \Psi_\sigma(\tau_2) \curvearrowright \Psi_\sigma(\tau_1)). \end{aligned}$$

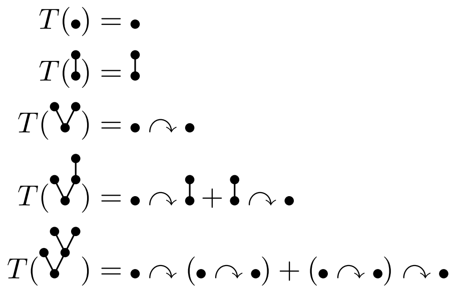
So $\Psi_\sigma([\tau_1 \sqcup \tau_2]) = \Psi_\sigma(\tau_1) \curvearrowright \Psi_\sigma(\tau_2) + \Psi_\sigma(\tau_2) \curvearrowright \Psi_\sigma(\tau_1)$. A similar computation could be done for two other cases:

- $\Psi(\tau_1) = \Psi(\tau_2) \implies \Psi_\sigma([\tau_1 \sqcup \tau_2]) = \Psi_\sigma(\tau_1) \curvearrowright \Psi_\sigma(\tau_2)$,
- $\tau = [\tau_1] \implies \Psi_\sigma([\tau_1]) = \bullet \curvearrowright \Psi_\sigma(\tau_1) + \Psi_\sigma(\tau_1) \curvearrowright \bullet$.

Now let us take a concrete example with $\tau = [\bullet \sqcup [\bullet]]$. Then

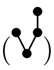
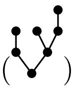
$$\begin{aligned} \Psi_\sigma([\bullet \sqcup [\bullet]]) &= \Psi_\sigma(\bullet) \curvearrowright \Psi_\sigma([\bullet]) + \Psi_\sigma([\bullet]) \curvearrowright \Psi_\sigma(\bullet) \\ &= [\bullet] \curvearrowright (\bullet \curvearrowright \Psi_\sigma(\bullet) + \Psi_\sigma(\bullet) \curvearrowright \bullet) + (\bullet \curvearrowright \Psi_\sigma(\bullet) + \Psi_\sigma(\bullet) \curvearrowright \bullet) \curvearrowright [\bullet] \\ &= [\bullet] \curvearrowright \Big(\bullet \curvearrowright [\bullet] + [\bullet] \curvearrowright \bullet\Big) + \Big(\bullet \curvearrowright [\bullet] + [\bullet] \curvearrowright \bullet\Big) \curvearrowright [\bullet] \\ &= [\bullet] \curvearrowright \Big(\bullet \curvearrowright [\bullet]\Big) + [\bullet] \curvearrowright \Big([\bullet] \curvearrowright \bullet\Big) + \Big(\bullet \curvearrowright [\bullet]\Big) \curvearrowright [\bullet] + \Big([\bullet] \curvearrowright \bullet\Big) \curvearrowright [\bullet]. \end{aligned}$$

We can notice a certain pattern. Let T be a linear map from the vector space generated by the trees with each vertex having at most two outgoing edges. Let the vertices with two outgoing edges correspond to the grafting of the successors in all possible orders. For example,



$$\dots$$

$$T([\tau_1, \tau_2]) = \curvearrowright \circ\, t(T(\tau_1) \sqcup T(\tau_2)).$$

Then we can see that $\psi_\sigma($$) = T($$)$. This fact is not random. It can be seen that due to the fact that $\Psi(\bullet) =$ and $\Psi(\emptyset) = \bullet$, the map Ψ_σ can be written as $\Psi_\sigma = T \circ \mathrm{Grow}$ where the map Grow attaches a leaf to each vertex with one or none outgoing edges. Therefore,

$$\hat{\Psi}(\tau) = \frac{1}{|r(\tau)|} A_\sigma \circ T \circ \mathrm{Grow}(\tau).$$

The goal now is to find $\ker(\hat{\Psi})$. Then $\hat{\Psi} : A_C/_{\ker(\hat{\Psi})} \to \hat{A_S}$ is injective. Notice that both A_σ and Grow are injective. We also know that

$$\ker(\curvearrowright) = \mathrm{span}_{\mathbb{R}}\{\tau \otimes (\gamma \curvearrowright \nu) - (\tau \curvearrowright \gamma) \otimes \nu - \gamma \otimes (\tau \curvearrowright \nu) + (\gamma \curvearrowright \tau) \otimes \nu \mid \tau, \gamma, \nu \in \mathcal{T}\},$$

as A_C is the free pre-Lie algebra. So

$$\ker(T) = (t \circ T \circ]\cdot[)^{-1} \left(\ker(\curvearrowright) \cap \mathrm{im}(t \circ T \circ]\cdot[)\right).$$

A task for the future would be to find a nice and full description of $\ker(T)$.

Proposition 6. *The Hopf algebra homomorphism $\hat{\Phi} : \hat{H}_{CEM} \to H_{CK}/_{ker(\hat{\Psi})}$ is an isomorphism between CEM Hopf algebra of non-rooted trees and the quotient of CK Hopf subalgebra of trees with each vertex having at most two outgoing edges.*

Proof. We have already proved that $\hat{\Phi}$ is injective. It remains to show that it is also surjective. Surjectivity can be expressed the following way, for all $\tau \in \mathcal{F}$ there exists $\gamma \in \mathcal{F}$ such that for all $\nu \in \mathcal{F}, \tau \neq \nu$ we have

$$(\hat{\Phi}(\gamma), \tau) = 1 \quad \text{and} \quad (\hat{\Phi}(\gamma), \nu) = 0.$$

Using the dual of $\hat{\Phi}$ we get the following condition on the dual. For all $\tau \in \mathcal{F}$ there exists $\gamma \in \mathcal{F}$ such that for all $\nu \in \mathcal{F}, \tau \neq \nu$ we have

$$(\gamma, \hat{\Psi}(\tau)) = 1 \quad \text{and} \quad (\gamma, \hat{\Psi}(\nu)) = 0.$$

This condition is exactly the injectivity of $\hat{\Psi}$. Thus, $\hat{\Phi}$ is surjective. □

5 Exotic B-series

Different generatlizations of trees were introduced in the stochastic context for the study of strong and weak errors. Burrage and Burrage [3] and Komori, Mitsui and Sugiura [19] introduced stochastic trees and B-series for studying the order conditions for strong convergence of SDE, and [4, 13, 14, 22, 23, 24, 25] for study of high order weak and strong methods on a finite time interval.

First studied in [20], exotic B-series are a generalization of B-series that presents the tools to work with stochastic differential equations and study the order for the invariant measure of numerical methods applied to such equations. The generalization is applicable only if several assumptions are made.

5.1 Assumptions and definitions

We consider a class of SDE of the form

$$dX(t) = f(X(t))dt + \sqrt{2}dW(t)$$

where $X(t) \in \mathbb{R}^d$ is an adapted stochactic process with $X(0) = X_0$, the vector field $f : \mathbb{R}^d \to \mathbb{R}^d$ smooth and globally Lipschitz, $\sqrt{2}$ is a constant that can be changed by rescaling of the problem, and $W(t)$ is a standard $d-$dimensional Wiener process fulfilling the usual assumptions.

Definition 2. A problem is *ergodic* if there exists a unique invariant measure μ satisfying for all deterministic initial conditions X_0 and all smooth test functions ϕ,

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T \phi(X(s))ds = \int_{\mathbb{R}^d} \phi(x)d\mu(x), \quad \text{almost surely.}$$

Assumption 1. *The vector field $f : \mathbb{R}^d \to \mathbb{R}^d$ is globally Lipschitz and C^∞, and there exist C_1, C_2 such that for all $x \in \mathbb{R}^d$,*

$$x^T f(x) \leq -C_1 x^T x + C_2.$$

Assumption 2. *There exists a C^∞ map $V : \mathbb{R}^d \to \mathbb{R}$ such that $f(x) = -\nabla V(x)$ is globally Lipschitz and there exist $C_1 > 0$ and C_2 such that for all $x \in \mathbb{R}^d$, $V(x) \geq C_1 x^T x - C_2$.*

It is known that such a problem is ergodic and the density of the unique invariant measure is $\rho_\infty = Z\exp(-V)$ where Z is such that $\int_{\mathbb{R}^d} \rho_\infty(x)dx = 1$. A useful property is that

$$\nabla\rho_\infty = \rho_\infty f \quad \text{or} \quad \nabla(\log\rho_\infty) = f.$$

Assumption 3. *The integrator $X_{n+1} = \Psi(X_n, h, \xi_n)$ has bounded moments of any order along time, i.e., for all integer $k \geq 0$,*

$$\sup_{n\geq 0} \mathbb{E}[|X_n|^{2k}] < \infty \quad \forall k \geq 0$$

Assumption 4. *The integrator $X_{n+1} = \Psi(X_n, h, \xi_n)$ has a weak Taylor expansion of the form*

$$\mathbb{E}[\phi(X_1)|X_0 = x] = \phi(x) + h\mathcal{A}_1\phi(x) + h^2\mathcal{A}_2\phi(x) + \cdots$$

for all $\phi \in C_P^\infty(\mathbb{R}^d, \mathbb{R})$, where $\mathcal{A}_i$, $i = 1, 2, \ldots$, are linear differential operators. For more details see [27].

Definition 3. A numerical method $X_{n+1} = \Psi(X_n, h, \xi_n)$ is **ergodic** if there exists a unique invariant probability law μ^h with finite moments of any order satisfying for all deterministic initial conditions $X_0 = x$ and all smooth test functions ϕ,

$$\lim_{N\to\infty} \frac{1}{N+1} \sum_{n=0}^{N} \phi(X_n) = \int_{\mathbb{R}^d} \phi(x) d\mu^h(x), \quad \text{almost surely.}$$

See [21] for more details.

Definition 4. A numerical method $X_{n+1} = \Psi(X_n, h, \xi_n)$ has order p with respect to the invariant measure of the SDE if

$$\left| \int_{\mathbb{R}^d} \phi(x) d\mu^h(x) - \int_{\mathbb{R}^d} \phi(x) d\mu(x) \right| \le Ch^p,$$

where C is independent of h assumed small enough.

Theorem 1. *[1] Take an integrator $X_{n+1} = \Psi(X_n, h, \xi_n)$. Assume the Assumptions 1, 2, 3, 4 to be true. If*

$$A_j^* \rho_\infty = 0, \quad j = 2, \dots, p-1$$

then the integrator has order p for the invariant measure.

5.2 Exotic trees

We consider $\xi \sim \mathcal{N}(0, I_d)$ a normally distributed random variable in $\mathbb{R}^d$.

Definition 5. *Grafted exotic trees* are trees that can have grafted nodes $\times$ as leaves and let the function $F_{f,\phi,\xi}$ which sends an exotic grafted tree to its corresponding elementary differential operator be defined as

$$F_{f,\phi,\xi}(\tau) := \sum_{i_{v_1},\dots,i_{v_m}=1}^{d} \left(\prod_{v \in V^0(\tau)} \partial_{I_{\pi(\tau,v)}} \left(F_{f,\phi,\xi}(v)\right)_{i_v} \right) \partial_{I_{\pi(\tau,r)}} \phi$$

with

$$F_{f,\phi,\xi}(v) = \begin{cases} f, & \text{for } v \text{ non-root vertex } \bullet, \\ \xi, & \text{for } v \text{ grafted leaf } \times, \end{cases}$$

where $V^0(\tau)$ is the set of non-root vertices, $I_{\pi(\tau,v)} = (i_{q_1}, \dots, i_{q_s})$ with the q_k being the predecessors of v, r is the root of τ, and

$$\partial_{I_{\pi(\tau,v)}} f = \frac{\partial^s f}{\partial x_{i_{q_1}} \cdots \partial x_{i_{q_s}}}.$$

Notice that this definition is a natural extension of the original definition seen in the beginning of the thesis. However, due to the Theorem 1, we will be interested in the expectation of $F_{f,\phi,\xi}(\tau)$. From the definition of $F_{f,\phi,\xi}$ it follows that the expectation depends only on the grafted notes, i.e. on term of the form $\mathbb{E}(\xi_{i_1} \cdots \xi_{i_m})$.

We know that if n is odd then the expectation is zero. Thus, we consider $\mathbb{E}(\xi_{i_1} \cdots \xi_{i_{2n}})$. We know that $\mathbb{E}(\xi_i \xi_j) = \mathbb{E}(\xi_i)\mathbb{E}(\xi_j)$ if $i \neq j$. Thus, the indeces must have even multiplicities.

The Isserlis theorem [18] will be helpful here. It states that if χ is a $2n$-dimensional normally distributed random vector with mean zero then

$$\mathbb{E}\left[\prod_{i=1}^{2n}\chi_i\right] = \sum_{p\in\mathcal{P}_2(2n)} \prod_{\substack{i<j\\ p(i)=p(j)}} \mathbb{E}\left[\chi_i\chi_j\right].$$

For example, $\mathbb{E}[\chi_1\chi_2\chi_3\chi_4] = \mathbb{E}[\chi_1\chi_2]\mathbb{E}[\chi_3\chi_4] + \mathbb{E}[\chi_1\chi_3]\mathbb{E}[\chi_2\chi_4] + \mathbb{E}[\chi_1\chi_4]\mathbb{E}[\chi_2\chi_3]$.

In our case, $\mathbb{E}\left[\prod_{i=1}^{2n}\chi_i\right] = |\mathcal{P}_2(2n)|$ because $\mathbb{E}[\xi_i\xi_i] = 1$. In the end, the Isserlis theorem is enough to show that expectations of random variable with even multiplicities greater than two can be reduced to the case where all multiplicities are equal to two.

Thus, if we come back to considering $\mathbb{E}\left[F_{f,\phi,\xi}(\tau)\right]$ we get

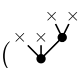
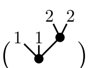
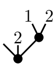
$$\mathbb{E}\left[F_{f,\phi,\xi}(\tau)\right] = \sum_{p\in\mathcal{P}_2(2n)} \prod_{\substack{i<j\\ p(i)=p(j)}} F_{f,\phi}(\tau_p)$$

where τ_p is the grafted exotic tree τ with grafted vertices are paired according to the pairing p. Such a tree is called *exotic tree*. The function $F_{f,\phi}$ is deterministic and is defined the same as $F_{f,\phi,\xi}$ with only exception that

$$F_{f,\phi}(v) = \begin{cases} f, & \text{for } v \text{ non-root vertex } \bullet, \\ 1, & \text{for } v \text{ numbered leaf }, \end{cases}$$

and $\pi(\tau_p, v)$ is a set of all predecessors of v with every numbered predecessor u being replaced by $p(u)$. That is, the paired numbered leaves have the same index corresponding to both of them.

The pairing of grafted vertices can be shown by connecting the grafts by *lianas* (hence the name exotic). In the next nection we are going to look at some algorithms on exotic trees and a package that implements those algorithms. For computational reasons the pairing will be shown by numbering the grafted vertices in an appropriate way. For example,

$\mathbb{E}\left[F_{f,\phi,\xi}(\right.$  $\left.)\right] = F_{f,\phi}($  $) + 2F_{f,\phi}($  $)$.

Exotic trees can be very useful if the method has $\mathcal{A}_i = F_{f,\phi}(\gamma)$ where γ is a linear combination of exotic trees.

Definition 6. *Exotic B-series* is a formal series over grafted exotic trees $\mathcal{ET}_g$ with $a : \mathcal{ET}_g \to \mathbb{R}$ a map, $f : \mathbb{R}^d \to \mathbb{R}^d$ and $\phi : \mathbb{R}^d \to \mathbb{R}$ two smooth functions, and $\xi \sim \mathcal{N}(0, I_d)$. It has the form

$$B(a)(\phi) := \sum_{\tau\in\mathcal{ET}_g} h^{|\tau|} a(\tau) F_{f,\phi,\xi}(\tau)$$

where $|\tau| = |\{\bullet \text{ vertices of } \tau\}| + \frac{1}{2}|\{\text{grafted leaves of } \tau\}| - 1$. Notice that a grafted leaf counts only as $\frac{1}{2}$. This is due to the fact that the random variable ξ has coefficient $\sqrt{h}$ while f has coefficient h.

Remark 1. An important fact to notice is that $\mathbb{E}\left[B(a)(\phi)\right]$ is a formal series over exotic trees $\mathcal{ET}$. This means that if a numerical method is an exotic B-series then the corresponding $\mathcal{A}_i$ is $F_{f,\phi}(\gamma)$ for some linear combination of exotic trees γ. Notice that

$$\mathcal{A}_i^* \rho_\infty = 0 \quad \iff \quad \int_{\mathbb{R}^d} (\mathcal{A}_i\phi)(x)\rho_\infty(x)dx = 0, \quad \forall \phi \in C_P^\infty$$

Thus, to show the order of a numerical method in invariant measure, it is enough to show that $\int F_{f,\phi}(\gamma)(\phi)\rho_\infty dx = 0$ for certain γ which can difficult to compute because of higher order differentials. To simplify the calculations, two simplifying techniques can be used: the integration by parts and the inversion of edge-liana.

6 Order conditions for invariant measure of ergodic SDE

In this section, we generate order conditions using the Theorem 1 from Section 5.1 based on the paper [20]. We describe the order conditions for exotic B-series following the remark 1 from Section 5.2.

Definition 7. Let $\gamma_1, \gamma_2 \in \text{span}_{\mathbb{R}}(\mathcal{ET})$. Let $\gamma_1 \sim \gamma_2$ if and only if

$$\int_{\mathbb{R}^d} F_{f,\phi}(\gamma_1)\rho_\infty dx = \int_{\mathbb{R}^d} F_{f,\phi}(\gamma_2)\rho_\infty dx.$$

It follows that to prove that a numerical method has order p in invariant measure we first have to find $\gamma_j = \mathcal{A}_j$ for $j < p$ and then show that $\gamma_j \sim 0$.

6.1 Simplification of exotic trees

6.1.1 Integration by parts (IBP)

Due to that fact that the equivalence relation is defined using integrals, we can use integration by parts (IBP) to decrease the order of the differentials that we work with. We know that

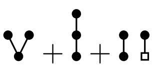
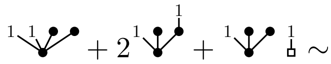
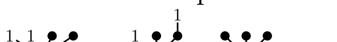
$$\int_{\mathbb{R}^d} \partial_i (F_{f,\phi}(\gamma \setminus e_i)\rho_\infty)dx = 0$$

where $\gamma \setminus e_i$ is the tree where we remove the edge coming from the vertex v_i. This is true due to the fact that f and ϕ have polynomial growth. Now open the parenthesis and we get

$$\sum_{v \in V(\gamma)} \int_{\mathbb{R}^d} F_{f,\phi}(v_i \to_v (\gamma \setminus e_i))\rho_\infty dx + \int_{\mathbb{R}^d} F_{f,\phi}(\gamma \setminus e_i)\partial_i \rho_\infty dx = 0.$$

Recall that $\nabla\rho_\infty = f\rho_\infty$. Take $g = \log(\rho_\infty)$ then $\nabla\rho_\infty = (\nabla g)\rho_\infty$. Let us introduce a new kind of vertex $\square$ called *aromatic root* that will correspond to g, i.e. $F_{f,\phi}(\square) = g$. Then

$$\sum_{v \in V(\gamma)} v_i \to_v (\gamma \setminus e_i) + (\gamma \setminus v_i) \sqcup$$  $$\sim \; 0.$$

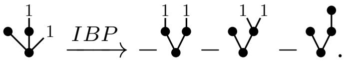
For example,  $\sim$ 0. Now apply this simplification to a numbered leaf. We notice that $\partial_i g = f_i$ and, therefore,  $\sim$ . This means that if we apply the integration by parts simplification only to numbered leaves then we can always get rid of the aromatic root which is not a part of our definition of exotic trees. For example,   Notice that the edge does not get attached to the numbered leaf because $F_{f,\phi}(1) = 1$ and its differential is 0. By choosing a numbered leaf v_i connected to the root (let it be numbered by 1) and moving the term of the sum where v_i is connected to the root to the other side we get

$$\sum_{\substack{v \in V(\gamma) \\ v \neq \text{root}}} v_i \to_v (\gamma \setminus e_i) + (\gamma \setminus v_i) \sqcup \overset{1}{\square} \sim -(v_i \to_{\text{root}} (\gamma \setminus e_i)) = -\gamma$$

and, therefore, we have

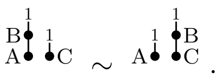
$$IBP(\gamma) = -\sum_{\substack{v \in V(\gamma) \\ v \neq \text{root}}} v_i \to_v (\gamma \setminus e_i) - (\gamma \setminus v_i) \sqcup \overset{1}{\square}.$$

In this thesis, the IBP is used to decrease the number of numbered leaves connected to the root. For example,

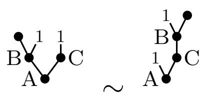


6.1.2 Inversion of edge-liana (IEL)

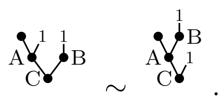
Another useful simplification that will be used is the inversion of edge-liana. Due to the fact that f is a gradient we know that f' is a symmetric matrix, i.e. $\partial_i f_j = \partial_j f_i$. This means that we have the following property



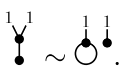
This can be used to move the liana through the edges of the tree. For example,



and



We allow only such inversions that give an exotic tree as a result. For example, the following inversion, even though it is correct, is not permitted.

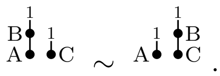


Definition 8. A *position* of a liana l in an exotic tree τ is the tuple (v_1, v_2) where $v_1, v_2 \in V(\tau)$ are the vertices connected by liana l. The position of l is denoted as $p(l)$.

Definition 9. An *orbit* of a liana l is the set of all the positions that the liana l can take when moved by the edge-liana inversion.

Proposition 7. *An orbit of a liana l in an exotic tree τ is the set O_l of all edges in τ that are part of a path from either end of liana to the root.*

Proof. Let us analyse what the inversion of edge-liana does in detail.

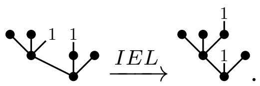


Let us assume that the left term is the original tree. In order to apply the inversion of edge-liana we need to have an outgoing edge connected to an end of the liana (B, in our case). Then we replace this outgoing edge by a liana and transform the liana (B, C) into an outgoing edge from B to C. This way liana "moved" through the vertex B and left an outgoing edge behind.

We see that the inversion of edge-liana is invertible and that the liana can be inverted only with an outgoing edge connected to its end. We know that if we take a tree and choose one of its vertices then we will always reach the root if we follow the outgoing edges from the chosen vertex.

This way if we repeatedly apply the inversion without ever inverting it we will trace a path from an end of the liana to the root. Thus, the orbit of a liana can be describe as a set of edges that are part of such a path. □

This proposition is used in Section 7 to define an algorithm that checks if two trees are similar. In this thesis, the IEL is used to move the liana to the root when possible. For example,



6.2 Order conditions

Let us use an algorithm involving the simplifications described above to find order conditions for the invariant measure of exotic B-series using the fact that an exotic B-series has weak Taylor expansion with $\mathcal{A}_i = F_{f,\cdot}(\gamma_i)$ where γ_i is the corresponding linear combination of exotic trees.

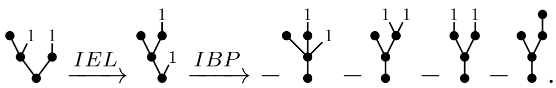
An exotic B-series is fully characterized by the map $a : \mathcal{ET} \to \mathbb{R}$ that gives the coefficients of the formal series over $\mathcal{ET}$. We have to find conditions on the values of a that would guarantee that the corresponding exotic B-series is of order p for the invariant measure. Such conditions should imply that $\gamma_i \sim 0$ for all $i = 1, \ldots, p-1$.

Let $\mathcal{A}_i$ have the following form

$$\mathcal{A}_i = \sum_{\substack{\tau \in \mathcal{ET} \\ |\tau| = i}} a(\tau) F_{f,\cdot}(\tau) = F_{f,\cdot}\Big(\sum_{\substack{\tau \in \mathcal{ET} \\ |\tau| = i}} a(\tau)\tau\Big) = F_{f,\cdot}(\gamma_i).$$

Proposition 8. *The Algorithm 1 ends.*

Proof. The algorithm acts on every tree which means that the algorithm is linear. Thus, it is enough to consider the operations of the algorithm applied to a single tree. This algorithm is guaranteed to end because every application of the integration by parts decreases the number of edges connected to the root. For example,



Algorithm 1: Apply the simplifications to get order conditions

Data: γ_i – a linear combination of exotic trees of a certain size.

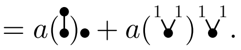
Result: A linear combination of exotic trees ν such that $\nu \sim \gamma_i$

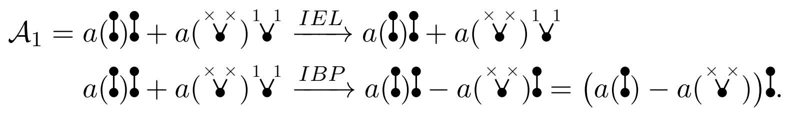
1. Apply the inversion of edge-liana to every exotic tree to move all lianas such that one end of liana is connected to the root if possible.

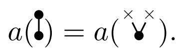
2. Apply the integration by parts to disconnect all lianas from the root. This will create new trees where the application of inversion of edge-liana is possible. **Go to step 1**.

This means that the IBP can be applied only a finite number of times. □

We can write the order conditions for invariant measure of exotic B-series. Table 6.2 lists A_2 and the result of execution of the Algorithm 1 applied to A_2. This table was generated using the symbolic package discussed in Section 7. One can also find order conditions for order 3 in Appendix A.

Let us apply the Algorithm 1 to 



Thus, an exotic B-series has order 1 for invariant measure if 

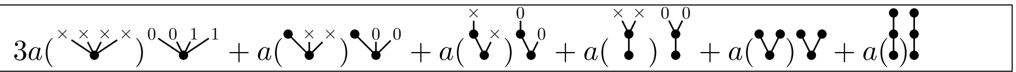
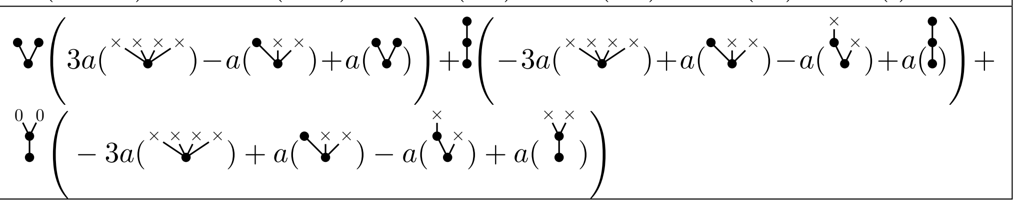
A_2	
Algo. 1 applied to A_2	

Table 1: Order conditions from $\mathcal{A}_2$

6.3 Order conditions for Runge-Kutta methods

It turns out that similarly with the classical theory of B-series, the Runge-Kutta methods of the form (1) are exotic B-series. The map $a : \mathcal{ET} \to \mathbb{R}$ corresponding to a Runge-Kutta method can be obtained the following way, where we denote $c_i := \sum_{j=1}^{s} a_{ij}$ to simplify notations.

Proposition 9. *Let a_{ij}, b_i, d_i be the coefficients defining a Runge-Kutta method. Then the method is an exotic B-series and the map $a : \mathcal{ET}_g \to \mathbb{R}$ is defined the following way. Take $\tau \in \mathcal{ET}_g$ and let $\tau = [\tau_1, \ldots, \tau_n]$. Then*

$$a(\tau) := \frac{1}{\sigma_g(\tau)} \sum_{\substack{i_v,\ v \in V(\tau) \\ \text{not root} \\ \text{not } \times}} \prod_{e \in E(\tau)} var(e)$$

where

$$var(e) = \begin{cases} b_{i_{o(e)}}, & \textit{if } t(e) \textit{ is the root, } o(e) \textit{ is } \bullet, \\ 1, & \textit{else if } t(e) \textit{ is the root, } o(e) \textit{ is } \times, \\ c_{i_{t(e)}}, & \textit{else if } o(e) \textit{ is a leaf } \bullet, \\ d_{i_{t(e)}}, & \textit{else if } o(e) \textit{ is } \times, \\ a_{i_{t(e)} i_{o(e)}}, & \textit{otherwise.} \end{cases}$$

and for $\tau \in \mathcal{ET}_g$ *with* $2l$ *grafted leaves we have*

$$\sigma_g(\tau) = \frac{\sigma(\tau)}{2^l}.$$

Proof is a straightforward extension of the theory presented in [15] in Chapter III.1.1. For $\tau_p \in \mathcal{ET}$ where p is a pairing of grafted leaves and $\tau \in \mathcal{ET}_g$ we have

$$a(\tau_p) = p(\tau, p)\, a(\tau)$$

where $p(\tau, p)$ is the number of pairings of grafted leaves that give the same tree τ_p.

Using this characterization of the map a for Runge-Kutta methods we can write the order conditions for invariant measure of Runge-Kutta methods. Table 6.3 lists γ_2, A_2, and the result of execution of the Algorithm 1 applied to A_2. This table was generated using a script that was programmed to perform the required operations on trees automatically. The symbolic package used to write the script will be discussed in more detail in Section 7. One can also find order conditions for order 3 and 4 in Appendix C.

The order conditions for order 3 generated by the script agree with those obtained in [20]. The order conditions for order 4 generated by the script were not computed before due to a large number of computations. This new result is not mathematically rigorous. The symbolic package implements several algorithms described in Section 7 that do not have a mathematically rigorous proof. Moreover, the implementation of the algorithms is not mathematically rigorous.

A way to see if the output of the script contains errors would be to apply the script to a simpler computation. We see that the script gives a correct output for the computation of order conditions for order 3. Another way to check the script for errors is to apply it to $\mathcal{A}_4$ of the Gaussian case in which $V = \sum_{i=1}^d C_i x_i^2$ and f^i is a polynomial of degree 1 for any $i = 1, \ldots, d$. In this case, any differential of degree 2 of f^i is 0 which means that all the trees where a non-root vertex has more than one incoming edge should be set to 0. This simplifies the calculations and makes it possible to check the output of the script. The case is called Gaussian because the invariant measure ρ_∞ is a Gaussian. The output of the script applied to this case can be found in the Appendix B.

γ_2	$\frac{b_0 b_1}{2} + b_0 c_0 + b_0 d_0^2 + 2 b_0 d_0 + b_0 + \frac{1}{6}$
A_2	$\frac{b_0 b_1}{2} + b_0 c_0 + b_0 d_0^2 + 2 b_0 d_0 + b_0 + \frac{1}{2}$
Algo. 1 applied to A_2	$\left(\frac{b_0 b_1}{2} - b_0 + \frac{1}{2}\right) + \left(b_0 c_0 - 2 b_0 d_0 + b_0 - \frac{1}{2}\right) + \left(b_0 d_0^2 - 2 b_0 d_0 + b_0 - \frac{1}{2}\right)$

Table 2: Order conditions from $\mathcal{A}_2$ for Runge-Kutta methods

6.4 Analysis of the order conditions

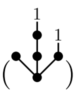
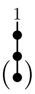
During the computation of the order conditions for the Runge-Kutta methods some interesting patterns were noticed. We discuss these patterns in this section. Take $\mathrm{Coef}(\tau)$ to be the coefficient of τ in the result of the Algorithm 1 applied to $\mathcal{A}_{|\tau|}$ of a Runge-Kutta method. Let us denote the coefficient in $a(\tau_p)$ by $\hat{\sigma}$ for simpicity, i.e.

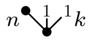
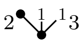
$$\hat{\sigma}(\tau_p) = \frac{p(\tau, p)}{\sigma_g(\tau)}.$$

Notice an interesting property. Take $\tau_1, \tau_2 \in \mathcal{ET}$. Then

$$a([\tau_1, \tau_2]) = \frac{\hat{\sigma}([\tau_1, \tau_2])}{\hat{\sigma}([\tau_1])\hat{\sigma}([\tau_2])} a([\tau_1])a([\tau_2])$$

due to the fact that $a([\tau_1, \tau_2])$ does not take into account the number of roots. For example,

$a($$) = a($$)a($$) = \sum b_{i_1} b_{i_2}$, $a($$) = a($$)a($$)a($$) = \sum b_{i_1} b_{i_2} a_{i_2,i_3} d_{i_3} b_{i_4} d_{i_4}$.

We notice that if τ_1 does not contain any trees isomorphic to any trees in τ_2 then $a([\tau_1, \tau_2]) = a([\tau_1])a([\tau_2])$. We consider a class of exotic trees that is denoted by  which means that there are n leaves and k liana loops connected to the root of the exotic tree. For example,  $=$ .

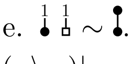
Let us characterize $\mathrm{Coef}(\tau)$ and find a formula for $\mathrm{Coef}($$)$. Define $[\tau_1] \circ [\tau_2] = [\tau_1, \tau_2]$, then we can write any exotic tree τ with $|\tau| > 1$ as $\tau = [\tau_1, \dots, \tau_n] = [\tau_1] \circ \cdots \circ [\tau_n]$ for some exotic trees $[\tau_1], \dots, [\tau_2]$ that cannot be further decomposed this way. Let $|\tau|_p = |[\tau_1, \dots, \tau_n]|_p = |[\tau_1] \circ \cdots \circ [\tau_n]|_p = (|[\tau_1]|, \dots, |[\tau_n]|)$. Notice that the order of elements in $|\tau|_p$ does not matter as the order of branches in the tree does not matter. Recall that a partition of a number N is a tuple $(N_1, \dots, N_k)$ of integers such that $N = N_1 + \cdots + N_k$.

Lemma 4. *Let τ be an exotic tree with $|\tau| > 1$ and $\tau = [\tau_1] \circ \cdots \circ [\tau_n]$. Then Coef$(\tau)$ does not contain exotic tree $\gamma = [\gamma_1] \circ \cdots \circ [\gamma_m]$ (i.e. $(\mathrm{Coef}(\tau), \gamma) = 0$) if the terms of $|\gamma|_p$ cannot be split into partitions of terms of $|\tau|_p$.*

Proof. We look at the relation between $|\gamma|_p$ and $|\nu|_p$ where ν is the result of the Algorithm 1 applied to γ. We know that $|\gamma|_p \in \mathbb{N}^n$. Then let $|\gamma|_p^{i \to j} \in \mathbb{N}^{n-1}$ such that the i-th element is removed and added to the j-th element. For example, $(5, 7, 9)^{2 \to 3} = (5, 16)$.

Recall that the integration by parts (IBP) is defined the following way

$$IBP(\gamma) = -\sum_{\substack{v \in V(\gamma) \\ v \neq \mathrm{root}}} v_i \to_v (\gamma \setminus e_i) - (\gamma \setminus v_i) \sqcup$$ .

We know that if a liana connects a vertex v to an aromatic root, then the liana and the aromatic root can be replaced by a leaf connected to v by a simple edge, i.e. . Thus, $|(\gamma \setminus v_i) \sqcup$ $|_p = |\gamma|_p$. Moreover, $|v_i \to_v (\gamma \setminus e_i)|_p = |\gamma|_p$ if $v, v_i \in [\gamma_k]$ and $|v_i \to_v (\gamma \setminus e_i)|_p = |\gamma|_p^{k \to j}$ if $v \in [\gamma_j], v_i \in [\gamma_k]$. Therefore,

$$|IBP(\gamma)|_p = -(|\gamma_k| + 1) \cdot |\gamma|_p - \sum_{j=1}^{m} |\gamma_j| \cdot |\gamma|_p^{k \to j}$$

where $v_i \in [\gamma_k]$. The term $|\gamma_k| + 1$ counts all the vertices of $[\gamma_k]$ to which the liana can be connected (and, therefore, $|\cdot|_p$ does not change) toghether with the case where the liana is connected to the aromatic root. We also notice that the inversion of edge-liana does not change $|\cdot|_p$ because it only moves the liana through the tree, so $|IEL(\gamma)|_p = |\gamma|_p$. Therefore the effect of the Algorithm 1 on the $|\cdot|_p$ is the same as the effect of IBP applied multiple times.

If $\hat{\gamma}$ is in the output of Algorithm 1 applied to γ, then the terms of $|\hat{\gamma}|_p$ are sums of terms in $|\gamma|_p$ where a term from $|\gamma|_p$ can appear only in one sum. Thus, if the terms of $|\tau|_p$ are not sums of terms from $|\gamma|_p$ then τ is not in the output of Algorithm 1 applied to γ and, therefore, γ is not in Coef(τ). ☐

Proposition 10. *Given a bouquet with $n \in \mathbb{N}$ vertices we have*

$$\mathit{Coef}({}^{n}\overset{\bullet}{\bullet}) = \frac{1}{n}\mathit{Coef}(\overset{\bullet}{\bullet})\,\mathit{Coef}({}^{n-1}\overset{\bullet}{\bullet}).$$

Proof. It follows from the lemma that Coef(${}^{n}\overset{\bullet}{\bullet}$) contains only the trees of the form ${}^{n-k}\!\overset{\bullet}{\searrow}\overset{1}{\downarrow}\swarrow^{1}{}^{k}$ for $k = 0, \dots, n$ because $|{}^{n}\overset{\bullet}{\bullet}|_p = (1, \dots, 1)$. The only way in which we can get the tree ${}^{n}\overset{\bullet}{\bullet}$ from ${}^{n-k}\!\overset{\bullet}{\searrow}\overset{1}{\downarrow}\swarrow^{1}{}^{k}$ is by disconnecting each liana from the root and connecting it to the aromatic root. This replaces the liana by a leaf. Therefore, we have

$$\mathrm{Coef}({}^{n}\overset{\bullet}{\bullet}) = \sum_{k=0}^{n}(-1)^k a({}^{n-k}\!\overset{\bullet}{\searrow}\overset{1}{\downarrow}\swarrow^{1}{}^{k}).$$

Using Proposition 9 we have

$$\begin{aligned} a({}^{n-k}\!\overset{\bullet}{\searrow}\overset{1}{\downarrow}\swarrow^{1}{}^{k}) &= \hat{\sigma}({}^{n-k}\!\overset{\bullet}{\searrow}\overset{1}{\downarrow}\swarrow^{1}{}^{k})\sum b_{i_1}\cdots b_{i_{n-k}} = \\ &= \frac{(2k)!}{2^k k!}\cdot\frac{2^k}{(n-k)!(2k)!}\sum b_{i_1}\cdots b_{i_{n-k}} = \frac{1}{k!(n-k)!}\sum b_{i_1}\cdots b_{i_{n-k}}. \end{aligned}$$

This means that the Coef(${}^{n}\overset{\bullet}{\bullet}$) has the following formula

$$\begin{aligned} \mathrm{Coef}({}^{n}\overset{\bullet}{\bullet}) &= \sum_{k=0}^{n}(-1)^k\frac{1}{k!(n-k)!}\sum b_{i_1}\cdots b_{i_{n-k}} = \\ &= \frac{1}{n!}\sum_{k=0}^{n}(-1)^k\binom{n}{k}\sum b_{i_1}\cdots b_{i_{n-k}} = \frac{1}{n!}\prod_{k=0}^{n}(\sum b_{i_k} - 1). \end{aligned}$$

Therefore, we have

$$\begin{aligned} \frac{1}{n}\mathrm{Coef}(\overset{\bullet}{\bullet})\mathrm{Coef}({}^{n-1}\overset{\bullet}{\bullet}) &= \frac{1}{n}(\sum b_i - 1)\cdot\frac{1}{(n-1)!}\prod_{k=0}^{n-1}(\sum b_{i_k} - 1) = \\ &= \frac{1}{n!}\prod_{k=0}^{n}(\sum b_{i_k} - 1) = \mathrm{Coef}({}^{n}\overset{\bullet}{\bullet}). \end{aligned}$$

This finishes the proof of the proposition. ☐

Proposition 11. *Let τ be a tree with $n-1$ vertices attached to the root where one vertex has a leaf or a liana loop attached to it. It has the form ${}^{n-2}$[tree] or ${}^{n-2}$[tree with liana loop 1 1] with $n-2$ leaves attached to the root. Then*

$$Coef({}^{n-2}[\text{tree}]) = Coef([\text{tree}])\,Coef({}^{n-2}[\text{tree}]) \quad and \quad Coef({}^{n-2}[\text{tree}]^{1\,1}) = Coef([\text{tree}]^{1\,1})\,Coef({}^{n-2}[\text{tree}])$$

Proof. The proof for $\tau = {}^{n-2}[\text{tree}]^{1\,1}$ is completely analogous to the proof for $\tau = {}^{n-2}[\text{tree}]$ with the only difference being that the leaf is a liana connected to a vertex and an aromatic root while a looped liana has both ends connected to the same vertex. We know that

$$\mathrm{Coef}([\text{tree}]) = a([\text{tree}]) - a([\text{tree}]^{1\,1}) + a([\text{tree}]^{1\,1}) - a([\text{tree}]^{1\,1\,2\,2})$$

and

$$\mathrm{Coef}({}^{n-2}[\text{tree}]) = \sum_{k=0}^{n-2} (-1)^k a({}^{n-k-2}[\text{tree}]^{1\,1\,k})$$

To write the formula of $\mathrm{Coef}(\tau)$ it is enough to find all trees that can be transformed into τ by Algorithm 1 and count the number of instances of τ obtained by transforming a particular tree. Using Lemma 4 we find the three classes of trees that can be transformed into τ. Notice that for these classes of trees the IEL is never applied and Algorithm 1 is reduced to IBP. There are three classes of such trees:

- ${}^{n-k-2}[\text{tree}]^{1\,1\,k}$ – apply IBP to every liana. The only one way in which we can get τ is by connecting each liana to an aromatic root. This is similar to the way we get a bouquet tree. Thus, every tree of this class gives one instance of τ.
- ${}^{n-k-2}[\text{tree}]^{2\,2\,1\,1\,k}$ – apply IBP to every liana. We get τ be connecting each liana to an aromatic root. Thus, every tree of this class gives one instance of τ.
- ${}^{n-k}[\text{tree}]^{1\,1\,k}$ – apply IBP to every liana. This class is trickier as each tree in this class gives several instances of τ. Let us apply the IBP to a liana and see what happens:

$$ {}^{n-k}[\text{tree}]^{1\,1\,k} \xrightarrow{IBP} (n-k)\,{}^{n-k-1}[\text{tree}]^{2\,2\,1\,1\,k-1} + {}^{n-k+1}[\text{tree}]^{1\,1\,k-1}. $$

  We see that the first term is a tree from the previous class which gives one instance of τ. The second term is a tree from the same class with one more leaf and one less liana. This gives a recursive expression of the number of instances of τ that can be obtained from ${}^{n-k}[\text{tree}]^{1\,1\,k}$. It can be written as

$$\sum_{i=0}^{k-1} n-k+i.$$

  Notice that the sum is only up to $k-1$. This is due to the fact that if all k lianas are attached to aromatic roots then we get a bouquet with n vertices which cannot be transformed into τ. Rewrite this sum in a more useful form:

$$\sum_{i=0}^{k-1} n-k+i = k(n-k) + \sum_{i=0}^{k-1} i = k(n-k) + \frac{k(k-1)}{2}.$$

As a result of the these computations we can write a formula for $\mathrm{Coef}({}^{n-2}\bullet\!\searrow\!\!\bullet\!\!\!\!\overset{\bullet}{|}\!\!\swarrow)$:

$$\mathrm{Coef}({}^{n-2}\bullet\!\searrow\!\overset{\bullet}{\overset{|}{\bullet}}\!\swarrow) = \sum_{k=0}^{n-2}(-1)^k a({}^{n-k-2}\bullet\!\searrow\!\overset{\bullet}{\overset{|}{\bullet}}\!\overset{1}{\downarrow}\!\swarrow^{1}k)- \tag{7}$$

$$-\sum_{k=0}^{n-2}(-1)^k a({}^{n-k-2}\bullet\!\overset{2}{\searrow}\!\overset{2}{\overset{\downarrow}{\bullet}}\!\overset{1}{\downarrow}\!\swarrow^{1}k)+ \tag{8}$$

$$+\sum_{k=0}^{n-2}(-1)^k (n-k-1)(k+1)a({}^{n-k-1}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k+1)- \tag{9}$$

$$-\sum_{k=0}^{n-2}(-1)^k \frac{(k+1)(k+2)}{2}a({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k+2). \tag{10}$$

The signs correspond to the number of ways we apply the IBP to obtain τ as each application of IBP changes the sign. The IBP has to applied k times to term (7), $k+1$ times to (8), $k+2$ times to (9), and $k+3$ times to (10). We have four terms each of which corresponds to a term in $\mathrm{Coef}(\overset{\bullet}{\overset{|}{\overset{\bullet}{\overset{|}{\bullet}}}})$. We recall that

$$a({}^{n-k-2}\bullet\!\searrow\!\overset{\bullet}{\overset{|}{\bullet}}\!\overset{1}{\downarrow}\!\swarrow^{1}k) = a(\overset{\bullet}{\overset{|}{\overset{\bullet}{\overset{|}{\bullet}}}})a({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k)$$

$$a({}^{n-k-2}\bullet\!\overset{2}{\searrow}\!\overset{2}{\overset{\downarrow}{\bullet}}\!\overset{1}{\downarrow}\!\swarrow^{1}k) = a({}^{1}\!\searrow\!\overset{1}{\overset{\downarrow}{\bullet}}\!\swarrow)a({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k)$$

$$a({}^{n-k-1}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k+1) = \frac{\hat\sigma({}^{n-k-1}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k+1)}{\hat\sigma({}^{1}\!\searrow\!\overset{1}{\downarrow}\!\swarrow\!\bullet)\hat\sigma({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k)}a({}^{1}\!\searrow\!\overset{1}{\downarrow}\!\swarrow\!\bullet)a({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k)$$

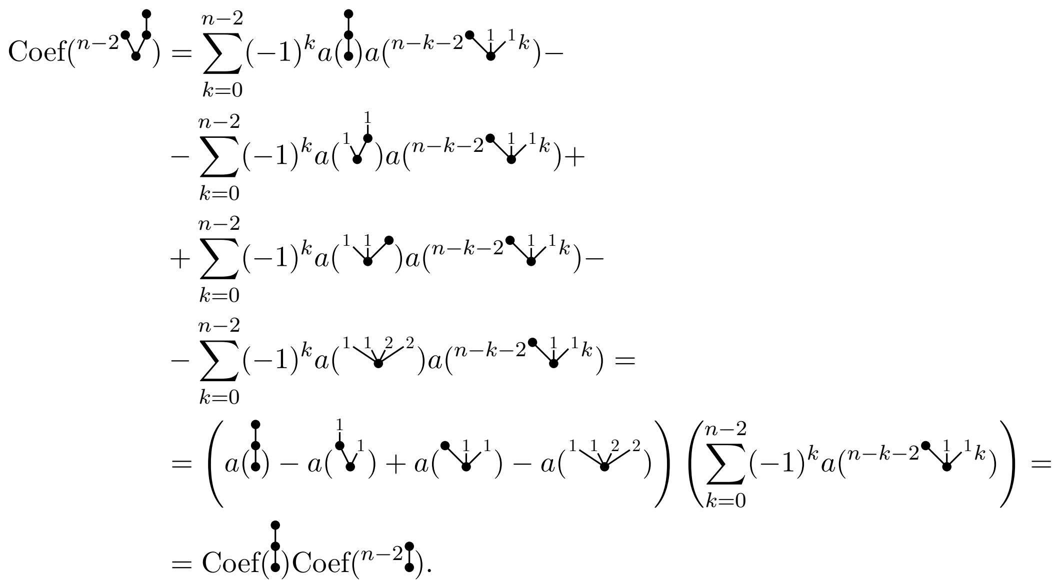
$$a({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k+2) = \frac{\hat\sigma({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k+2)}{\hat\sigma({}^{1}\!\searrow\!\overset{1}{\searrow}\!\overset{2}{\swarrow}\!\swarrow^{2})\hat\sigma({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k)}a({}^{1}\!\searrow\!\overset{1}{\searrow}\!\overset{2}{\swarrow}\!\swarrow^{2})a({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k).$$

where

$$\frac{\hat\sigma({}^{n-k-1}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k+1)}{\hat\sigma({}^{1}\!\searrow\!\overset{1}{\downarrow}\!\swarrow\!\bullet)\hat\sigma({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k)} = \frac{1\cdot k!(n-k-2)!}{(k+1)!(n-k-1)!} = \frac{1}{(k+1)(n-k-1)}$$

$$\frac{\hat\sigma({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k+2)}{\hat\sigma({}^{1}\!\searrow\!\overset{1}{\searrow}\!\overset{2}{\swarrow}\!\swarrow^{2})\hat\sigma({}^{n-k-2}\bullet\!\searrow\!\overset{1}{\downarrow}\!\swarrow^{1}k)} = \frac{2!\cdot(n-k-2)!k!}{(n-k-2)!(k+2)!} = \frac{2}{(k+2)!(k+1)!}.$$

Therefore, we can replace all $a(\gamma)$ in $\mathrm{Coef}(\tau)$ by respective $a(\gamma_1)a(\gamma_2)$ and get



This finishes the proof of Proposition 11. $\square$

Remark 2. Further analysis of the order conditions would require a general formula for $\mathrm{Coef}(\tau)$. By studying the Algorithm 1 and using its recursive nature a formula could be conjectured.

$$\mathrm{Coef}(\tau) = \sum_{\tau_l \in O_L^r(\tau)} a(\tau_l) - \sum_{v \in V_{V,N}^r(\tau)} \frac{\sigma_{s(v)}(\tau_v)}{\sigma_v(\tau)} \mathrm{Coef}(\tau_{v \to r})$$

where $O_L^r(\tau)$ is the set of trees where the lianas connected to the root are moved through the tree in all possible ways, $V_{V,N}^r(\tau)$ is the set of all numbered leaves not connected to the root and black leaves, $\tau_{v\to r}$ is the tree τ with v attached to the root, $s(v)$ is the successor of v in τ, and $\sigma_v(\tau)$ is the number of vertices that v is mapped to by automorphisms of τ. The first term is the dual of the first step of Algorithm 1 while the second term is the dual of IBP from second step of Algorithm 1.

Another possible way to generalize the result from Proposition 11 would be to prove

$$\mathrm{Coef}([\tau_1, \tau_2]) = \frac{\hat{\sigma}([\tau_1, \tau_2])}{\hat{\sigma}([\tau_1])\hat{\sigma}([\tau_2])} \mathrm{Coef}([\tau_1])\mathrm{Coef}([\tau_2]) \tag{11}$$

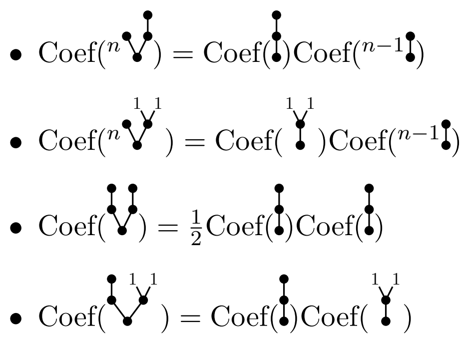
for a larger class of $(\tau_1, \tau_2) \in \mathcal{ET}^2$ by taking

$$\mathrm{Coef}(\tau) = \sum_{\gamma \in X} C(\tau, \gamma) a(\gamma)$$

where X is the set of trees that can be transformed into τ by Algorithm 1 and $C(\tau, \gamma)$ is the number of instances of τ that could be obtained from γ. There is no systematic way to get X and no general formula for $C(\tau, \gamma)$.

It is curious that the equation (11) is true for a few interesting examples:

- 



7 PyTreeHopf – operations on trees

The success of the introduction of trees in the study of deterministic numerical methods inspires new research directions that generalize the concept of a tree and use it to gain new results. Even though the idea of a tree is intuitive, the computations using trees can become messy very quickly, especially if the definition of the tree was expanded. This is the case with exotic trees. Table 7 shows the number of exotic trees by order.

1	2	3	4
2	6	21	85

Table 3: Number of exotic trees

Compare it to the number of rooted trees listed in Table 7.

2	3	4	5
1	2	4	9

Table 4: Number of rooted trees

Even a small number of trees could cause computational difficulties if the algorithm is difficult enough. The combinatorial nature of trees and of operations on them means that these operations could be easily programmed into a computer. In this section, we discuss the pyTreeHopf package the purpose of which is to simplify the automation of computations involving trees.

The general workflow of the package is shown in the figure 1. In order to apply the implemented tree operations to a linear combination of exotic trees, we first have to assign SymPy variables to trees and build the linear combination using the corresponding SymPy variables. Then the tree operations can be applied to the SymPy expression and return a SymPy expression as a result. The result can then be compiled into a PDF, PNG, and TEX output. An example of the usage of the symbolic package can be found in Appendix D.

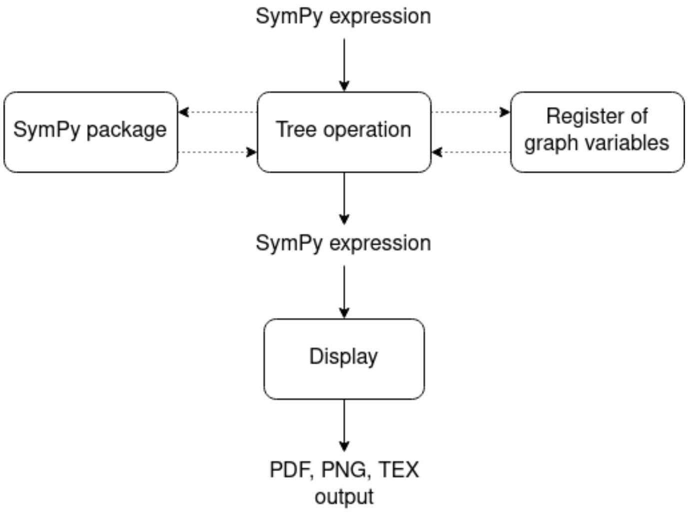


Figure 1: General workflow of the package

7.1 Algorithms related to exotic trees

One of the main uses of the package during the work on this master thesis was to generate the order conditions for invariant measure of ergodic SDE, confirm, and possibly extend the work done in [20]. However, the computations done in [20] were done manually and the task of implementing these computations proved to be non-trivial.

One step of the computations required us to have an algorithm to check if one exotic tree could be transformed using inversion of edge-liana into another exotic tree (or vice versa). Algorithm 3 does exactly that. Algorithm 3 uses Algorithm 2 described below.

The idea of Algorithm 2: We try to move the lianas of τ in a certain order to target positions from P in all possible ways. If, at some point for each way, we find a liana that cannot be moved then the algorithm returns False.

The idea of Algorithm 3: It was noticed that lianas change the direction of edges when they pass over them. To ignore this fact, we make all edges bidirectional. Then, the isomorphisms between the bidirectional trees are found. If there are none then the algorithm returns False. The isomorphisms should preserve the roots as lianas do not affect the roots. Then we try to check if lianas of the first tree can be moved to target positions given by isomorphisms from second tree in all possible orders. If we find an isomorphism and an order of lianas that make such movement possible then we return True. Otherwise, if all possible options were checked and none were suitable then we return False. Notice that the algorithm works only with lianas which are not loops. Looped lianas cannot be moved and the correspondence between looped lianas of two trees is not shown in the algorithm.

Algorithm 2: Check if it is possible to move the lianas of exotic tree τ to other positions in a certain order using edge-liana inversion.

Data: Exotic tree τ, an ordered list of lianas L^o, and an unorded list of new positions P.

Result: `True` if possible and `False` if not.

1. Let $T_0 := \{\tau\}$.
2. (A) For each liana l_i in the order L^o where i is its position in the order:

    2.1. Let $T_i := \emptyset$.

    2.2. (B) For each exotic tree γ in T_{i-1}:

    2.2.1. Get the orbit of liana l_i in γ and denote it by O_{l_i}.

    2.2.2. If $P \cap O_{l_i} \setminus \{p(l_j) \ : \ j < i\} = \emptyset$ then move to the next iteration of (B) (i.e. take another exotic tree from T_{i-1}).

    2.2.3. (C) For each position p in $P \cap O_{l_i} \setminus \{p(l_j) \ : \ j < i\}$:

    2.2.3.1. Move l_i to p in γ and add the result to T_i.

    2.3. If $T_i = \emptyset$ then return `False`.

3. return `True`.

Algorithm 3: Check if two exotic trees are similar through edge-liana inversion

Data: Exotic trees τ_1 and τ_2

Result: `True` if $\tau_1 \sim \tau_2$ or `False` otherwise

1. Let L_1 and L_2 be the sets of all lianas of τ_1 and τ_2, respectively.
2. Replace all lianas by bidirectional edges, make all edges of τ_1 and τ_2 bidirectional, and denote the results as τ_1^* and τ_2^*.
3. List all isomorphisms between τ_1^* and τ_2^* that preserve the roots. If there are no such isomorphisms, return `False`.
4. (A) For each such isomorphism $\phi : \tau_2^* \to \tau_1^*$:
    (B) For each possible order of L_1 denoted by L_1^o:

    4.1. Apply Algorithm 1 with $\tau = \tau_1$, $L^o = L_1^o$, and $P = p(\phi(L_2))$.

    4.2. If Algorithm 1 returns `True` then return `True`.

5. return `False`.

8 Conclusion

Let us review the content of the master thesis and present some suggestions for further work. The content of the master thesis was split into three main parts that covered: algebraic framework for the study of B-series, exotic B-series and order conditions for the invariant measure of ergodic SDE, and a symbolic package PyTreeHopf.

We have described the Butcher and Substitution groups formed by B-series and looked at the Hopf and pre-Lie algebraic structures related to them. We introduced a unique group homomorphism between the Butcher and Substitution groups and analyzed the corresponding Hopf and pre-Lie algebraic homomorphisms Φ and Ψ. We have proved that Φ is injective on the Calaque, Ebrahimi-Fard, Manchon Hopf algebra of non-rooted trees and found a nice formula for Ψ. An interesting task for the future would be to find a nice description of the image of Φ and look at the role of Φ and Ψ in the context of numerical analysis.

We have presented the exotic B-series, exotic trees, and the tools they provide for the computation of order conditions for invariant measure of ergodic SDE. We reformulated the operations and algorithms used in [20] to compute the order conditions for invariant measure for orders 2 and 3, and proved that this algorithm ends. We have also analysed the order conditions for invariant measure of Runge-Kutta methods and proved several instances of a curious multiplicative property that allows us to express conditions for high orders in terms of conditions for lower orders. It remains to prove that this multiplicative property is true in general. Another prospect would be to extend the theory to include a larger class of problems.

We have developed a symbolic package and implemented new algorithms on exotic trees in order to generate order conditions for invariant measure automatically. This allowed us to confirm the results obtained in [20] and generate a new result – order conditions for the invariant measure of Runge-Kutta methods for order 4. The package could serve as a powerful aid in the study of structures on trees as it is easy to use and to extend. Future versions of the package will strive for a greater number of available tools and a greater level of mathematical rigorousness.

References

[1] A. Abdulle, G. Vilmart, and K. C. Zygalakis. High order numerical approximation of the invariant measure of ergodic SDEs. *SIAM Journal on Numerical Analysis*, 52(4):1600–1622, 2014. ID: unige:41847.

[2] C. Brouder. Runge–Kutta methods and renormalization. *The European Physical Journal C*, 12(3):521–534, Feb 2000.

[3] K. Burrage and P. Burrage. High strong order explicit Runge-Kutta methods for stochastic ordinary differential equations. *Applied Numerical Mathematics*, 22(1):81 – 101, 1996. Special Issue Celebrating the Centenary of Runge-Kutta Methods.

[4] K. Burrage and P. M. Burrage. Order conditions of stochastic Runge-Kutta methods by B-series. *SIAM Journal on Numerical Analysis*, 38(5):1626–1646, 2001.

[5] J. Butcher. An algebraic theory of integration methods. *Math. Comp.*, 26:79–79, 01 1972.

[6] J. Butcher and G. Wanner. Runge-Kutta methods: some historical notes. *Applied Numerical Mathematics*, 22(1):113 – 151, 1996. Special Issue Celebrating the Centenary of Runge-Kutta Methods.

[7] D. Calaque, K. Ebrahimi-Fard, and D. Manchon. Two interacting Hopf algebras of trees: A Hopf-algebraic approach to composition and substitution of B-series. *Advances in Applied Mathematics*, 47(2):282 – 308, 2011.

[8] A. Cayley. Xxviii. on the theory of the analytical forms called trees. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 13(85):172–176, 1857.

[9] F. Chapoton and M. Livernet. Pre-Lie algebras and the rooted trees operad. *International Mathematics Research Notices*, 2001(8):395–408, 01 2001.

[10] P. Chartier, E. Hairer, and G. Vilmart. A substitution law for B-series vector fields. Research Report RR-5498, INRIA, 2005.

[11] P. Chartier, E. Hairer, and G. Vilmart. Algebraic structures of B-series. *Foundations of Computational Mathematics*, 10:407–427, 08 2010.

[12] A. Connes and D. Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Commun. Math. Phys.*, 199:203–242, 1998.

[13] K. Debrabant and A. Kværnø. B–series analysis of stochastic Runge–Kutta methods that use an iterative scheme to compute their internal stage values. *SIAM Journal on Numerical Analysis*, 47:181–203, 01 2008.

[14] K. Debrabant and A. Kværnø. Composition of stochastic B-series with applications to implicit Taylor methods. *Applied Numerical Mathematics*, 61(4):501 – 511, 2011.

[15] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations; 2nd ed.* Springer, Dordrecht, 2006.

[16] E. Hairer, S. P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.* Springer, Berlin, 2nd rev. ed. 1993. corr. 3rd printing edition, 1993. ID: unige:12346.

[17] E. Hairer and G. Wanner. On the Butcher group and general multi-value methods. *Computing*, 13(1):1–15, 1974. ID: unige:12533.

[18] L. Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables, Nov. 1918.

[19] Y. Komori, T. Mitsui, and H. Sugiura. Rooted tree analysis of the order conditions of ROW-type scheme for stochastic differential equations. *BIT*, 37:43–66, 03 1997.

[20] A. Laurent and G. Vilmart. Exotic aromatic B-series for the study of long time integrators for a class of ergodic SDEs. *Mathematics of Computation*, 89(321):169–202, Jun 2019.

[21] J. Mattingly, A. Stuart, and D. Higham. Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise. *Stochastic Processes and their Applications*, 101(2):185 – 232, 2002.

[22] A. Rößler. Stochastic Taylor expansions for the expectation of functionals of diffusion processes. *Stochastic Analysis and Applications*, 22(6):1553–1576, Jan 2004.

[23] A. Rößler. Rooted tree analysis for order conditions of stochastic Runge-Kutta methods for the weak approximation of stochastic differential equations. *Stochastic Analysis and Applications*, 24(1):97–134, Mar 2006.

[24] A. Rößler. Runge–Kutta methods for Itô stochastic differential equations with scalar noise. *BIT Numerical Mathematics*, 46:97–110, 03 2006.

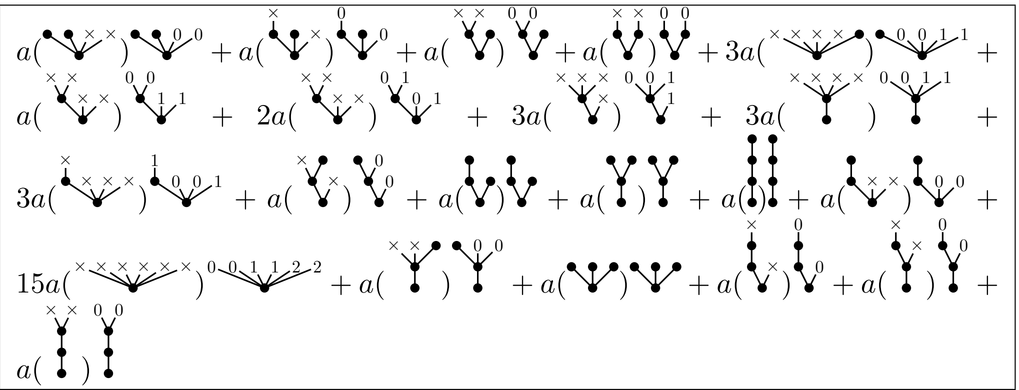
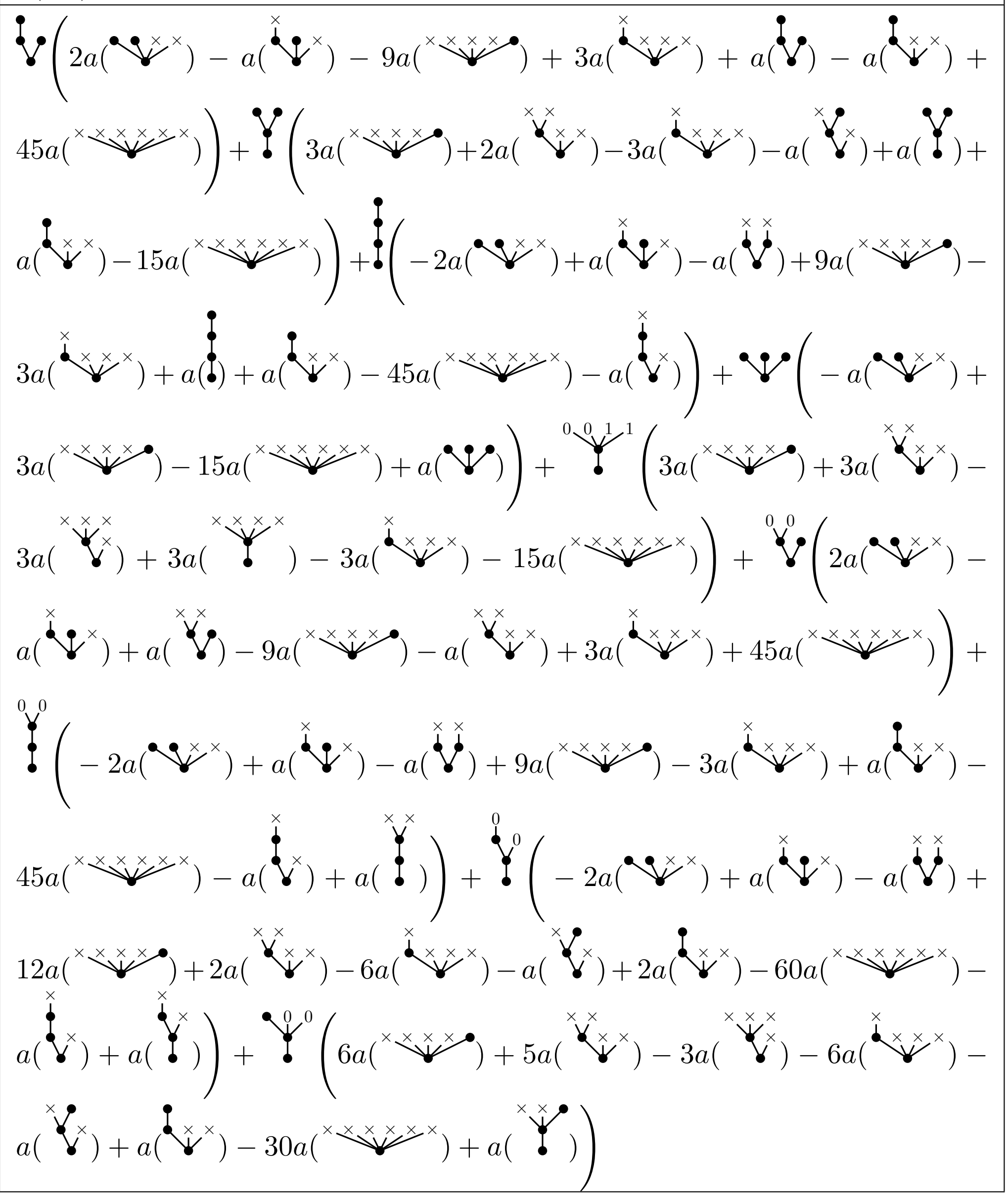
[25] A. Rößler. Strong and weak approximation methods for stochastic differential equations — Some recent developments. *Recent Developments in Applied Probability and Statistics - Dedicated to the Memory of Jurgen Lehn*, 01 2010.

[26] J. M. Sanz-Serna and A. Murua. Formal series and numerical integrators: some history and some new techniques, 2015.

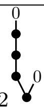
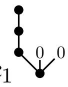
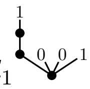
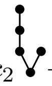
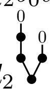
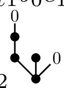
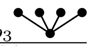
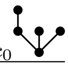
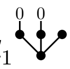
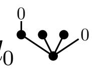
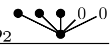
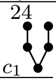
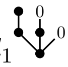
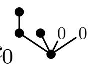
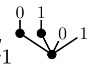
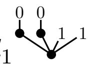
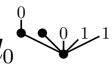
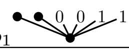
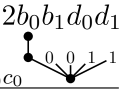
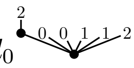
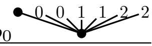
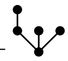
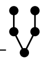
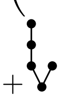
[27] D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Anal. Appl.*, 8, 01 1990.

Appendices

A Order conditions for order 3 for exotic B-series

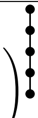
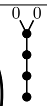
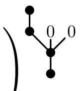
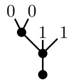
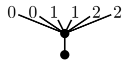
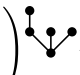
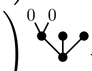
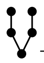
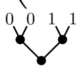
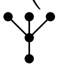
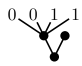
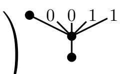
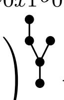
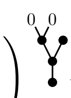
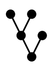
A_3	
Algo. 1 applied to A_3	

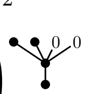
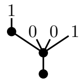
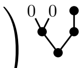
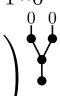
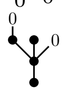
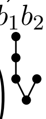
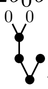
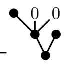
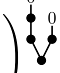
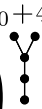
B Order conditions for order 4 for Runge-Kutta methods in Gaussian case

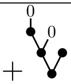
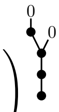
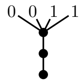
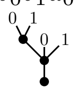
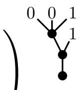
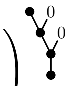
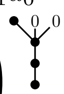
A_4	$a_{0x1}a_{1x2}b_0c_2$  $+ \; 2a_{0x1}a_{1x2}b_0d_2$  $+ \; a_{0x1}b_0c_1$  $+ \; 2a_{0x1}b_0d_1$  $+$ $a_{0x2}b_0b_1c_2$  $+2a_{0x2}b_0b_1d_1d_2$  $+2a_{0x2}b_0b_1d_2$  $+ \frac{b_0b_1b_2b_3}{24}$  $+ \frac{b_0b_1b_2c_0}{2}$  $+$ $b_0b_1b_2d_0d_1$  $+ b_0b_1b_2d_0$  $+ \frac{b_0b_1b_2}{6}$  $+ \frac{b_0b_1c_0c_1}{2}$  $+ 2b_0b_1c_0d_1$  $+$ $b_0b_1c_0$  $+ \; 2b_0b_1d_0d_1$  $+ \; b_0b_1d_0d_1$  $+ \; 2b_0b_1d_0$  $+$ $\frac{b_0b_1}{4}$  $+ \frac{b_0c_0}{2}$  $+ b_0d_0$  $+ \frac{b_0}{6}$  $+ \frac{1}{24}$ 
Algo. 1 applied to A_4	 $\left(\frac{b_0b_1b_2b_3}{24} - \frac{b_0b_1b_2}{6} + \frac{b_0b_1}{4} - \frac{b_0}{6} + \frac{1}{24} \right) +$  $\left(\frac{b_0b_1b_2c_0}{2} - b_0b_1b_2d_0 + \frac{b_0b_1b_2}{2} - b_0b_1c_0 + 2b_0b_1d_0 - \frac{5b_0b_1}{4} + \frac{b_0c_0}{2} - b_0d_0 + b_0 - \frac{1}{4} \right) +$  $\left(\frac{b_0b_1c_0c_1}{2} - 2b_0b_1c_0d_1 + b_0b_1c_0 + 2b_0b_1d_0d_1 - 2b_0b_1d_0 + \frac{b_0b_1}{2} - \frac{b_0c_0}{2} + b_0d_0 - \frac{b_0}{2} + \frac{1}{8} \right) +$  $\left(-a_{0x1}b_0c_1 + 2a_{0x1}b_0d_1 + a_{0x2}b_0b_1c_2 - 2a_{0x2}b_0b_1d_2 - b_0b_1b_2d_0d_1 + 2b_0b_1b_2d_0 - b_0b_1b_2 + b_0b_1c_0 + b_0b_1d_0d_1 - 4b_0b_1d_0 + \frac{5b_0b_1}{2} - b_0c_0 + 2b_0d_0 - 2b_0 + \frac{1}{2} \right) +$  $\left(a_{0x1}a_{1x2}b_0c_2 - 2a_{0x1}a_{1x2}b_0d_2 + a_{0x1}b_0c_1 - 2a_{0x1}b_0d_1 - 2a_{0x2}b_0b_1d_1d_2 + 2a_{0x2}b_0b_1d_2 + b_0b_1b_2d_0d_1 - 2b_0b_1b_2d_0 + b_0b_1b_2 + 2b_0b_1c_0d_1 - 2b_0b_1c_0 - 3b_0b_1d_0d_1 + 6b_0b_1d_0 - 3b_0b_1 + \frac{3b_0c_0}{2} - 3b_0d_0 + \frac{5b_0}{2} - \frac{5}{8} \right)$

C Order conditions for orders 3 and 4 for Runge-Kutta methods

γ_3	$a_{0x1}b_0c_1 + 2a_{0x1}b_0d_0d_1 + a_{0x1}b_0d_1^2 + 2a_{0x1}b_0d_1 + \frac{b_0b_1b_2}{6} + b_0b_1c_0 + b_0b_1d_0^2 + b_0b_1d_0d_1 + 2b_0b_1d_0 + \frac{b_0b_1}{2} + \frac{b_0c_0^2}{2} + b_0c_0d_0^2 + 2b_0c_0d_0 + b_0c_0 + \frac{b_0d_0^4}{6} + \frac{2b_0d_0^3}{3} + b_0d_0^2 + \frac{2b_0d_0}{3} + \frac{b_0}{6} + \frac{\;}{90}$
A_3	$a_{0x1}b_0c_1 + 2a_{0x1}b_0d_0d_1 + a_{0x1}b_0d_1^2 + 2a_{0x1}b_0d_1 + \frac{b_0b_1b_2}{6} + b_0b_1c_0 + b_0b_1d_0^2 + b_0b_1d_0d_1 + 2b_0b_1d_0 + \frac{b_0b_1}{2} + \frac{b_0c_0^2}{2} + b_0c_0d_0^2 + 2b_0c_0d_0 + b_0c_0 + \frac{b_0d_0^4}{2} + 2b_0d_0^3 + 2b_0d_0^2 + b_0d_0^2 + 2b_0d_0 + \frac{b_0}{2} + \frac{\;}{6}$
Algo. 1 applied to A_3	$\left(\frac{b_0b_1b_2}{6} - \frac{b_0b_1}{2} + \frac{b_0}{2} - \frac{1}{6}\right) + \left(\frac{b_0d_0^4}{2} - 2b_0d_0^3 + 3b_0d_0^2 - 2b_0d_0 + \frac{b_0}{2} - \frac{1}{6}\right) + \left(\frac{b_0c_0^2}{2} - 2b_0c_0d_0 + b_0c_0 + 2b_0d_0^2 - 2b_0d_0 + \frac{b_0}{2} - \frac{1}{6}\right) + \left(b_0b_1c_0 - 2b_0b_1d_0 + b_0b_1 - b_0c_0 + 2b_0d_0 - \frac{3b_0}{2} + \frac{1}{2}\right) + \left(b_0b_1d_0^2 - 2b_0b_1d_0 + b_0b_1 - b_0d_0^2 + 2b_0d_0 - \frac{3b_0}{2} + \frac{1}{2}\right) + \left(b_0c_0d_0^2 - 2b_0c_0d_0 + b_0c_0 - 2b_0d_0^3 + 5b_0d_0^2 - 4b_0d_0 + b_0 - \frac{1}{3}\right) + \left(a_{0x1}b_0c_1 - 2a_{0x1}b_0d_1 - b_0b_1d_0d_1 + 2b_0b_1d_0 - b_0b_1 + b_0c_0 - 2b_0d_0 + \frac{3b_0}{2} - \frac{1}{2}\right) + \left(a_{0x1}b_0d_1^2 - 2a_{0x1}b_0d_1 - b_0b_1d_0d_1 + 2b_0b_1d_0 - b_0b_1 + b_0c_0 - 2b_0d_0 + \frac{3b_0}{2} - \frac{1}{2}\right) + \left(2a_{0x1}b_0d_0d_1 - 2a_{0x1}b_0d_1 - b_0b_1d_0d_1 + 2b_0b_1d_0 - b_0b_1 - 2b_0c_0d_0 + 2b_0c_0 + 2b_0d_0^2 - 4b_0d_0 + 2b_0 - \frac{2}{3}\right)$

Algo. 1 applied to A_4	$\left(a_{0x1}a_{1x2}b_0c_2 - 2a_{0x1}a_{1x2}b_0d_2\right)$  $+ \left(a_{0x1}a_{1x2}b_0d_2^2 - 2a_{0x1}a_{1x2}b_0d_2\right)$  $+ \left(a_{0x1}b_0c_1d_0^2 - 2a_{0x1}b_0c_1d_0 + a_{0x1}b_0c_1 - 2a_{0x1}b_0d_0^2d_1\right)$  $+ \left(a_{0x1}b_0d_0^2d_1^2 - 2a_{0x1}b_0d_0^2d_1 - 2a_{0x1}b_0d_0d_1^2 + a_{0x1}b_0d_1^2\right)$  $+ \left(\frac{b_0b_1b_2b_3}{24} - \frac{b_0b_1b_2}{6} + \frac{b_0b_1}{4} - \frac{b_0}{6} + \frac{1}{24}\right)$  $+ \left(\frac{b_0d_0^6}{6} - b_0d_0^5 + \frac{5b_0d_0^4}{2} - \frac{10b_0d_0^3}{3} + \frac{5b_0d_0^2}{2} - b_0d_0 + \frac{b_0}{6} - \frac{1}{24}\right)$  $+ \left(\frac{b_0b_1b_2c_0}{2} - b_0b_1b_2d_0 + \frac{b_0b_1b_2}{2} - b_0b_1c_0 + 2b_0b_1d_0 - \frac{5b_0b_1}{4} + \frac{b_0c_0}{2} - b_0d_0 + b_0 - \frac{1}{4}\right)$  $+ \left(\frac{b_0b_1b_2d_0^2}{2} - b_0b_1b_2d_0 + \frac{b_0b_1b_2}{2} - b_0b_1d_0^2 + 2b_0b_1d_0 - \frac{5b_0b_1}{4} + \frac{b_0d_0^2}{2} - b_0d_0 + b_0 - \frac{1}{4}\right)$  $+ \left(\frac{b_0b_1c_0c_1}{2} - 2b_0b_1c_0d_1 + b_0b_1c_0 + 2b_0b_1d_0d_1 - 2b_0b_1d_0 + \frac{b_0b_1}{2} - \frac{b_0c_0}{2} + b_0d_0 - \frac{b_0}{2} + \frac{1}{8}\right)$  $+ \left(\frac{b_0b_1d_0^2d_1^2}{2} - 2b_0b_1d_0^2d_1 + b_0b_1d_0^2 + 2b_0b_1d_0d_1 - 2b_0b_1d_0 + \frac{b_0b_1}{2} - \frac{b_0d_0^2}{2} + b_0d_0 - \frac{b_0}{2} + \frac{1}{8}\right)$  $+ \left(\frac{b_0c_0^3}{6} - b_0c_0^2d_0 + \frac{b_0c_0^2}{2} + 2b_0c_0d_0^2 - 2b_0c_0d_0 + \frac{b_0c_0}{2} - \frac{4b_0d_0^3}{3} + 2b_0d_0^2 - b_0d_0 + \frac{b_0}{6} - \frac{1}{24}\right)$  $+ \left(\frac{b_0b_1d_0^4}{2} - 2b_0b_1d_0^3 + 3b_0b_1d_0^2 - 2b_0b_1d_0 + \frac{b_0b_1}{2} - \frac{b_0d_0^4}{2} + 2b_0d_0^3 - 3b_0d_0^2 + 2b_0d_0 - \frac{2b_0}{3} + \frac{1}{6}\right)$  $+ \left(\frac{b_0c_0d_0^4}{2} - 2b_0c_0d_0^3 + 3b_0c_0d_0^2 - 2b_0c_0d_0 + \frac{b_0c_0}{2} - b_0d_0^5 + \frac{9b_0d_0^4}{2} - 8b_0d_0^3 + 7b_0d_0^2 - 3b_0d_0 + \frac{b_0}{2} - \frac{1}{8}\right)$  $+ \left(a_{0x1}b_0c_0c_1 - 2a_{0x1}b_0c_0d_1 - 2a_{0x1}b_0c_1d_0 + a_{0x1}b_0c_1 + 4a_{0x1}b_0d_0d_1 + b_0c_0^2 - 4b_0c_0d_0 + b_0c_0 + 4b_0d_0^2 - 2b_0d_0 + \frac{b_0}{3} - \frac{1}{12}\right)$  $+ \left(a_{0x1}b_0c_0d_1^2 - 2a_{0x1}b_0c_0d_1 - 2a_{0x1}b_0d_0d_1^2 + 4a_{0x1}b_0d_0d_1 + a_{0x1}b_0d_1^2 + b_0c_0^2 - 4b_0c_0d_0 + b_0c_0 + 4b_0d_0^2 - 2b_0d_0 + \frac{b_0}{3} - \frac{1}{12}\right)$  $+ \left(\frac{b_0b_1c_0^2}{2} - 2b_0b_1c_0d_0 + b_0b_1c_0 + 2b_0b_1d_0^2 - 2b_0b_1d_0 + \frac{b_0b_1}{2} - \frac{b_0c_0^2}{2} + 2b_0c_0d_0 - b_0c_0 - 2b_0d_0^2 + 2b_0d_0 - \frac{2b_0}{3} + \frac{1}{6}\right)$  $+ \cdots$

Algo. 1 applied to A_4 (cont.)	$\cdots + \Big(\frac{b_0 c_0^2 d_0^2}{2} - b_0 c_0^2 d_0 + \frac{b_0 c_0^2}{2} - 2b_0c_0d_0^3 + 5b_0c_0d_0^2 - 4b_0c_0d_0 + b_0c_0 + 2b_0d_0^4 - 6b_0d_0^3 + \frac{13b_0d_0^2}{2} - 3b_0d_0 + \frac{b_0}{2} - \frac{1}{8}\Big)$  $+ \Big(2a_{0x1}b_0d_0^3d_1 - 6a_{0x1}b_0d_0^2d_1 + 2a_{0x1}b_0d_0d_1 - 2b_0c_0d_0^3 + 5b_0c_0d_0^2 - 4b_0c_0d_0 + b_0c_0 + 2b_0d_0^4 - 6b_0d_0^3 + \frac{13b_0d_0^2}{2} - 3b_0d_0 + \frac{b_0}{2} - \frac{1}{8}\Big)$  $+ \Big(-2b_0b_1c_0d_1 + b_0b_1c_0 + b_0b_1c_1d_0^2 - 2b_0b_1d_0^2d_1 + b_0b_1d_0^2 + 4b_0b_1d_0d_1 - 4b_0b_1d_0 + b_0b_1 - \frac{b_0c_0}{2} - \frac{b_0d_0^2}{2} + 2b_0d_0 - b_0 + \frac{1}{4}\Big)$  $+ \Big(a_{0x1}a_{0x2}b_0d_1d_2 + 2a_{0x1}a_{1x2}b_0d_0d_2 - 2a_{0x1}a_{1x2}b_0d_2 - 2a_{0x1}b_0c_0d_1 - 2a_{0x1}b_0c_1d_0 + a_{0x1}b_0c_1 + 4a_{0x1}b_0d_0d_1 + b_0c_0^2 - 4b_0c_0d_0 + b_0c_0 + 4b_0d_0^2 - 2b_0d_0 + \frac{b_0}{3} - \frac{1}{12}\Big)$  $+ \Big(2a_{0x1}b_0c_0d_0d_1 - 2a_{0x1}b_0c_0d_1 - 4a_{0x1}b_0d_0^2d_1 + 6a_{0x1}b_0d_0d_1 - 2b_0c_0^2d_0 + 2b_0c_0^2 + 6b_0c_0d_0^2 - 10b_0c_0d_0 + \frac{5b_0c_0}{2} - 4b_0d_0^3 + 10b_0d_0^2 - 5b_0d_0 + \frac{5b_0}{6} - \frac{5}{24}\Big)$  $+ \Big(-a_{0x1}b_0c_1 + 2a_{0x1}b_0d_1 + a_{0x2}b_0b_1c_2 - 2a_{0x2}b_0b_1d_2 - b_0b_1b_2d_0d_1 + 2b_0b_1b_2d_0 - b_0b_1b_2 + b_0b_1c_0 + b_0b_1d_0d_1 - 4b_0b_1d_0 + \frac{5b_0b_1}{2} - b_0c_0 + 2b_0d_0 - 2b_0 + \frac{1}{2}\Big)$  $+ \Big(-a_{0x1}b_0d_1^2 + 2a_{0x1}b_0d_1 + a_{0x2}b_0b_1d_2^2 - 2a_{0x2}b_0b_1d_2 - b_0b_1b_2d_0d_1 + 2b_0b_1b_2d_0 - b_0b_1b_2 + b_0b_1c_0 + b_0b_1d_0d_1 - 4b_0b_1d_0 + \frac{5b_0b_1}{2} - b_0c_0 + 2b_0d_0 - 2b_0 + \frac{1}{2}\Big)$  $+$  $\Big(b_0b_1c_0d_0^2 - 2b_0b_1c_0d_0 + b_0b_1c_0 - 2b_0b_1d_0^3 + 5b_0b_1d_0^2 - 4b_0b_1d_0 + b_0b_1 - b_0c_0d_0^2 + 2b_0c_0d_0 - b_0c_0 + 2b_0d_0^3 - 5b_0d_0^2 + 4b_0d_0 - \frac{4b_0}{3} + \frac{1}{3}\Big) + \Big(-a_{0x1}b_0c_1 + 2a_{0x1}b_0d_1 + 2a_{0x2}b_0b_1d_1d_2 - 2a_{0x2}b_0b_1d_2 - b_0b_1b_2d_0d_1 + 2b_0b_1b_2d_0 - b_0b_1b_2 - 2b_0b_1c_0d_1 + 2b_0b_1c_0 + 3b_0b_1d_0d_1 - 6b_0b_1d_0 + 3b_0b_1 - \frac{3b_0c_0}{2} + 3b_0d_0 - \frac{5b_0}{2} + \frac{5}{8}\Big)$  $+ \Big(\frac{a_{0x1}b_0c_1^2}{2} - 2a_{0x1}b_0c_1d_1 + a_{0x1}b_0c_1 + 2a_{0x1}b_0d_1^2 - 4a_{0x1}b_0d_1 - 2b_0b_1c_0d_0d_1 + 2b_0b_1c_0d_0 + 2b_0b_1c_0d_1 - 2b_0b_1c_0 + 4b_0b_1d_0^2d_1 - 4b_0b_1d_0^2 - 6b_0b_1d_0d_1 + 8b_0b_1d_0 - 2b_0b_1 + 2b_0c_0 - 4b_0d_0 + 2b_0 - \frac{1}{2}\Big)$  $+ \cdots$

Algo. 1 applied to A_4 (cont.)	$\cdots +$  $\Big(-2a_{0x1}b_0d_0d_1 + 2a_{0x1}b_0d_1 + 2a_{0x2}b_0b_1d_0d_2 - 2a_{0x2}b_0b_1d_2 - b_0b_1b_2d_0d_1 + 2b_0b_1b_2d_0 - b_0b_1b_2 - 2b_0b_1c_0d_0 + 2b_0b_1c_0 + 2b_0b_1d_0^2 + b_0b_1d_0d_1 - 6b_0b_1d_0 + 3b_0b_1 + 2b_0c_0d_0 - 2b_0c_0 - 2b_0d_0^2 + 4b_0d_0 - \frac{8b_0}{3} + \frac{2}{3}\Big) + \Big(2a_{0x1}a_{1x2}b_0d_1d_2 - 2a_{0x1}a_{1x2}b_0d_2 - 2a_{0x1}b_0c_1d_1 + a_{0x1}b_0c_1 + 2a_{0x1}b_0d_1^2 - 4a_{0x1}b_0d_1 - 2b_0b_1c_0d_0d_1 + 2b_0b_1c_0d_0 + 2b_0b_1c_0d_1 - 2b_0b_1c_0 + 4b_0b_1d_0^2d_1 - 4b_0b_1d_0^2 - 6b_0b_1d_0d_1 + 8b_0b_1d_0 - 2b_0b_1 + 2b_0c_0 - 4b_0d_0 + 2b_0 - \frac{1}{2}\Big)$  $+ \Big(4a_{0x1}b_0d_0d_1 + \frac{a_{0x1}b_0d_1^4}{2} - 2a_{0x1}b_0d_1^3 + 3a_{0x1}b_0d_1^2 - 4a_{0x1}b_0d_1 - 2b_0b_1d_0^3d_1 + 2b_0b_1d_0^3 + 6b_0b_1d_0^2d_1 - 6b_0b_1d_0^2 - 6b_0b_1d_0d_1 + 8b_0b_1d_0 - 2b_0b_1 + b_0c_0d_0^2 - 2b_0c_0d_0 + \frac{3b_0c_0}{2} - 2b_0d_0^3 + \frac{11b_0d_0^2}{2} - 6b_0d_0 + \frac{7b_0}{3} - \frac{7}{12}\Big)$  $+ \Big(2a_{0x1}b_0d_0^2d_1^2 - 4a_{0x1}b_0d_0^2d_1 - 4a_{0x1}b_0d_0d_1^2 + 8a_{0x1}b_0d_0d_1 + 2a_{0x1}b_0d_1^2 - 4a_{0x1}b_0d_1 - b_0b_1d_0^2d_1^2 + 4b_0b_1d_0^2d_1 - 2b_0b_1d_0^2 - 4b_0b_1d_0d_1 + 4b_0b_1d_0 - b_0b_1 + 2b_0c_0d_0^2 - 4b_0c_0d_0 + 2b_0c_0 - \frac{4b_0d_0^3}{3} + 4b_0d_0^2 - 4b_0d_0 + \frac{4b_0}{3} - \frac{1}{3}\Big)$  $+ \Big(2a_{0x1}b_0d_0d_1^3 - 6a_{0x1}b_0d_0d_1^2 + 10a_{0x1}b_0d_0d_1 - 2a_{0x1}b_0d_1^3 + 6a_{0x1}b_0d_1^2 - 8a_{0x1}b_0d_1 - 2b_0b_1d_0^3d_1 + 2b_0b_1d_0^3 - b_0b_1d_0^2d_1^2 + 10b_0b_1d_0^2d_1 - 8b_0b_1d_0^2 - 10b_0b_1d_0d_1 + 12b_0b_1d_0 - 3b_0b_1 + b_0c_0d_0^2 - 4b_0c_0d_0 + 3b_0c_0 - 2b_0d_0^3 + \frac{15b_0d_0^2}{2} - 9b_0d_0 + \frac{7b_0}{2} - \frac{7}{8}\Big)$  $+ \Big(2a_{0x1}b_0c_1d_0d_1 - 2a_{0x1}b_0c_1d_0 - 2a_{0x1}b_0c_1d_1 + 2a_{0x1}b_0c_1 - 4a_{0x1}b_0d_0d_1^2 + 6a_{0x1}b_0d_0d_1 + 4a_{0x1}b_0d_1^2 - 8a_{0x1}b_0d_1 - 2b_0b_1c_0d_0d_1 + 2b_0b_1c_0d_0 + 2b_0b_1c_0d_1 - 2b_0b_1c_0 - b_0b_1d_0^2d_1^2 + 8b_0b_1d_0^2d_1 - 6b_0b_1d_0^2 - 10b_0b_1d_0d_1 + 12b_0b_1d_0 - 3b_0b_1 - 2b_0c_0d_0 + \frac{7b_0c_0}{2} + 2b_0d_0^2 - 7b_0d_0 + \frac{19b_0}{6} - \frac{19}{24}\Big)$  $+ \Big(a_{0x1}b_0c_1d_1^2 - 2a_{0x1}b_0c_1d_1 + a_{0x1}b_0c_1 + 4a_{0x1}b_0d_0d_1 - 2a_{0x1}b_0d_1^3 + 5a_{0x1}b_0d_1^2 - 8a_{0x1}b_0d_1 - 2b_0b_1c_0d_0d_1 + 2b_0b_1c_0d_0 + 2b_0b_1c_0d_1 - 2b_0b_1c_0 - 2b_0b_1d_0^3d_1 + 2b_0b_1d_0^3 + 10b_0b_1d_0^2d_1 - 10b_0b_1d_0^2 - 12b_0b_1d_0d_1 + 16b_0b_1d_0 - 4b_0b_1 + b_0c_0d_0^2 - 2b_0c_0d_0 + \frac{7b_0c_0}{2} - 2b_0d_0^3 + \frac{11b_0d_0^2}{2} - 10b_0d_0 + \frac{13b_0}{3} - \frac{13}{12}\Big)$ 

D Example of a script using the symbolic package

```python
from pyTreeHopf.algorithms.graft import graft
from pyTreeHopf.algorithms.composition import composition
from pyTreeHopf.variables import get_graph_variable
from pyTreeHopf.input import from_brackets
from pyTreeHopf.output import display

from sympy.physics.quantum import TensorProduct

tree1 = from_brackets('b[b]')
tree2 = from_brackets('b[b,b[b,b]]')

t_1 = get_graph_variable(tree1)
t_2 = get_graph_variable(tree2)

graft_result = graft(TensorProduct(t_1, t_2))

composition_coproduct_result = composition(t_1 + t_2)

display(t_1)
display('$ \curvearrowright $')
display(t_2)
display(' = ')
display(graft_result)

display('\n\n')

display('$\Delta_{CK} ($')
display(t_1 + t_2)
display('$) = $')
display(composition_coproduct_result)
```

The output of this code is