Algebraic structures and numerical methods for invariant measure sampling of Langevin dynamics

Bronasco, Eugen

This publication URL: https://archive-ouverte.unige.ch/unige:185162
Publication DOI: 10.13097/archive-ouverte/unige:185162

UNIVERSITÉ DE GENÈVE
Section de mathématiques

FACULTÉ DES SCIENCES
Dr. Gilles Vilmart

CNRS et UNIVERSITÉ CLERMONT-AUVERGNE
(UMR 6620), France

Dr. Dominique Manchon

Algebraic structures and numerical methods for invariant measure sampling of Langevin dynamics

THÈSE

Présentée à la Faculté des Sciences de l'Université de Genève
pour obtenir le grade de Docteur ès Sciences, mention Mathématiques.

par
Eugen Bronasco
de
Balti (Moldavie)

Ph.D. N° 5897

Centre d'impression de l'UNIGE
Genève, 14.02.2025

UNIVERSITÉ
DE GENÈVE

FACULTÉ DES SCIENCES

DOCTORAT ÈS SCIENCES, MENTION MATHEMATIQUES

Thèse de Monsieur Eugen BRONASCO

intitulée :

«Algebraic Structures and Numerical Methods for Invariant Measure Sampling of Langevin Dynamics»

La Faculté des sciences, sur le préavis de

Monsieur G. VILMART, docteur et directeur de thèse
Section de mathématiques

Monsieur D. MANCHON, docteur et codirecteur de thèse
CNRS et Université Clermont-Auvergne, France

Monsieur B. VANDEREYCKEN, professeur associé
Section de mathématiques

Monsieur H. Z.. MUNTHE-KAAS, professeur
Department of Mathematics and Statistics, UiT The Arctic University of Norway, Tromsø, Norway and Department of Mathematics, University of Bergen, Bergen, Norway

Madame M. OTTOBRE, professeure
Mathematics Department, Heriot Watt University, Edinburgh, United Kingdom

autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 26 mars 2025

Thèse - 5897 -

Le décanat

N.B. - La thèse doit porter la déclaration précédente et remplir les conditions énumérées dans les "Informations relatives aux thèses de doctorat à l'Université de Genève".

Résumé de la thèse en français

L'objectif de cette thèse est de développer de nouveaux outils pour la construction d'intégrateurs numériques pour les équations différentielles stochastiques (EDS), atteignant un ordre élevé en convergence faible et/ou un ordre élevé pour l'échantillonnage de la mesure invariante. Ces outils s'appuient sur le cadre des forêts aromatiques exotiques afin d'analyser la convergence des intégrateurs pour l'équation de Langevin sur-amortie à diffusion constante.

Nous étendons ce cadre aux EDS avec bruit multiplicatif. Nous introduisons les séries de Butcher et les S-séries définies sur les forêts aromatiques exotiques, permettant d'étudier les structures algébriques sous-jacentes aux méthodes stochastiques de type Runge-Kutta. Ces structures, dans le contexte de la convergence faible et de l'échantillonnage de la mesure invariante, diffèrent fortement de celles rencontrées dans le cadre déterministe, où la structure pré-Lie des arbres décorés donne naissance à l'algèbre de Hopf de Grossman-Larson sur les forêts.

En exploitant la structure pré-Lie-Rinehart des arbres aromatiques décorés, nous construisons l'algèbre D-traciale libre, l'algébroïde de Hopf de Grossman-Larson des forêts aromatiques décorées ainsi que l'algébroïde pré-Hopf correspondant. En utilisant la dualité entre le coproduit de Butcher-Connes-Kreimer et le produit de Grossman-Larson, nous décrivons la loi de composition des S-séries définies sur les forêts aromatiques décorées. Ceci nous permet de formaliser la composition des méthodes numériques et le post-traitement dans le cadre des S-séries. De plus, nous introduisons les forêts agglomérées décorées et définissons la coaction de Calaque–Ebrahimi-Fard–Manchon (CEM) des forêts agglomérées et aromatiques, fournissant un formalisme pour la loi de substitution. Enfin, nous construisons l'espace des forêts aromatiques exotiques à partir des forêts aromatiques décorées et analysons l'impact de cette extension sur les structures algébriques sous-jacentes. Ces fondements algébriques permettent de décrire rigoureusement l'analyse d'erreur rétrograde et les techniques d'équations modifiées dans le cadre de l'échantillonnage de la mesure invariante.

Nous formalisons l'obtention des conditions d'ordre pour l'échantillonnage de la mesure invariante et présentons un algorithme systématique pour leur génération. En exploitant les propriétés de cet algorithme ainsi que la structure algébrique des forêts aromatiques exotiques, nous établissons la propriété multiplicative des conditions d'ordre. Ce résultat réduit considérablement le nombre de conditions nécessaires pour atteindre un ordre donné en échantillonnage de la mesure invariante.

À partir de ces développements, nous construisons un nouvel intégrateur pour les EDS à bruit multiplicatif. Cette méthode atteint l'ordre un en convergence faible et l'ordre deux pour l'échantillonnage de la mesure invariante, généralisant ainsi l'intégrateur de Leimkuhler-Matthews aux EDS avec bruit additif. Sa construction repose sur les concepts de forêts aromatiques exotiques et leurs propriétés algébriques.

L'approche initiale pour la construction de cette méthode consistait à résoudre 93 conditions d'ordre, mais cela s'est avéré irréalisable. Nous avons donc adopté une autre stratégie : partir d'une méthode d'ordre deux en convergence faible, déjà connue pour satisfaire les propriétés nécessaires, et la modifier afin de réduire son coût de calcul.

Nous analysons la stabilité du nouvel intégrateur et le validons numériquement sur une série de problèmes tests, démontrant ainsi son efficacité et sa précision. Notamment, la performance de la méthode reste indépendante de la dimension du problème, un avantage crucial dans le cadre de la dynamique moléculaire.

Pour gérer la complexité croissante des calculs associés au formalisme des forêts aromatiques exotiques pour les EDS à bruit multiplicatif, nous développons le package Arboretum.hs. Ce logiciel automatise les calculs usuels sur les algèbres de graphes, incluant l'implémentation des produits de greffage, de Grossman-Larson et d'insertion des forêts aromatiques décorées. Conçu comme un cadre flexible, Arboretum.hs se veut applicable à une grande variété de problèmes au-delà de ceux abordés dans cette thèse.

Les résultats présentés dans cette thèse sont publiés ou à paraître dans [7, 11, 12]. Le package Arboretum.hs est disponible publiquement sur GitLab:

`https://gitlab.unige.ch/Eugen.Bronasco/arboretum.hs`

Acknowledgements

The last four years have been among the most exciting of my life, filled with both personal and professional growth. As this PhD thesis marks the end of this wonderful chapter, I would like to take the opportunity to express my deepest gratitude to the people who made it so special.

I would like to thank my wife Aušra and my son Arthème for making my time during the PhD so much easier and happier. Their love laid the foundation for all my achievements by giving me the peace of mind and focus necessary to be productive. The impact of my wife's presence on my work cannot be overstated—she could be considered an indirect collaborator on everything I've done. One of the unexpected and indirect perks of having a little son was the two-hour walks I took with him every day while he slept. These walks gave me uninterrupted time to think while surrounded by nature. Many ideas came to me, and many difficult concepts were absorbed during these moments—time I would never have carved out otherwise.

I am also grateful to my parents, Valentin and Nelli, my brother Alex, and my friend Alex Usinevici for believing in me and encouraging me to pursue this path. Making them proud has always been one of my strongest motivations, and it ultimately helped me reach this important milestone.

I would like to thank my PhD supervisors, Gilles Vilmart and Dominique Manchon, for their guidance and availability. Every meeting—whether over Zoom or in person—opened new perspectives on what to do, correct, or understand. Their combined expertise in algebra and numerical analysis meant that no matter the direction I needed to explore, I always had a mentor to light the way. The care and attention I received made me feel like one of the luckiest doctoral students and gave me a model of what a mentor–student relationship should be.

I would also like to thank my supervisor Gilles Vilmart and the SNSF for supporting my participation in some of the best conferences in Europe and across the globe. This was a privilege I couldn't have even dreamed of before starting my PhD.

I would like to express my sincere gratitude to the members of my PhD jury—Hans Zanna Munthe-Kaas, Michela Ottobre, and Bart Vandereycken—for the time, care, and patience they devoted to reading my thesis and for participating in my defense. It was an honor to present my work before such an esteemed committee, and a privilege to receive their thoughtful feedback.

I am thankful for all the wonderful friends I met during my thesis who shared the office—and many great memories—with me (in alphabetical order): Benjamin Carrel, Ramona Häberli, Adrien Laurent, Liudi Lu, Xiaoyu Ma, Nicolas Masson, Yiying Wang, and Rik Voorhaar. The long lunches and durak games were often the highlights of slower days, rescuing me when I felt stuck. I cherish the memories of the trips we took and the conferences we attended together. The feedback I received on how to improve my presentations was an essential part of my growth—thank you all, and especially my wife, for that.

I would also like to thank my collaborators—Jean-Luc Falcone, Joscha Fregin, Adrien Laurent, Benedict Leimkuhler, Dominic Phillips, and Daniel Ruprecht—for the productive discussions and everything I learned through our work together. I'm also grateful to everyone who took the time to discuss my research during conferences. In particular, I would like to thank Geir Bogfjellmo, Elena Celledoni, Kurusch Ebrahimi-Fard, Hans Zanna Munthe-Kaas, Brynjulf Owren, Ludwig Rahm, and Akash Sharma for their insightful comments and generous exchanges—they have had a lasting impact on my thinking and work.

Last but not least, I would like to thank the secretariat, especially Joselle Besson, for their invaluable help with residence permit applications, conference travel, and all the administrative matters in between.

Contents

Chapter 1

Introduction

The aim of this thesis is to develop new tools for constructing time integrators for stochastic differential equations (SDEs) that achieve high weak order and/or high order for invariant measure sampling. These tools build upon the framework of exotic aromatic forests to analyze the convergence of integrators for the overdamped Langevin equation with constant diffusion.

We extend the framework of exotic aromatic forests to SDEs with multiplicative noise. We introduce Butcher series and S-series defined over exotic aromatic forests, allowing us to investigate the algebraic structures underlying stochastic Runge-Kutta-type methods. These structures, in the context of weak convergence and invariant measure sampling, differ significantly from those appearing in the deterministic setting, where the pre-Lie structure of decorated trees gives rise to the Grossman-Larson Hopf algebra of forests.

Using the pre-Lie-Rinehart structure of decorated aromatic trees, we construct the free tracial D-algebra, the Grossman-Larson Hopf algebroid of decorated aromatic forests, and the corresponding pre-Hopf algebroid. Leveraging the duality between the Butcher-Connes-Kreimer coproduct and the Grossman-Larson product, we describe the composition law for S-series over decorated aromatic forests. This enables us to formulate the composition of numerical methods and post-processing within the S-series framework. Furthermore, we introduce decorated clumped forests and define the Calaque–Ebrahimi-Fard–Manchon (CEM) coaction of clumped and aromatic forests, providing a formalism for the substitution law. Finally, we construct the space of exotic aromatic forests using decorated aromatic forests and analyze how this extension affects the underlying algebraic structures. These algebraic foundations allow us to rigorously describe backward error analysis and modified equation techniques in the context of invariant measure sampling.

We formalize the derivation of order conditions for invariant measure sampling and present a systematic algorithm for this process. By exploiting the properties of this algorithm and the algebraic structure of exotic aromatic forests, we establish the multiplicative property of order conditions. This result significantly reduces the number of conditions required to achieve a desired order for invariant measure sampling.

Using these insights, we construct a new integrator for SDEs with multiplicative noise. This method achieves weak order one and order two with respect to invariant measure sampling, generalizing the Leimkuhler-Matthews integrator for SDEs with additive noise. Its construction is based on the concepts of exotic aromatic forests and their algebraic properties.

The initial approach to constructing this method involved solving 93 order conditions, but this proved infeasible. Instead, we adopted an alternative strategy: starting with a weak order two method—already known to satisfy the necessary convergence properties—and modifying it to reduce computational cost.

We analyze the stability of the new integrator and validate it numerically on a set of test problems, demonstrating its efficiency and accuracy. Notably, the method's performance remains independent of the problem's dimensionality, a crucial advantage in the context of molecular dynamics.

To manage the increasing computational complexity associated with the exotic aromatic forests formalism for SDEs with multiplicative noise, we introduce the Arboretum.hs package. This software automates common computations involving algebras of graphs, including the implementations of grafting, Grossman-Larson, and insertion products of decorated aromatic forests. Crucially, Arboretum.hs is designed as a versatile framework, making it applicable to a variety of problems beyond those addressed in this thesis.

The results presented in this thesis are either published or under consideration for publication in [7, 11, 12]. The Arboretum.hs package is publicly available on GitLab:

`https://gitlab.unige.ch/Eugen.Bronasco/arboretum.hs`

1.1 Overdamped Langevin dynamics with variable diffusion

Langevin dynamics describes the movement of a particle in a potential field under the influence of friction and random noise. The noise term models the stochastic collisions between the particle and the surrounding medium. The system is governed by the Langevin equation:

$$dX(t) = P(X)dt, \quad dP(t) = -\nabla V(X)dt - \gamma P(X)dt + \sigma\sqrt{\gamma}\, dW(t),$$

where $V : \mathbb{R}^d \to \mathbb{R}$ is a potential function with a smooth and globally Lipschitz gradient $\nabla V : \mathbb{R}^d \to \mathbb{R}^d$. The parameter $\gamma > 0$ represents the friction coefficient, and σ is a constant related to the temperature. The term $W(t)$ is a standard d-dimensional Wiener process, and the initial conditions $(X(0), P(0))$ are deterministic. It can be shown that taking the limit as $\gamma \to \infty$ removes the inertia term, leading to the overdamped Langevin equation, also known as the Langevin equation without inertia. See [63] for more details. This limiting process results in the equation:

$$dX(t) = -\nabla V(X(t))dt + \sigma\, dW(t). \tag{1.1.1}$$

This equation describes Brownian motion, a model for the random movement of particles suspended in a fluid, introduced in 1905 by Einstein [30]. Appropriate assumptions on the potential guarantee the problem to be ergodic [52, 63] with the density of the unique invariant measure being $\rho_\infty = Z\exp(-\frac{2}{\sigma^2}V)$ where Z is such that $\int_{\mathbb{R}^d} \rho_\infty(x)dx = 1$.

Definition 1.1.1. 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,

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

In addition, the expectation of ϕ evaluated with the solution $X(t)$ typically converges exponentially fast as $t \to +\infty$ to the corresponding equilibrium average taken with respect to the invariant measure:

$$\left|\mathbb{E}\big(\phi(X(t))\big) - \int_{\mathbb{R}^d} \phi(x) d\pi(x)\right| \leq Ce^{-\lambda t} \tag{1.1.2}$$

for all $t > 0$ and constants $C, \lambda > 0$ independent of t (but C depends on ϕ and X_0).

To enhance the dynamics of $X(t)$ and improve its ability to sample the invariant measure, we consider position-dependent diffusion, which modifies the behavior of the noise without altering the invariant measure. This approach helps mitigate the issue of metastable trajectories that arise when the invariant measure contains multiple high-probability regions separated by low-probability regions. Such trajectories often undersample the invariant measure. The overdamped Langevin equation with a position-dependent diffusion matrix $D(x)$ (symmetric and positive definite) takes the form:

$$dX(t) = F(X(t))dt + \sigma D(X(t))^{1/2} dW(t), \qquad F = -D\nabla V + \frac{\sigma^2}{2}\operatorname{div}(D), \tag{1.1.3}$$

We consider a symmetric diffusion matrix $D(x) = (D_{ij}(x))_{i,j=1,\ldots,d}$ of size $d \times d$ with columns $D_j = (D_{ij})_{i=1,\ldots,d}$ for all $j = 1, \ldots d$ assumed smooth with respect to x. The *divergence* of the smooth matrix $D(x)$ is defined as the vector whose jth component is the divergence $\operatorname{div} D_j(x) = \sum_{i=1}^d \frac{\partial D_{ij}}{\partial x_i}(x)$, of the jth column D_j of the diffusion matrix D,

$$\operatorname{div}(D) = \begin{pmatrix} \operatorname{div} D_1 \\ \vdots \\ \operatorname{div} D_d \end{pmatrix}.$$

Defining the symmetric matrix $D(x)$ in the form $D = \Sigma^T \Sigma$, one can also consider the system

$$dX = -(\Sigma^T\Sigma)(X)\nabla V(X)dt + \frac{\sigma^2}{2}\operatorname{div}(\Sigma^T\Sigma)(X)dt + \sigma\Sigma(X)dW, \tag{1.1.4}$$

which is equivalent to (1.1.3) with symmetric diffusion matrix $\Sigma = D^{1/2}$. Without loosing generality, we shall assume for simplicity that Σ is symmetric. [1]

1.2 Invariant measure sampling and post-processing

We study ergodic numerical time integrators as a method for sampling the invariant measure of a Langevin equation. In this context, we recall the definition of an ergodic integrator and outline several standard assumptions.

Definition 1.2.1. A numerical method $X_1 = \Psi_h(X_0)$ 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,

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

[1] Indeed, observe that replacing $\Sigma(x)$ by the symmetric positive definite matrix $(\Sigma(x)^T\Sigma(x))^{1/2}$ in (1.1.4) does not change the law of the solution $X(t)$.

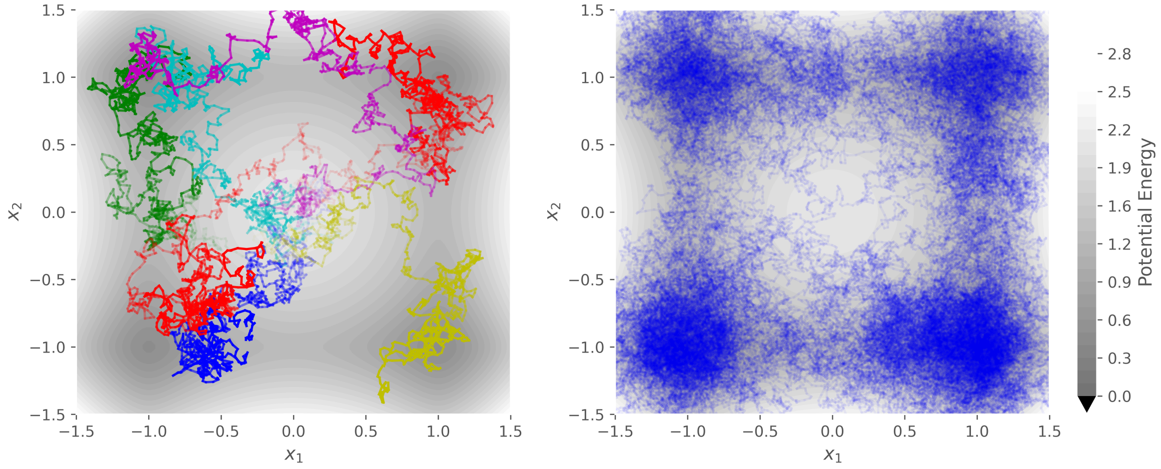


Figure 1.2.1: Trajectories generated by the new method introduced in Chapter 2, solving the overdamped Langevin equation (1.1.1) with $\sigma = 1$ using a stepsize of $h = 0.001$ and the potential $V(x)$ defined in Section 2.3.2. The plot on the left displays seven short trajectories ($T = 1$) all starting from the center, while the plot on the right shows a single long trajectory ($T = 100$) which illustrates the ergodicity of the overdamped Langevin dynamics.

See [52] for more details.

Figure 1.2.1 shows an example of trajectories generated by the new method introduced in Chapter 2 solving the overdamped Langevin equation with additive noise with the potential $V(x)$ defined in Section 2.3.2.

The following assumptions are commonly used in the analysis of the weak error of numerical integrators and are typically satisfied for stochastic Runge-Kutta type methods.

Assumption 1.2.2. *The integrator $X_1 = \Psi_h(X_0)$ has bounded moments of any order along time, i.e., for all integer $n \geq 0$,*

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

We recall that the exact solution has the weak Taylor expansion of the form

$$\mathbb{E}[\phi(X(h))|X_0 = x] = \phi(x) + h\mathcal{L}\phi(x) + h^2\frac{\mathcal{L}^2\phi(x)}{2!} + \cdots + h^k\frac{\mathcal{L}^k\phi(x)}{k!} + \cdots ,$$

with the generator $\mathcal{L}\phi := F \cdot \nabla\phi + \frac{\sigma^2}{2}\sum_{a=1}^{d}\phi''(\Sigma_a, \Sigma_a)$ coming from the backward Kolmogorov equation, which states that the map $u(x,t)$ defined by

$$u(x,t) := \mathbb{E}[\phi(X(t)) : X(0) = x]$$

is the solution of the following deterministic parabolic PDE in $\mathbb{R}^d$:

$$\frac{\partial u}{\partial t} = \mathcal{L}u, \quad u(x,0) = \phi(x), \quad x \in \mathbb{R}^d, t > 0\,.$$

The convergence analysis, in the weak sense and for the invariant measure sampling, is based on the weak Taylor expansion of the integrator and the exact solution. Let

$C_P^\infty(\mathbb{R}^d, \mathbb{R})$ be the space of smooth test functions such that all partial derivatives up to all orders have a polynomial growth, that is, for $\phi \in C_P^\infty(\mathbb{R}^d, \mathbb{R})$, we have,

$$|\frac{\partial^n \phi}{\partial x_{i_1} \cdots \partial x_{i_n}}| \leq C(1 + |x|^s),$$

for some s and C independent of x.

Assumption 1.2.3. *The integrator $X_1 = \Psi_h(X_0)$ 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 [75]. We assume that $\mathcal{A}_1 = \mathcal{L}$, that is, the integrator has at least weak order 1.

We note that an integrator $X_1 = \Psi_h(X_0)$ has weak order p if

$$\mathcal{A}_k = \frac{\mathcal{L}^k}{k!}, \quad \text{for all } k = 1, \ldots, p\,.$$

Definition 1.2.4. A numerical method $X_1 = \Psi_h(X_0)$ has order p with respect to the invariant measure of the SDE if

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

where C is independent of h assumed small enough.

We rely on the following theorem, which establishes the order conditions for invariant measure sampling. We let $\mathcal{A}_j^*$ be the adjoint of $\mathcal{A}_j$ with respect to the L_2 inner product,

$$\langle \phi_1, \phi_2 \rangle_{L_2} := \int_{\mathbb{R}^d} \phi_1(x)\phi_2(x)dx\,.$$

Theorem 1.2.5. *[2] Consider an ergodic integrator $X_1 = \Psi_h(X_0)$. Assume that Assumptions 1 and 2 hold. If, for $j = 2, \ldots, p$, we have $\mathcal{A}_j^*\rho_\infty = 0$, then the integrator achieves order p with respect to the invariant measure.*

We note that $\mathcal{L}^*\rho_\infty = 0$, which implies that an integrator of weak order p has at least order p with respect to the invariant measure. We also note that several sources of error arise when sampling the invariant measure, including bias due to the integrator's order, Monte Carlo error, and the rate λ in (1.1.2) of convergence to the invariant measure. The Monte Carlo error, which scales as $M^{-\frac{1}{2}}$, can be reduced by increasing the number of trajectories M used to approximate the expectation. The rate of convergence to the invariant measure can be improved by choosing an appropriate diffusion matrix D, however, this aspect is not addressed in this thesis.

Post-processing is a powerful technique originating from the deterministic context, where it is paired with pre-processing. It was introduced by Butcher in [17] in the context of ODEs. Given an integrator Ψ_h, we can increase the order of integration by finding perturbations of the identity χ_h and χ_h^{-1}, which are called post- and pre-processors and are inverses of each other. These are composed with Ψ_h as follows:

$$\Psi_h^{(2)} := \chi_h \circ \Psi_h \circ \chi_h^{-1}\,.$$

Notably, computing n steps of the new integrator $\Psi_h^{(2)}$ incurs the same computational cost as n steps of Ψ_h, assuming the computational cost of χ_h and χ_h^{-1} is negligible.

Post-processing was extended to the stochastic context in [79]. In the context of SDEs, particularly for invariant measure sampling, preprocessing is omitted as it does not influence the long-time behavior of the integrator. Post-processing, however, is employed to increase the order of the integrator with respect to the invariant measure, as described in the following theorem. To simplify notation, let $\langle \mathcal{A}_j \rangle$ denote the integral with respect to the invariant measure over $\mathbb{R}^d$ of the differential operator $\mathcal{A}_j$ applied to an arbitrary test function ϕ, that is,

$$\langle \mathcal{A}_j \rangle := \int_{\mathbb{R}^d} \mathcal{A}_j[\phi](x)\rho_\infty(x)\,dx\,.$$

Theorem 1.2.6. *[79] Assume the hypotheses of Theorem 1.2.5 and consider a post-processor χ_h that admits the following weak Taylor expansion for all $C_P^\infty(\mathbb{R}^d)$:*

$$\mathbb{E}[\phi(\chi_h(X_0))] = \phi(X_0) + \sum_{i=1}^{p-1} \alpha_i h^i \mathcal{L}^i \phi(X_0) + h^p \overline{\mathcal{A}}_p \phi(X_0) + \cdots\,,$$

for some constants α_i and differential operator $\overline{\mathcal{A}}_p$. Assume further that

$$\langle \mathcal{A}_{p+1} + [\mathcal{L}, \overline{\mathcal{A}}_p] \rangle = 0\,,$$

where $[\mathcal{L}, \overline{\mathcal{A}}_p] = \mathcal{L}\overline{\mathcal{A}}_p - \overline{\mathcal{A}}_p\mathcal{L}$ is the Lie bracket. Then, $\chi \circ \Psi_h$ yields an approximation of order $p+1$ for the invariant measure.

An example of a post-processed integrator is the Leimkuhler-Matthews method, which is of order 2 with respect to the invariant measure and of order 1 in the weak sense. The original non-Markovian formulation, introduced in [45, 44], is

$$X_{n+1} = X_n - h\nabla V(X_n) + \sqrt{h}\sigma \frac{\xi_n + \xi_{n+1}}{2}. \tag{1.2.1}$$

We can write it in a post-processed form, with $\overline{X}_{n+1}$ being the output, as

$$\begin{aligned} X_{n+1} &= X_n - h\nabla V(\overline{X}_n) + \sqrt{h}\sigma\xi_n, \\ \overline{X}_n &= X_n + \frac{1}{2}\sqrt{h}\sigma\xi_n, \end{aligned} \tag{1.2.2}$$

by substituting $X_{n+1} = \overline{X}_{n+1} - \frac{1}{2}\sqrt{h}\sigma\xi_{n+1}$ and $X_n = \overline{X}_n - \frac{1}{2}\sqrt{h}\sigma\xi_n$ into the first equation of (1.2.2).

We introduce a generalization of method (1.2.2) of order 2 with respect to the invariant measure and weak order 1 for the case of a position-dependent matrix Σ in (1.1.4). The new method is defined as:

$$\begin{aligned} X_{n+1} &= X_n + hF(\overline{X}_n) + \hat{\Phi}_h^\Sigma(X_n + \frac{1}{4}hF(\overline{X}_{n-1})), \\ \overline{X}_n &= X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n, \quad \text{with } \overline{X}_{-1} = X_0, \end{aligned} \tag{1.2.3}$$

where $\Phi_h^\Sigma(X_n) = X_n + \hat{\Phi}_h^\Sigma(X_n)$ is an integrator of weak order 2 applied to the SDE problem with noise only,

$$dX = \sigma\Sigma(X)dW, \tag{1.2.4}$$

where $\Phi^{\Sigma}(X_0) = X_0 + \sqrt{h}\sigma\Sigma(X_0)\xi_n + O(h)$. The output of the method (1.2.3) is given by $\overline{X}_{n+1}$ and it requires one evaluation of F per timestep with the number of evaluations of Σ depending on our choice of Φ_h^{Σ}. We remark that the new method (1.2.3) uses a similar postprocessor compared to (1.2.2),

$$\overline{X}_n = \Psi_h(X_n) = X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n, \tag{1.2.5}$$

and that it becomes equivalent to (1.2.2) for the additive noise case $\Sigma(x) = I$.

1.3 The framework of B-series and S-series

The correspondence between the elementary differentials in the Taylor expansion of the exact solution of an ODE and non-planar rooted trees is a classical result [20]. In the 1960s, Butcher used this correspondence to formulate the order conditions for Runge-Kutta methods [16] for arbitrary order, enabling the construction of many new high-order integrators. This approach was further developed by Hairer and Wanner [33], who introduced the modern concept of Butcher series (B-series) and described the underlying algebraic structures. In particular, the composition of Runge-Kutta methods, as well as other B-series, can be expressed using the Butcher-Connes-Kreimer Hopf algebra [26], introduced in the context of renormalization in quantum field theory.

The theory of B-series gained renewed interest in the 1990s with the rise of geometric numerical integration [34], as B-series proved to be a powerful tool for describing symplectic integrators, as well as for backward error analysis and modified equation techniques.

There are several equivalent ways to introduce the B-series framework. We adopt an approach based on the algebraic structures of the space of vector fields, following the perspective of [50]. Let us consider the space $\mathcal{X}$ of vector fields on $\mathbb{R}^d$. Let $f, g \in \mathcal{X}$ and let $f[g]$ denote the differentiation of g in the direction of f, that is, for $p \in \mathbb{R}^d$, we have

$$f[g](p) = \big(f(p)[g]\big)(p) = \sum_{i=1}^{d} f^i(p)\partial_i g(p), \quad \text{with } \partial_i g := \frac{\partial g}{\partial x_i}\,.$$

This way, vector fields define differential operators of degree one. The differential operators of higher degrees can be obtained by pointwise composition of vector fields, for example, let $f, g, h \in \mathcal{X}$ and $\partial_{ij} := \partial_i\partial_j$, then, for $p \in \mathbb{R}^d$, we have

$$(fg)[h](p) = \big((f(p)g(p))[h]\big)(p) = \sum_{i,j=1}^{d} f^i(p)g^j(p)\partial_{ij}h(p)\,,$$

From now on we will omit writting p and the differentiation will be written as

$$f[g] = \sum_{i=1}^{d} f^i\partial_i g, \quad \text{and} \quad fg[h] = \sum_{i,j=1}^{d} f^i g^j \partial_{ij} h\,.$$

Due to the fact that the pointwise composition is commutative, differentiation is a pre-Lie product, that is, it satisfies the following relation

$$f[g[h]] - f[g][h] = g[f[h]] - g[f][h], \quad \text{for } f, g, h \in \mathcal{X}\,.$$

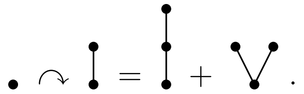
We consider an initial value ODE of the form

$$\frac{dy}{dt} = f(y), \quad y(0) = y_0. \tag{1.3.1}$$

The elementary differentials that appear as terms in the Taylor expansion of $y(h)$ around 0 form a pre-Lie algebra with the product given by differentiation. For example,

$$hf[h^2 f'f] = h^3 f'f'f + h^3 f''(f,f)\,.$$

Let us consider the pre-Lie algebra of non-planar rooted trees $(\mathcal{T}, \curvearrowright)$ with $\curvearrowright$ being the grafting product on trees defined by attaching the root of the left operand to a vertex of the right operand in all possible ways, for example,



We extend the grafting product to the commutative algebra of forests $(\mathcal{F}, \cdot)$ which is the symmetric algebra on trees, $(\mathcal{F}, \cdot) := S_{\mathbb{R}}(\mathcal{T})$. Let $\tau \in \mathcal{T}$ and $\pi_1, \pi_2 \in \mathcal{F}$, then

$$(\tau \cdot \pi_1) \curvearrowright \pi_2 = \tau \curvearrowright (\pi_1 \curvearrowright \pi_2) - (\tau \curvearrowright \pi_1) \curvearrowright \pi_2\,,$$

$$\tau \curvearrowright (\pi_1 \cdot \pi_2) = (\tau \curvearrowright \pi_1) \cdot \pi_2 + \pi_1 \cdot (\tau \curvearrowright \pi_2)\,.$$

We note that this definition of grafting on forests is well-defined since grafting is a pre-Lie product. The details can be found in [61].

Remark 1.3.1. *The algebra $(\mathcal{F}, \cdot, \curvearrowright)$ forms a commutative version of D-algebra structure introduced in [56].*

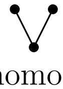
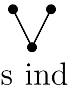
In [21], it is proven that the algebra $(\mathcal{T}_n, \curvearrowright)$ with $n-$colored trees is the free pre-Lie algebra with n generators. Therefore, there exists a surjective homomorphism from the pre-Lie algebra $(\mathcal{T}, \curvearrowright)$ onto the pre-Lie algebra of elementary differentials generated by hf. The homomorphism is extended to $(\mathcal{F}, \cdot, \curvearrowright)$ by sending the commutative product to the pointwise composition product of vector fields. The homomorphism is denoted by $\mathbb{F}$ and we give the explicit formula in Definition 1.3.2. The homomorphism $\mathbb{F}$ should not be confused with the expectation $\mathbb{E}$. Let us use the following notation, $[d] := \{1, \ldots, d\}$. Let $B_{\bullet}^{+} : F \to T$ be defined as $B_{\bullet}^{+}(\pi) = \pi \curvearrowright \bullet$ for $\pi \in F$.

Definition 1.3.2. Let $\tau = B_{\bullet}^{+}(\gamma_1 \cdots \gamma_n)$ be a tree, then,

$$\mathbb{F}(\tau) = \mathbb{F}(\gamma_1) \cdots \mathbb{F}(\gamma_n)[hf]\,,$$

analogously, using the fact that $f[g] = \sum_{i=1}^{d} f^i \partial_i g$ for vector fields f and g, we have

$$\mathbb{F}(\tau) = h \sum_{i_1, \ldots, i_n = 1}^{d} \mathbb{F}(\gamma_1)^{i_1} \cdots \mathbb{F}(\gamma_n)^{i_n} \partial_{i_1 \cdots i_n} f\,.$$

For example, $\mathbb{F}($  $) = h^3 \sum_{i,j=1}^{d} f^i f^j \partial_{ij} f$, or, analogously, $\mathbb{F}($  $) = h^3 f''(f,f)$.

We use the homomorphism $\mathbb{F}$ to define B-series as formal sums indexed by rooted non-planar trees, which represent the Taylor expansion of numerical integrators or exact solutions. Consequently, B-series establish a connection between numerical analysis and combinatorial algebra, linking the properties of numerical integrators to the algebraic and combinatorial structures of trees and forests. This framework provides a combinatorial representation of order conditions by comparing the coefficients of the B-series associated with the numerical integrator and the exact solution. Throughout this thesis, we will use this link to relate the described algebraic structures to applications in numerical analysis.

Definition 1.3.3. [16, 33] B-series are formal sums of vector fields of the following form

$$B(a) = \sum_{\tau \in T} \frac{a(\tau)}{\sigma(\tau)} \mathbb{F}(\tau)\,,$$

where T is the set of rooted non-planar trees, $a : T \to \mathbb{R}$ is a functional, and $\sigma(\tau)$ is the size of the automorphism group of τ.

The exact solution $y(h)$ and one-step of a Runge-Kutta method $\Psi_h(A, b, f)$ can be expanded using B-series as $y_0 \mapsto y_0 + B(a)(y_0)$ with the functionals $a : T \to \mathbb{R}$ defined appropriately. The concept of S-series was used to study the first integrals of B-series [60]. Let $I : \mathbb{R}^d \to \mathbb{R}$ be a first integral, then we have the following property

$$I\big(y_0 + B(a)(y_0)\big) = S(a)[I](y_0) = \sum_{\pi \in F} \frac{a(\pi)}{\sigma(\pi)} \mathbb{F}(\pi)[I](y_0)\,,$$

where $S(a)$ is called an S-series, F is the set of forests, the functional $a : F \to \mathbb{R}$ extends to forests by $a(\pi_1 \cdot \pi_2) = a(\pi_1)a(\pi_2)$ for $\pi_1, \pi_2 \in F$. We note that $y_0 + B(a)(y_0) = S(a)[\mathrm{Id}](y_0)$ where Id is the identity $\mathrm{Id}(x) = x$. Similar ideas are used to write the flow of a differential equation as the exponential of $\mathbb{F}(\bullet)$, i.e.

$$y(h) = \exp\big(\mathbb{F}(\bullet)\big)\mathrm{Id}(y_0) = S(\alpha)[\mathrm{Id}](y_0)\,,$$

where α is an appropriate functional on forests. The details can be found in Chapter III.5.1 of [34] in the context of the Baker-Campbell-Hausdorff formula for splitting integrators. We note that in a stochastic context, we can replace the first integral I with a test function ϕ and use S-series to study the expectation of a functional of one-step of a numerical integrator, i.e. $\mathbb{E}[\phi(y_0 + B(a)(y_0))]$, using its weak Taylor expansion [75].

An important feature of B-series and S-series is that they are completely characterized by the functionals $a : T \to \mathbb{R}$. This allows us to use combinatorial properties and algebraic structures on trees and forests to study the properties and operations of numerical integrators. For example, an important property of $\mathbb{F}$ is presented in the following proposition.

Proposition 1.3.4. *[61] Let $\diamond$ denote the Grossman-Larson product, then,*

$$\mathbb{F}(\pi_1 \diamond \pi_2)[\cdot] = \mathbb{F}(\pi_1)\big[\mathbb{F}(\pi_2)[\cdot]\big]\,,$$

where π_1 and π_2 are forests.

The Grossman-Larson product is defined as

$$\pi \diamond \eta = \sum_{(\pi)} \pi_{(1)} \cdot \left(\pi_{(2)} \curvearrowright \eta\right),$$

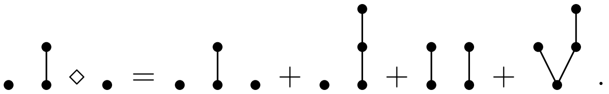
for $\pi, \eta \in F$ and $\Delta(\pi) = \sum_{(\pi)} \pi_{(1)} \otimes \pi_{(2)}$ being the deshuffle coproduct in Sweedler notation. The deshuffle coproduct is defined as

$$\Delta(\tau) = \tau \otimes \mathbf{1} + \mathbf{1} \otimes \tau \,,$$

where $\tau \in T$ is a tree, and, for $\pi, \eta \in F$,

$$\Delta(\pi \cdot \eta) = \Delta(\pi) \cdot \Delta(\eta) \,,$$

where concatenation $\cdot$ is defined on $F \otimes F$ by component-wise concatenation. For example,



1.4 Exotic forests

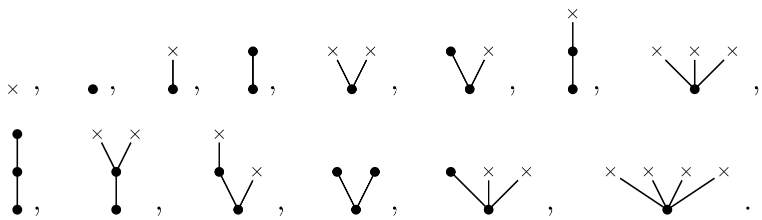
Exotic forests where first introduced in [41] for the generation of order conditions for invariant measure sampling of Langevin dynamics with additive noise. Similar formalisms using trees were introduced in [14, 67] in the context of finite time weak and strong integrators. In contrast, we focus in this work on the invariant measure accuracy. We define exotic forests using the concept of decorated forests, which are $\alpha : V(\pi) \to D$ that send vertices of π to decorations from the set D which is defined depending on the type of forests we want to represent. The set of decorated forests with an abstract set D is denoted by F_D and the vector space by $\mathcal{F}_D$.

Definition 1.4.1. A morphism $\varphi : (\pi_1, \alpha_1) \to (\pi_2, \alpha_2)$ between two decorated forests is a morphism between the forests $\varphi : \pi_1 \to \pi_2$ such that $\alpha_1 = \alpha_2 \circ \varphi$.

Let us consider the space of bicolored forests $\mathcal{F}_{\bullet,\times}$ spanned by forests $\pi \in F$ together with decorations $\alpha_g : V(\pi) \to \{\bullet, \times\}$. Let the space of grafted forests be defined as the quotient space $\mathcal{F}_g := \mathcal{F}_{\bullet,\times}/\mathcal{K}_\times$ with

$$\mathcal{K}_\times := \mathrm{span}\{(\pi, \alpha_g) \in F_{\bullet,\times} \;:\; \exists (v,u) \in E(\pi),\, \alpha_g(u) = \times\} \,.$$

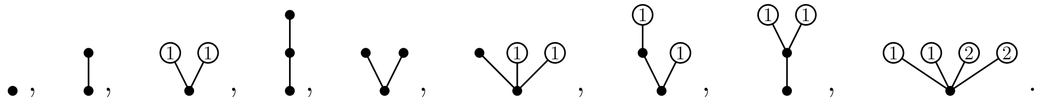
That is, grafted forests are bicolored forests (π, α_g) for which $\alpha_g^{-1}(\times)$ is a subset of leaves of π. For example, some grafted trees are listed below



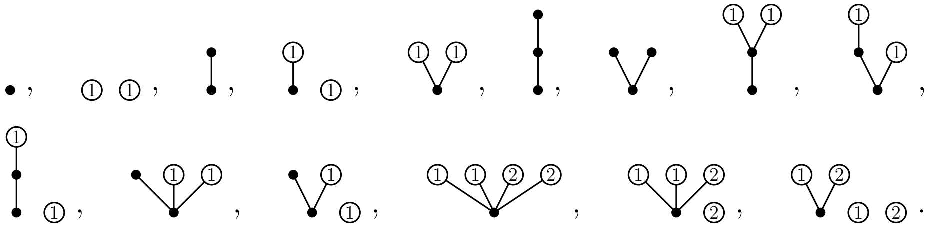
The size of a grafted forest is taken to be the sum of weights of vertices with black vertices having weight 1 and grafted vertices having weight 0.5. Grafted forests arise when we consider the overdamped Langevin equation (1.1.1) and the B-series [41] that are used to study it. The sets of grafted forests and trees are denoted by F_g and T_g, and the corresponding vector spaces are denoted by $\mathcal{F}_g$ and $\mathcal{T}_g$.

Exotic forests are grafted forests in which all grafted vertices are paired. Each pair of grafted vertices is called a *liana* and is assigned a natural number as its identifier.

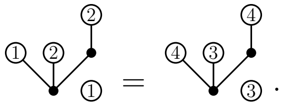
Exotic forests were first introduced in [41], where lianas were represented using a new type of edge. In this work, we adopt a different formalism, representing a liana by assigning the same natural number to its two paired grafted vertices. Exotic forests form a set EF and a vector space $\mathcal{EF}$. For example, some exotic trees are listed below



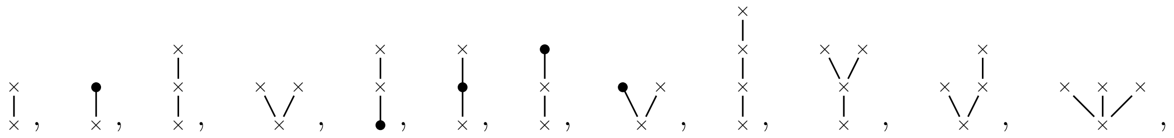
The pairing between two grafted vertices is denoted by assigning a natural number to the pair. The choice of the natural number doesn't matter as long as the natural numbers are distinct. An exotic forest is called connected if it cannot be written as a concatenation of non-trivial exotic subforests. For example, all connected exotic forests up to size 3 are:



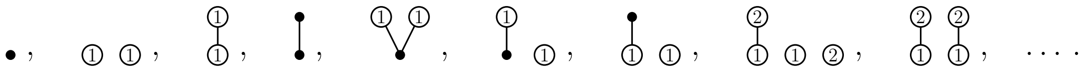
The following exotic forests are identical,



We extend the definition of the grafted tree by allowing internal vertices to be colored by ×. For example, we allow the following grafted trees up to size 2.5,



and the following connected exotic forests up to size 3,



Definition 1.4.2. An exotic forest is a decorated forests (π, α_e) where $D := \{\bullet\} \cup \mathbb{N}$ and $|\alpha_e^{-1}(k)| \in \{0, 2\}$ with two exotic forests $(\pi_1, \alpha_{e,1})$ and $(\pi_2, \alpha_{e,2})$ considered equivalent if $\pi_1 = \pi_2$ and there exists $\varphi \in \mathcal{S}_{\mathbb{N}}$ with $\varphi(\bullet) = \bullet$ such that $\alpha_{e,1} = \varphi \circ \alpha_{e,2}$.

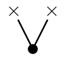
These grafted and exotic forests arise when we study the overdamped Langevin equation (1.1.4) with position-dependent diffusion matrix. We extend the homomorphism $\mathbb{F}$ to the space of decorated forests.

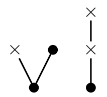
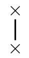
Definition 1.4.3. Let F_D be a set of decorated forests and let $\mathbb{F}(d)$ be a vector field corresponding to a decoration $d \in D$. Then, let $\tau = B_d^+(\pi)$, then,

$$\mathbb{F}(\tau) = \mathbb{F}(\pi)[\mathbb{F}(d)]\,,$$

where $\mathbb{F}(\pi) = \mathbb{F}(\gamma_1) \cdots \mathbb{F}(\gamma_n)$ for $\pi = \gamma_1 \cdots \gamma_n$. For grafted forests, $\mathbb{F}(\bullet) = F(X_0)$, $\mathbb{F}(\times) = \sigma\Sigma(X_0)\xi$.

For example, let all F and Σ be evaluated at X_0,

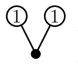
$\mathbb{F}($  $) = \mathbb{F}(\times \quad \times)[F] = \sigma^2 \sum_{j,k=1}^{d} (\Sigma\xi)^j (\Sigma\xi)^k (\partial_{j,k}F),$

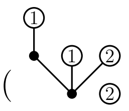
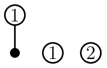
$\mathbb{F}($  $) = \sum_{i,j=1}^{d} \mathbb{F}(\times \quad \bullet)[F]^i \mathbb{F}($  $)[F]^j \partial_{i,j}$

$$= \sigma^3 \sum_{i,j,k,l,m,n=1}^{d} (\Sigma\xi)^k F^l (\partial_{k,l}F)^i (\Sigma\xi)^n (\partial_n \Sigma\xi)^m (\partial_m F)^j \partial_{i,j}$$

Definition 1.4.4. Let $\mathbb{F}$ be extended to exotic forests $\mathcal{EF}$ by $\mathbb{F}(\textcircled{\scriptsize k}) = \sigma \sum_{a_k=1}^{d} \Sigma_{a_k}$ for $k \in \mathbb{N}$.

For example,

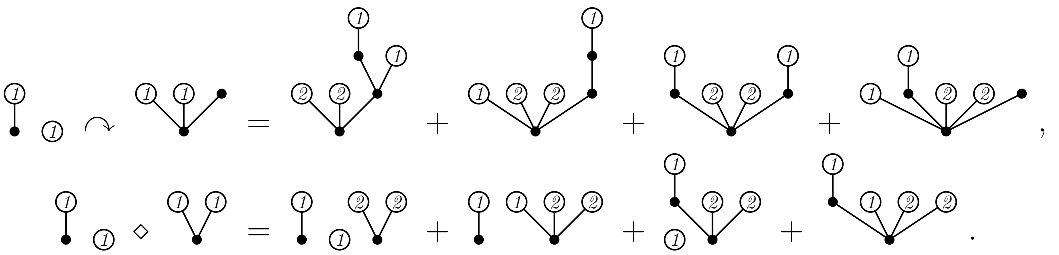
$\mathbb{F}($  $) = \mathbb{F}(\textcircled{1} \; \textcircled{1})[F] = \sigma^2 \sum_{j,k,a=1}^{d} \Sigma_{ja}\Sigma_{ka}(\partial_{j,k}F),$

$\mathbb{F}($  $) = \sigma \sum_{i,j,a_2=1}^{d} \mathbb{F}($  $)[F]^i \Sigma_{ja_2}\partial_{i,j}$

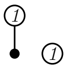
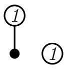
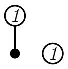
$$= \sigma^3 \sum_{i,j,k,l,m,n,a_1,a_2=1}^{d} \Sigma_{na_1}(\partial_n F)^k \Sigma_{la_1}\Sigma_{ma_2}(\partial_{k,l,m}F)^i \Sigma_{ja_2}\partial_{i,j}.$$

From the proposed examples, we can clearly see the advantage of using the exotic formalism to represent the corresponding vector fields and differential operators. It provides a concise notation and allows us to focus on the combinatorial aspects of our computations, greatly simplifying them. Proposition 1.3.4 is extended to exotic forests in [7, 41]. It allows us to study the weak convergence of integrators without leaving the framework of exotic forests. A detailed study of the grafting and Grossman-Larson products as well as deshuffle coproduct on exotic forests can be found in Chapter 3.

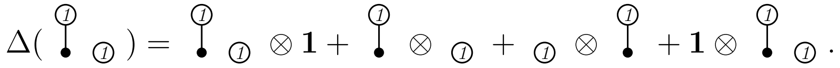
Example 1.4.5. *Several examples of the grafting and Grossman-Larson products on exotic forests are found below*



The deshuffle coproduct over exotic forests cannot break a connected component, for example,

$\Delta($  $) =$  $\otimes \mathbf{1} + \mathbf{1} \otimes$  $,$

while in the case of classical decorated forests, we would have



We use exotic forests to express the generator $\mathcal{L}$ and the weak Taylor expansion of the exact solution as

$$\mathcal{L}\phi = \mathbb{F}(\bullet + \frac{1}{2} ① ①)[\phi], \quad \mathbb{E}[\phi(X(h))] = \mathbb{F}(\exp^{\diamond}(\bullet + \frac{1}{2} ① ①))[\phi]\,.$$

This allows us to express the exact solution $\mathbb{E}[\phi(X(h))]$ as an S-series over exotic forests, $S(\alpha)[\phi]$, for a suitably defined functional $\alpha : EF \to \mathbb{R}$. If the weak Taylor expansion of an integrator can be expressed using an S-series $S(a)$, then the weak order can be verified by checking that $a(\pi) = \alpha(\pi)$ for all π of size at most q, where q is the desired order of accuracy.

In a similar fashion, we use exotic forests to express order conditions for the invariant measure sampling. To simplify notation, let $\langle \pi \rangle$ denote the integral with respect to the invariant measure over $\mathbb{R}^d$ of the differential corresponding to π applied to an arbitrary test function ϕ, that is,

$$\langle \pi \rangle := \int_{\mathbb{R}^d} \mathbb{F}(\pi)[\phi](x)\rho_\infty(x)\,dx\,.$$

Then, an integrator with a weak Taylor expansion $S(a)$ is of order p with respect to the invariant measure if

$$\langle \sum_{|\pi|\leq p} \frac{a(\pi)}{\sigma(\pi)}\pi \rangle = 0\,.$$

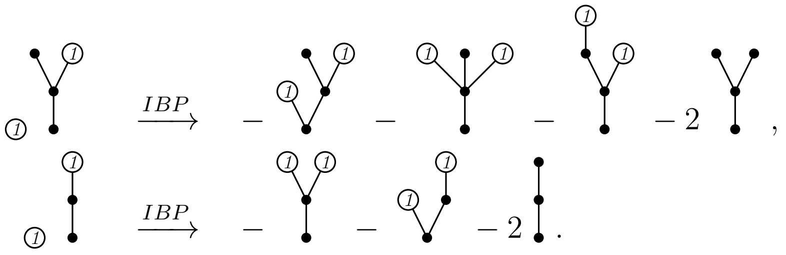
The integration by parts (IBP) technique, as described in Chapter 3, induces a transformation on exotic forests that preserves the value of $\langle - \rangle$ [41, 79]. Originally introduced in the context of SDE (1.1.1), this technique is also shown to be applicable in the context of (1.1.4).

Theorem 1.4.6 (IBP). *Let $\pi \in EF$ be an exotic forest and choose a grafted root v paired to a grafted leaf u. Then,*

$$\langle \pi \rangle = -\langle \sum_{\substack{w \in V(\pi) \\ w \notin \{v,u\}}} \pi^{v \to w} + 2\pi^{\bullet} \rangle\,,$$

where $V(\pi)$ is the set of vertices of π, $\pi^{v\to w}$ is the exotic forest π in which v is connected to w, and $\pi^{\bullet}$ is the exotic forest π in which v is removed and u is replaced by a new black vertex.

Example 1.4.7. *Some examples of the IBP transformation are given below*



We use IBP in Chapter 2 to derive a new order-two method for invariant measure sampling and in Chapter 3.1 to introduce a formal algorithm for the generation of order conditions for invariant measure sampling. The generated order conditions for order p have the form $\omega(\pi) = 0$ for all $\pi \in EF$ of size $|\pi| \leq p$ with $\omega \in \mathcal{EF}^*$ being called the *order condition map*. The algebraic properties of the algorithm enable us to obtain Theorem 3.1.8 which is presented and proved in Section 3.1.

Theorem 3.1.8. *Let $\cdot$ denote the concatenation product and let ω be the order condition map for a numerical method that can be expanded as a B-series over grafted trees, then,*

$$\omega(\pi_1 \cdot \pi_2) = \omega(\pi_1)\omega(\pi_2), \quad \textit{for } \pi_1, \pi_2 \in EF.$$

Theorem 3.1.8 allows us to decrease the number of order conditions with respect to the invariant measure for a class of numerical methods that includes stochastic Runge-Kutta-type methods. If an exotic forest π can be written as $\pi = \pi_1 \cdot \pi_2$, then the order condition $\omega(\pi) = 0$ is automatically satisfied if $\omega(\pi_1) = 0$ is satisfied. For example, Theorem 3.1.8 implies the following relations between the order conditions:

$$\omega(\,\bullet\ \ \bullet\,) = \omega(\,\bullet\,)^2, \qquad \omega(\,\bullet\ \ \bullet\ \ \bullet\,) = \omega(\,\bullet\,)^3,$$
$$\omega(\,\overset{\bullet}{\underset{\bullet}{|}}\ \ \bullet\,) = \omega(\,\overset{\bullet}{\underset{\bullet}{|}}\,)\omega(\,\bullet\,), \quad \omega(\,\overset{\textcircled{1}\,\textcircled{1}}{\underset{\bullet}{\vee}}\ \ \bullet\,) = \omega(\,\overset{\textcircled{1}\,\textcircled{1}}{\underset{\bullet}{\vee}}\,)\omega(\,\bullet\,),$$

which decreases the number of order conditions, in particular, for order 3 from 13 to 9. The values of $\omega(\,\overset{\bullet}{\underset{\bullet}{|}}\ \ \bullet\,), \omega(\,\overset{\bullet}{\underset{\bullet}{|}}\,)$, and $\omega(\,\bullet\,)$ for stochastic Runge-Kutta methods with coefficients b_i, a_{ij}, d_i with $i, j = 1, \ldots, s$ are presented below,

$$\omega(\,\overset{\bullet}{\underset{\bullet}{|}}\,) = \sum_{i,j=1}^{s} b_i a_{ij} - \frac{1}{2} + \sum_{i=1}^{s} b_i - 2\sum_{i=1}^{s} b_i d_i, \qquad \omega(\,\bullet\,) = \sum_{i=1}^{s} b_i - 1,$$
$$\omega(\,\overset{\bullet}{\underset{\bullet}{|}}\ \ \bullet\,) = -2\sum_{i,j=1}^{s} b_i d_i b_j - \frac{3}{2}\sum_{i=1}^{s} b_i + \sum_{i,j=1}^{s} b_i b_j + \sum_{i,j,k=1}^{s} b_i a_{ij} b_k + 2\sum_{i=1}^{s} b_i d_i + \frac{1}{2} - \sum_{i,j=1}^{s} b_i a_{ij}.$$

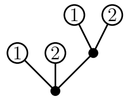
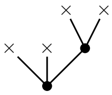
The list of values of ω for all exotic trees up to size 3 can be found in Table .0.2 in the Appendix. This property was first observed for order 3 by manual computation in [41]. In this paper, we prove the property for arbitrary high order.

Relation between grafted and exotic forests

We relate grafted and exotic forests by examining the relation between the corresponding sets of decorations. We note that the differential operators and vector fields corresponding to exotic forests are obtained as the expectation of the differential operators and vector fields corresponding to grafted forests. This relationship is then translated into a relation between grafted and exotic forests.

Definition 1.4.8. Let $\alpha : V(\pi) \to D$ and $\hat{\alpha} : V(\pi) \to \hat{D}$ be two decorations of a forest $\pi \in F$. Decoration α is said to be *finer* than $\hat{\alpha}$ if there exists a surjective map $\Phi : D \to \hat{D}$ such that $\hat{\alpha} = \Phi \circ \alpha$.

For example, let $\Phi : D_e \to D_g$ for $D_e = \{\bullet\} \sqcup \mathbb{N}$ and $D_g = \{\bullet, \times\}$ be the map defined as $\Phi(\bullet) := \bullet$ and $\Phi(k) = \times$ for all $k \in \mathbb{N}$, then, it induces a map $\Phi_\pi : (\pi, \alpha_e) \to (\pi, \alpha_g)$, e.g.

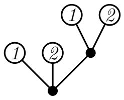
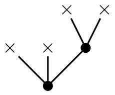
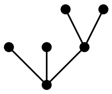
$\Phi_\pi($  $) =$  .

We note that Φ_π is well-defined on the equivalence classes (π, α_e) that are used to define the exotic forests. We say that the decoration of the exotic forests is finer than the decoration of the grafted forests.

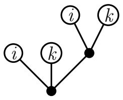
Definition 1.4.9. Let $p(\pi, \alpha, \hat{\alpha})$, with α being finer than $\hat{\alpha}$, denote the number of decorations $\tilde{\alpha}$ such that $(\pi, \tilde{\alpha}) \cong (\pi, \alpha)$ and $\hat{\alpha} = \Phi \circ \tilde{\alpha}$ where Φ is the map such that $\hat{\alpha} = \Phi \circ \alpha$.

If $(\pi, \alpha_e) \in EF$ is an exotic forest, then $p(\pi, \alpha_e, \alpha_g)$ is the number of ways to pair grafted vertices of $(\pi, \alpha_g) \in F_g$ to obtain a forest isomorphic to (π, α_e).

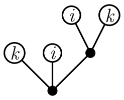
Example 1.4.10. *Let us consider*

$(\pi, \alpha_e) =$  , $(\pi, \alpha_g) =$  , *with* $\pi =$  ,

and find the value of $p(\pi, \alpha_e, \alpha_g)$. *Let us list the elements of the equivalence class* (π, α_e)*:*

$(\pi, \alpha_e) = \{$  $: \ i, k \in \mathbb{N}\}$,

and notice that the only choice for $\tilde{\alpha}_e \neq \alpha_e$ *such that* $(\pi, \tilde{\alpha}_e) \cong (\pi, \alpha_e)$ *and* $\Phi \circ \tilde{\alpha}_e = \alpha_g$ *is*

$(\pi, \tilde{\alpha}_e) = \{$  $: \ i, k \in \mathbb{N}\}$,

Therefore, $p(\pi, \alpha_e, \alpha_g) = 2$.

Proposition 1.4.11. *[7] Let* $\pi \in F$ *be a forest and let* $\alpha : V(\pi) \to D$ *be finer than* $\hat{\alpha} : V(\pi) \to \hat{D}$*, then*

$$p(\pi, \alpha, \hat{\alpha}) = \frac{\sigma(\pi, \hat{\alpha})}{\sigma(\pi, \alpha)}.$$

The differential operators corresponding to grafted and exotic forests are related through expectation, which forms the pairings between the grafted vertices via the Isserlis theorem, as they correspond to Gaussian random variables. For more details, see [41].

Theorem 1.4.12. *[41, 7] Let* $(\pi, \alpha_g) \in F_g$ *be a grafted forest with an even number of grafted vertices. Then, the expectation of* $\mathbb{F}(\pi, \alpha_g)$ *is given by*

$$\mathbb{E}[\mathbb{F}(\pi, \alpha_g)] = \sum_{\alpha_e \in \Phi^{-1}(\alpha_g)} \frac{\sigma(\pi, \alpha_g)}{\sigma(\pi, \alpha_e)} \mathbb{F}(\pi, \alpha_e),$$

where the sum is over all decorations α_e *defined by Definition 1.4.2 such that* $\Phi \circ \alpha_e = \alpha_g$ *where* $\Phi(\bullet) = \bullet$ *and* $\Phi(k) = \times$ *for all* $k \in \mathbb{N}$.

Proof. We use the definitions of grafted and exotic forests, as well as Proposition 1.4.11. □

Theorem 1.4.12 is used to express the relation between the S-series over grafted and exotic forests in Chapter 3.

1.5 Main contributions

The main contributions of this thesis are as follows:

- Construction of an efficient second-order integrator for sampling the invariant measure of Langevin dynamics with a position-dependent diffusion matrix (Chapter 2).
- Introduction of exotic aromatic S-series, formulation of a formal algorithm for generating order conditions for invariant measure sampling, and proof that the order condition map ω is a character, which permits the reduction of the number of order conditions (Section 3.1).
- Study of the Grossman-Larson Hopf algebroid of decorated aromatic forests, including the pre-Hopf algebroid and the composition law (Section 3.2).
- Introduction of clumped forests, analysis of the multi-pre-Lie insertion algebra of decorated aromatic trees, and description of the substitution law (Section 3.3).
- Extension of these results to the exotic aromatic context and introduction of backward error analysis and modified equation techniques in the context of invariant measure sampling (Section 3.4).
- Development of a Haskell package to automate computations related to algebras of forest-like graphs (Chapter 4).

Chapter 2

Construction of a new method for Langevin dynamics

In this chapter, we present a framework for constructing new numerical integrators of a desired order. The key ingredient in this methodology is the use of exotic forests to represent the differential operators that appear in the Taylor expansion of the exact solution and the numerical integrator.

We use our framework to introduce a generalization of the method (1.2.2) of order 2 with respect to the invariant measure for the case of a position-dependent matrix Σ in (1.1.4). An essential feature of the new method is that it requires only one evaluation of the drift per timestep. The procedure for constructing the new method is as follows:

1. Compute the differential operator corresponding to h^2 of the exact solution of the SDE (1.1.4) using the tree formalism.

2. Simplify the expression using integration by parts to eliminate terms requiring multiple evaluations of the drift, without losing the order with respect to invariant measure sampling.

3. Construct a post-processed Runge-Kutta-type method whose Taylor expansion matches the simplified expression obtained in the second step.

This chapter follows closely the paper [12], which has been submitted for publication and was written in collaboration with Benedict Leimkuhler, Dominic Phillips, and Gilles Vilmart. The new method is defined as:

$$\begin{aligned} X_{n+1} &= X_n + hF(\overline{X}_n) + \hat{\Phi}_h^{\Sigma}(X_n + \frac{1}{4}hF(\overline{X}_{n-1})), \\ \overline{X}_n &= X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n, \quad \text{with } \overline{X}_{-1} = X_0, \end{aligned} \tag{2.0.1}$$

where $\Phi_h^{\Sigma}(X_n) = X_n + \hat{\Phi}_h^{\Sigma}(X_n)$ is an integrator of weak order 2 applied to the SDE problem with noise only,

$$dX = \sigma\Sigma(X)dW, \tag{2.0.2}$$

where $\Phi^{\Sigma}(X_0) = X_0 + \sqrt{h}\sigma\Sigma(X_0)\xi_n + O(h)$. The output of the method is given by $\overline{X}_{n+1}$ and it is referred to as Second-Order Postprocessed method for Variable Diffusion (PVD-2). PVD-2 (2.0.1) has one evaluation of F and the number of evaluations of Σ

depends on our choice of Φ_h^Σ. We remark that the new method (2.0.1) uses a similar post-processor compared to (1.2.2),

$$\overline{X}_n = \Psi_h(X_n) = X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n, \tag{2.0.3}$$

and that it becomes equivalent to (1.2.2) for the additive noise case $\Sigma(x) = I$. There are different natural choices for the noise integrator Φ_h^Σ involved in the new scheme (2.0.1). Possible weak second order noise integrators for (2.0.2) include:

(Method MT2) The tensor Σ is evaluated 5 times if we choose Φ_h^Σ to be the method from [4, eq. (3.7)], a derivative-free variant of the so-called Milstein-Talay method [53, 27, p. 103, eq. (2.18)] where written in the particular case of a null drift function:

$$\begin{aligned} X_1 = X_0 &+ \frac{1}{2}\sum_{a=1}^{d}\Big(\sigma\Sigma_a\big(X_0 + h\sigma\Sigma(X_0)J_a\big) - \sigma\Sigma_a\big(X_0 - h\sigma\Sigma(X_0)J_a\big)\Big) \\ &+ \frac{\sigma\sqrt{h}}{2}\Big(\sigma\Sigma\big(X_0 + \sqrt{\frac{h}{2}}\sigma\Sigma(X_0)\chi\big) + \sigma\Sigma\big(X_0 - \sqrt{\frac{h}{2}}\sigma\Sigma(X_0)\chi\big)\Big)\xi_n, \end{aligned}$$

where $J_a = (J_{a,b})_{b=1}^d, \chi = (\chi_b)_{b=1}^d$, and for $a, b = 1, \dots, d$ we have

$$\mathbb{P}(\chi_b = \pm 1) = \frac{1}{2},$$

$$J_{a,b} = \begin{cases} (\xi_{n,b}\xi_{n,b} - 1)/2, & \text{if } a = b, \\ (\xi_{n,a}\xi_{n,b} - \chi_a)/2, & \text{if } a > b, \\ (\xi_{n,a}\xi_{n,b} + \chi_b)/2, & \text{if } a < b. \end{cases}$$

(Method W2Ito1) The tensor Σ is evaluated 3 times if we take Φ_h^Σ to be the method introduced in [76, table 2] with a null drift function:

$$\begin{aligned} X_1 &= X_0 + \sqrt{h}\sum_{a=1}^{d}\big(-\sigma\Sigma_a(X_0) + \sigma\Sigma_a(K_1^{(a)}) + \sigma\Sigma_a(K_2^{(a)})\big)\xi_{n,a} \\ &\qquad + 2\sqrt{h}\sum_{a=1}^{d}\big(\sigma\Sigma_a(X_0) - \sigma\Sigma_a(K_2^{(a)})\big)\hat{J}_{a,a}, \\ K_1^{(a)} &= X_0 + \frac{\sqrt{h}}{2}\sigma\Sigma_a(X_0)\hat{\chi}_1 + \sqrt{h}\sum_{\substack{b=1\\ b\neq a}}^{d}\sigma\Sigma_b(X_0)\hat{J}_{a,b}, \\ K_2^{(a)} &= X_0 - \frac{\sqrt{h}}{2}\sigma\Sigma_a(X_0)\hat{\chi}_1. \end{aligned}$$

with $\mathbb{P}(\hat{\chi}_i = \pm 1) = \frac{1}{2}$ for $i = 1, 2$ and

$$\hat{J}_{a,b} = \begin{cases} \hat{\chi}_1(\xi_{n,a}^2 - 1)/2, & \text{if } a = b, \\ \xi_{n,b}(1 + \hat{\chi}_2)/2, & \text{if } a > b, \\ \xi_{n,b}(1 - \hat{\chi}_2)/2, & \text{if } a < b. \end{cases}$$

We emphasis that the above methods MT2 and W2Ito1 are Runge-Kutta type methods with noise increments in the internal stages, an idea first introduced in [68] to obtain a number of diffusion tensor evaluations that is independent of the dimension of the noise. We denote versions of the PVD-2 method that use these noise integrators as PVD-2[MT2] and PVD-2[W2Ito1] respectively.

Theorem 2.0.1. *The method presented in* (2.0.1) *is of order* 2 *with respect to the invariant measure for the SDE* (1.1.4)*.*

The proof of Theorem 2.0.1 is provided in Section 2.1. Section 2.2 examines the stability of the new method and introduces several modifications that enhance its stability. Experimental results confirming convergence are presented in Section 2.3.

2.1 Convergence analysis for invariant measure sampling

We recall that the weak Taylor expansion of the solution of (1.1.4) has the form

$$\mathbb{E}[\phi(X(h))|X(0)=X_0] = \phi(X_0) + h(\mathcal{L}\phi)(X_0) + h^2\frac{\mathcal{L}^2\phi}{2!}(X_0) + \cdots ,$$

where the generator of the SDE is given by $\mathcal{L}\phi := F\cdot\nabla\phi + \frac{\sigma^2}{2}\sum_{a=1}^{d}\phi''(\Sigma_a,\Sigma_a)$.

2.1.1 Integration by parts

The order conditions for the invariant measure of a post-processed integrator, as stated in Theorems 1.2.5 and 1.2.6, involve integrals of differential operators applied to test functions ϕ, we use integration by parts and Lemma 2.1.1 to manipulate the expressions. The integration by parts technique is demonstrated in the context of the SDE (1.1.4), confirming that Theorem 1.4.6 remains applicable.

Lemma 2.1.1. *We have the following identities:*

1. $\mathrm{div}(\Sigma^2) = \sum_{a=1}^{d}\Sigma_a\,\mathrm{div}(\Sigma_a) + \sum_{a=1}^{d}\Sigma_a'\Sigma_a$,
2. $\Sigma^2 f = \sum_{a=1}^{d}\Sigma_a(\Sigma_a\cdot f)$.

We apply Lemma 2.1.1 in the following example where we demonstrate integration by parts and show the benefits of using the tree formalism.

Example 2.1.2. *Let us consider a case with* Σ *assumed constant,*

$$\sum_{i,j,k,a=1}^{d}\langle\Sigma_{ia}\Sigma_{ja}F^k\partial_{i,j,k}\phi\rangle = \sum_{i,j,k,a=1}^{d}\int_{\mathbb{R}^d}\Sigma_{ia}\Sigma_{ja}F^k\partial_{i,j,k}\phi\rho_\infty dx \tag{A}$$

apply integration by parts

$$\begin{aligned}(A) = &-\sum_{i,j,k,a=1}^{d}\int_{\mathbb{R}^d}\Sigma_{ia}\Sigma_{ja}(\partial_i F^k)(\partial_{j,k}\phi)\rho_\infty dx\\ &-\sum_{i,j,k,a=1}^{d}\int_{\mathbb{R}^d}\Sigma_{ia}\Sigma_{ja}F^k(\partial_{j,k}\phi)(\partial_i\rho_\infty)dx\end{aligned}$$

we use $\partial_i\rho_\infty = \frac{2}{\sigma^2}f^i\rho_\infty$ *and rewrite the expression without the integral notation*

$$(A) = -\sum_{j,k,a=1}^{d}\langle\sum_{i=1}^{d}\Sigma_{ia}\Sigma_{ja}(\partial_i F^k)(\partial_{j,k}\phi) + \frac{2}{\sigma^2}(\Sigma_a\cdot f)\Sigma_{ja}F^k(\partial_{j,k}\phi)\rangle$$

applying Lemma 2.1.1 and the definition of F we get

$$(A) = -\sum_{j,k,a=1}^{d} \langle \frac{2}{\sigma^2} F^j F^k (\partial_{j,k}\phi) + \sum_{i=1}^{d} \Sigma_{ia}\Sigma_{ja}(\partial_i F^k)(\partial_{j,k}\phi)\rangle \,.$$

Next, we consider the same case with Σ being non-constant,

$$\sum_{i,j,k,a=1}^{d} \langle \Sigma_{ia}\Sigma_{ja} F^k (\partial_{i,j,k}\phi)\rangle = \sum_{i,j,k,a=1}^{d} \int_{\mathbb{R}^d} \Sigma_{ia}\Sigma_{ja} F^k (\partial_{i,j,k}\phi)\rho_\infty dx \tag{B}$$

apply integration by parts

$$\begin{aligned}(B) = &-\sum_{i,j,k,a=1}^{d} \int_{\mathbb{R}^d} (\partial_i\Sigma_{ia})\Sigma_{ja} F^k (\partial_{j,k}\phi)\rho_\infty dx - \sum_{i,j,k,a=1}^{d} \int_{\mathbb{R}^d} \Sigma_{ia}(\partial_i\Sigma_{ja}) F^k (\partial_{j,k}\phi)\rho_\infty dx \\ &- \sum_{i,j,k,a=1}^{d} \int_{\mathbb{R}^d} \Sigma_{ia}\Sigma_{ja}(\partial_i F^k)(\partial_{j,k}\phi)\rho_\infty dx - \sum_{i,j,k,a=1}^{d} \int_{\mathbb{R}^d} \Sigma_{ia}\Sigma_{ja} F^k (\partial_{j,k}\phi)(\partial_i\rho_\infty) dx\end{aligned}$$

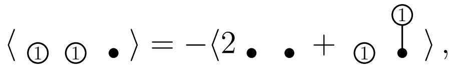
we use $\partial_i\rho_\infty = \frac{2}{\sigma^2}F^i\rho_\infty$ and rewrite the expression without the integral notation

$$\begin{aligned}(B) = &-\sum_{j,k,a=1}^{d} \langle (\Sigma_a \operatorname{div}(\Sigma_a) + \Sigma_a'\Sigma_a)^j F^k (\partial_{j,k}\phi) \\ &+ \sum_{i=1}^{d} \Sigma_{ia}\Sigma_{ja}(\partial_i F^k)(\partial_{j,k}\phi) + \frac{2}{\sigma^2}(\Sigma_a \cdot F)\Sigma_{ja} F^k (\partial_{j,k}\phi)\rangle\end{aligned}$$

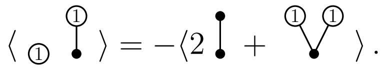
applying Lemma 2.1.1 and the definition of F we get

$$(B) = -\sum_{j,k,a=1}^{d} \langle \frac{2}{\sigma^2} F^j F^k (\partial_{j,k}\phi) + \sum_{i=1}^{d} \Sigma_{ia}\Sigma_{ja}(\partial_i F^k)(\partial_{j,k}\phi)\rangle \,.$$

Using the tree formalism introduced in Chapter 1, we can write this example as



which agrees with Theorem 1.4.6. An analogous computation can be performed to show that



(2.1.1)

Equation (2.1.1) is used in the proof of Theorem 2.0.1.

2.1.2 Proof of convergence

Let us start by recalling the statement and the setting of Theorem 2.0.1. We consider the following integrator,

$$\begin{aligned} X_{n+1} &= X_n + hF(\overline{X}_n) + \hat{\Phi}_h^\Sigma(X_n + \frac{1}{4}hF(\overline{X}_{n-1}))\,, \\ \overline{X}_n &= X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n\,, \quad \text{with } \overline{X}_{-1} = X_0\,, \end{aligned} \tag{2.1.2}$$

where $\Phi_h^\Sigma(X_n) = X_n + \hat{\Phi}_h^\Sigma(X_n) = X_n + \sqrt{h}\sigma\Sigma(X_n)\xi_n + \mathcal{O}(h)$ is an integrator of weak order 2 applied to the problem $dX = \sigma\Sigma(X)dW$.

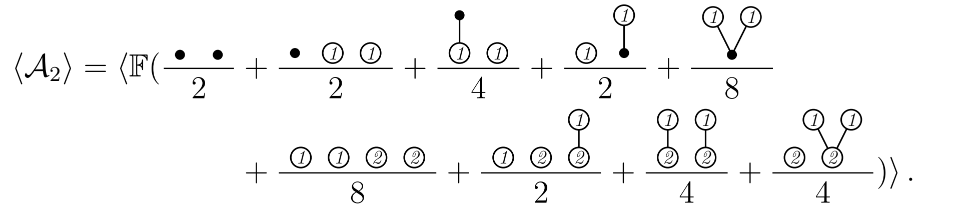
Theorem 2.0.1. *The integrator of the form (2.1.2) is of order* 2 *with respect to the invariant measure.*

We consider a modification of integrator (2.1.2) which is written as an integrator $X_{n+1} = \Phi_h(X_n)$ and a post-processor $\overline{X}_n = \Psi_h(X_n)$ with

$$\begin{aligned}\Phi_h(X_n) &= X_n + hF(Y) + \hat{\Phi}_h^{\Sigma}(X_n + \frac{1}{4}hF(X_n))\,,\\ Y &= X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n\,,\\ \Psi_h(X_n) &= X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n\,, \qquad (2.1.3)\end{aligned}$$

where $F(\overline{X}_{n-1})$ of (2.1.2) is replaced by $F(X_n)$. We note that the integrator (2.1.3) requires 2 evaluations of F per step.

Lemma 2.1.3. *Given a post-processor of the form (2.1.3) and an integrator* Φ_h *which admits weak Taylor expansion with* $\mathcal{A}_1 = \mathcal{L}$ *has order* 2 *with respect to the invariant measure if*

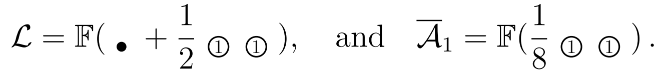


(2.1.4)

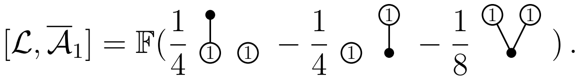
Proof. By Theorem 1.2.6, an integrator Φ_h with $\mathcal{A}_1 = \mathcal{L}$ has order 2 with respect to the invariant measure if $\mathcal{A}_2$ satisfies

$$\langle \mathcal{A}_2 \rangle = \langle \frac{\mathcal{L}^2}{2} - [\mathcal{L}, \overline{\mathcal{A}}_1] \rangle\,, \qquad (2.1.5)$$

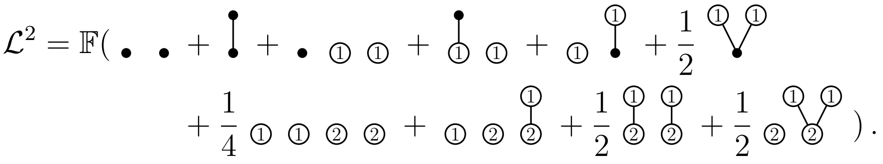
where we note that $\langle \frac{\mathcal{L}^2}{2} \rangle = 0$. We use the tree formalism to express the condition (2.1.5) explicitely. We start by noting that



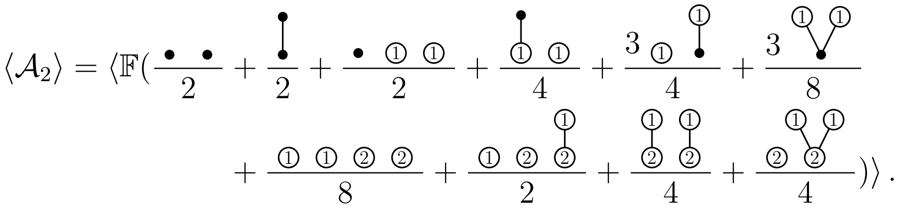
This implies, using Proposition 1.3.4, that



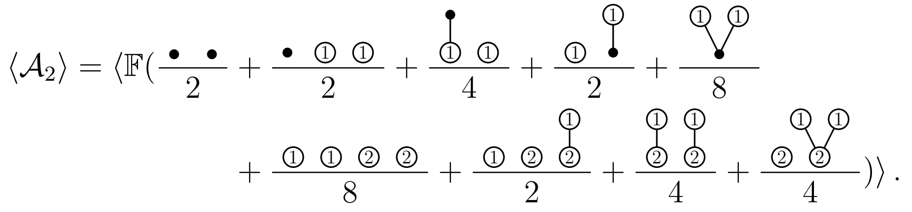
Using Proposition 1.3.4, we express $\mathcal{L}^2$ as



This gives us the following condition on $\mathcal{A}_2$:



Using (2.1.1) we finally get

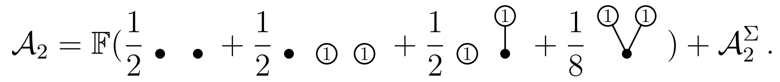


This finishes the proof. □

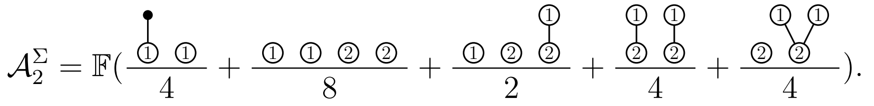
We use Lemma 2.1.3 to prove Proposition 2.1.4.

Proposition 2.1.4. *The post-processed integrator $\overline{X}_n = (\Psi_h \circ \Phi_h^n)(X_0)$ of (2.1.3) is of order 2 with respect to the invariant measure.*

Proof. The post-processed integrator $\overline{X}_n = (\Psi_h \circ \Phi_h^n)(X_0)$ of (2.1.3) has $\mathcal{A}_1 = \mathcal{L}$. To see that it is of order 2 with respect to the invariant measure, we use Lemma 2.1.3, that is, we check that $\mathcal{A}_2$ satisfies (2.1.4). The differential operators appearing in the weak Taylor expansion of Φ_h^Σ are denoted by $\mathcal{A}_j^\Sigma$ where $j \in \mathbb{N}$ with $\mathcal{A}_1^\Sigma = \mathbb{F}(\frac{1}{2}$ ① ① $)$. Therefore, the differential operator $\mathcal{A}_2$ has the form



Since Φ_h^Σ is weak order 2, $\mathcal{A}_2^\Sigma$ of $\Phi_h^\Sigma(X_n + \frac{1}{4}hF(X_n))$ is



Therefore, the condition (2.1.4) is satisfied and the method has order 2 with respect to the invariant measure. □

We are now ready to prove Theorem 2.0.1 and show that the post-processed integrator of (2.1.2) has order 2 with respect to the invariant measure.

Proof. **of Theorem 2.0.1** We recall that the only difference between the integrators described by (2.1.2) and (2.1.3) is the replacement of $F(\overline{X}_{n-1})$ by $F(X_n)$ which simplifies the analysis. Let us now show that the differential operators $\mathcal{A}_2$ of (2.1.2) and (2.1.3) are identical and, therefore, the integrator (2.1.2) is of order 2 with respect to the invariant measure.

We have the following identity

$$\begin{aligned}\overline{X}_{n-1} &= X_{n-1} + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_{n-1})\xi_{n-1} \\ &= X_n - hF(\overline{X}_{n-1}) - \hat{\Phi}_h^\Sigma(X_{n-1} + \frac{1}{4}hF(\overline{X}_{n-2})) + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_{n-1})\xi_{n-1} \\ &= X_n - \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_{n-1} + \mathcal{O}(h).\end{aligned}$$

Therefore, the $\hat{\Phi}_h^\Sigma$ term of (2.1.2) has the following form

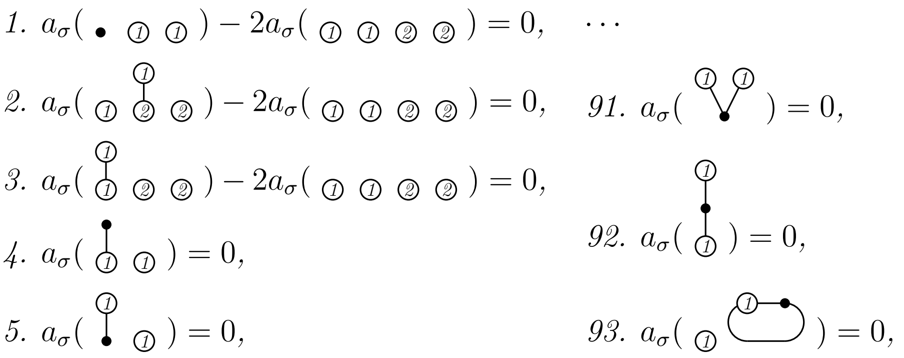
$$\hat{\Phi}_h^\Sigma\big(X_n + \frac{1}{4}hF(\overline{X}_{n-1})\big) = \hat{\Phi}_h^\Sigma\big(X_n + \frac{1}{4}hF(X_n) - \frac{1}{8}h\sqrt{h}\sigma F'(X_n)\Sigma(X_n)\xi_{n-1} + \mathcal{O}(h^2)\big)\,.$$

We see that the Taylor expansion of the $\hat{\Phi}_h^{\Sigma}$ term of (2.1.2) differs from the Taylor expansion of the $\hat{\Phi}_h^{\Sigma}$ term of (2.1.3) by

$$-\frac{1}{8}h^2\sigma^2(\Sigma\xi_n)'F'\Sigma\xi_{n-1} + \mathcal{O}(h^{2.5}),$$

which has expectation $\mathcal{O}(h^3)$ due to the fact that each term corresponding to $h^{2.5}$ has an odd number of gaussians ξ_n or ξ_{n-1} with mean 0. Therefore, the differential operators $\mathcal{A}_2$ of (2.1.2) and (2.1.3) are identical. □

Remark 2.1.5. *A direct approach, involving the computation of order conditions for second order with respect to the invariant measure, followed by the solution of the resulting system, proved to be too challenging to perform manually. This involved handling 93 order conditions that needed to be satisfied. A subset of these conditions is listed below:*


1. $a_\sigma($ $) - 2a_\sigma($ $) = 0,$ ···
2. $a_\sigma($ $) - 2a_\sigma($ $) = 0,$
3. $a_\sigma($ $) - 2a_\sigma($ $) = 0,$
4. $a_\sigma($ $) = 0,$
5. $a_\sigma($ $) = 0,$
91. $a_\sigma($ $) = 0,$
92. $a_\sigma($ $) = 0,$
93. $a_\sigma($ $) = 0,$
1
2

where $a_\sigma(\pi)$ denotes the Runge-Kutta coefficient corresponding to the forest π and divided by its symmetry.

2.2 Mean-square stability analysis

We observe that the next step, X_{n+1}, in the method (2.0.1) proposed here is dependent on both X_n and $\overline{X}_{n-1}$. To analyze the stability of this method effectively, we express it in a partitioned form $X^P_{n+1} = \Phi^P_h(X^P_n)$ where $X^P_n = (X_n^T, \overline{X}_{n-1}^T)^T$:

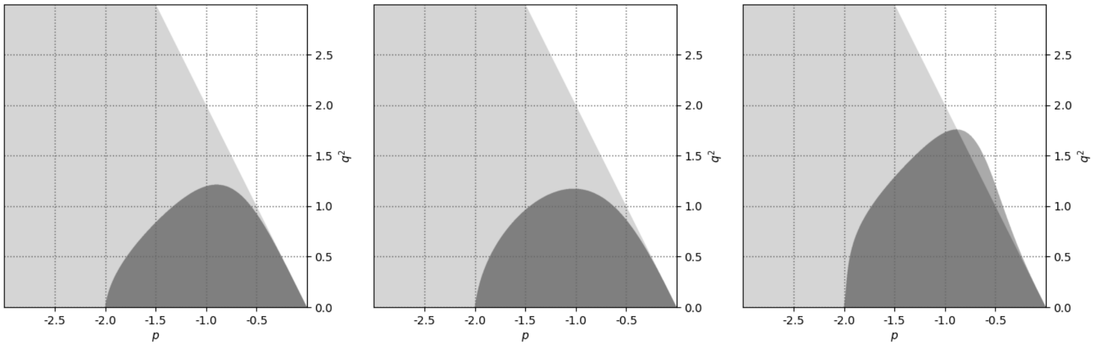
$$\begin{pmatrix} X_{n+1} \\ \overline{X}_n \end{pmatrix} = \Phi_h^P \begin{pmatrix} X_n \\ \overline{X}_{n-1} \end{pmatrix} = \begin{pmatrix} X_n + hF(\overline{X}_n) + \hat{\Phi}_h^{\Sigma}\left(X_n + \frac{1}{4}hF(\overline{X}_{n-1})\right) \\ X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n \end{pmatrix}. \quad (2.2.1)$$

2.2.1 Stability domain for mean-square stiff problems

We consider the following test problem in dimension $d = 1$, which is introduced in [71] and widely used in the literature [1, 15, 35, 77] for studying the mean-square stability of integrators applied to stiff problems::

$$dX(t) = \lambda X(t)dt + \mu X(t)dW(t), \quad X(0) = 1, \quad (2.2.2)$$

where λ and μ are fixed complex parameters. After applying the new method to the test problem, we obtain the stability matrix $R(p, q, \xi_n)$ of the following form, with $p = \lambda h$ and $q = \mu\sqrt{h}$,



(a) New method (2.0.1) (b) Modification 1 (2.2.6) (c) Modification 2 (2.2.7)

Figure 2.2.1: Mean-square stability domains of the PVD-2 method (2.0.1) and modifications (2.2.6) and (2.2.7), for which $\mathbb{E}[\overline{X}_n^2] \to 0$ for the scalar test problem (2.2.2) in the (p, q^2)–plane where $p = \lambda h, q = \mu\sqrt{h}$.

$$\begin{pmatrix} X_{n+1} \\ \overline{X}_n \end{pmatrix} = \begin{pmatrix} 1 + p + \frac{1}{2}pq\xi_n + \hat{R}^{\Sigma}(p,q,\xi_n) & \frac{1}{4}p\hat{R}^{\Sigma}(p,q,\xi_n) \\ 1 + \frac{1}{2}q\xi_n & 0 \end{pmatrix} \begin{pmatrix} X_n \\ \overline{X}_{n-1} \end{pmatrix}, \tag{2.2.3}$$

where $R^{\Sigma}(p,q,\xi_n)$ is the stability function of the noise integrator and $\hat{R}^{\Sigma} = R^{\Sigma} - 1$. Following the ideas from Saito-Mitsui [71], we consider $\mathbb{E}[X_{n+1}^P {X_{n+1}^P}^T]$ and obtain the following equation with $R = R(p,q,\xi_n)$ being the stability matrix from (2.2.3),

$$\begin{pmatrix} \mathbb{E}[X_{n+1}^2] \\ \mathbb{E}[\overline{X}_n^2] \\ \mathbb{E}[X_{n+1}\overline{X}_n] \end{pmatrix} = \begin{pmatrix} \mathbb{E}[R_{11}^2] & \mathbb{E}[R_{12}^2] & 2\mathbb{E}[R_{11}R_{12}] \\ \mathbb{E}[R_{21}^2] & 0 & 0 \\ \mathbb{E}[R_{21}R_{11}] & 0 & \mathbb{E}[R_{21}R_{12}] \end{pmatrix} \begin{pmatrix} \mathbb{E}[X_n^2] \\ \mathbb{E}[\overline{X}_{n-1}^2] \\ \mathbb{E}[X_n\overline{X}_{n-1}] \end{pmatrix}. \tag{2.2.4}$$

The mean-square stability region of the method (2.0.1), that is the domain of p, q such that the second moments of the numerical solution $\mathbb{E}[X_n^2], \mathbb{E}[\overline{X}_n^2]$ tend to 0 as $n \to +\infty$, is then computed by checking the values of p and q for which the largest eigenvalue of the matrix in (2.2.4) is smaller than 1. To do this, we need to choose the noise integrator. We note that both noise integrators MT2 and W2Ito1 have the following stability function

$$R^{\Sigma}(p,q,\xi_n) = 1 + q\xi_n + \frac{q^2}{2}(\xi_n^2 - 1). \tag{2.2.5}$$

The resulting stability region is computed numerically for real p and is presented in Figure 2.2.1a. For comparison, the light gray region in Figure 2.2.1a is the stability region of the exact solution for which $\mathbb{E}(X(t)^2) \to 0$ as $t \to +\infty$ and is given by the condition $\mathrm{Re}(\lambda) + \frac{|\mu|^2}{2} < 0$.

2.2.2 Improving the stability

We consider slight modifications to the method to enhance the stability region without compromising the order of convergence or significantly increasing the computational cost. The first modification brings the term $hF(\overline{X}_{n-1})$ inside the $\sqrt{h}$ term of the Taylor expansion of the noise integrator. This is enough to obtain the desired order 2 and simplifies the stability matrix in (2.2.3). We obtain the following method

$$\begin{aligned} X_{n+1} &= X_n + hF(\overline{X}_n) + \hat{\Phi}_h^{\Sigma}(X_n, X_n + \frac{h}{4}F(\overline{X}_{n-1})), \\ \overline{X}_n &= X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n, \quad \text{with } \overline{X}_{-1} = X_0, \end{aligned} \tag{2.2.6}$$

with the following choice of the noise integrator $\Phi_h^{\Sigma}(X_n, X_n^{(1)})$:

1. Modified MT2 weak order 2 method from [4]

$$\begin{aligned} \Phi_h^{\Sigma}(X_n, X_n^{(1)}) = X_n &+ \frac{1}{2}\sum_{a=1}^{d}\Big(\sigma\Sigma_a\big(X_n + \sigma\Sigma(X_n)J_a\big) - \sigma\Sigma_a\big(X_n - \sigma\Sigma(X_n)J_a\big)\Big) \\ &+ \frac{\sigma\sqrt{h}}{2}\Big(\sigma\Sigma\big(X_n^{(1)} + \sqrt{\frac{h}{2}}\sigma\Sigma(X_n)\chi\big) + \sigma\Sigma\big(X_n^{(1)} - \sqrt{\frac{h}{2}}\sigma\Sigma(X_n)\chi\big)\Big)\xi_n. \end{aligned}$$

2. Modified W2Ito1 method from [76]

$$\begin{aligned} \Phi_h^{\Sigma}(X_n, X_n^{(1)}) &= X_n + \sqrt{h}\sum_{a=1}^{d}\big(-\sigma\Sigma_a(X_n) + \sigma\Sigma_a(K_1^{(a)}) + \sigma\Sigma_a(K_2^{(a)})\big)\xi_{n,a} \\ &\qquad + 2\sqrt{h}\sum_{a=1}^{d}\big(\sigma\Sigma_a(X_n) - \sigma\Sigma_a(K_2^{(a)})\big)\hat{J}_{a,a}, \\ K_1^{(a)} &= X_n^{(1)} + \frac{\sqrt{h}}{2}\sigma\Sigma_a(X_n)\hat{\chi}_1 + \sqrt{h}\sum_{\substack{b=1 \\ b\neq a}}^{d}\sigma\Sigma_b(X_n)\hat{J}_{a,b}, \\ K_2^{(a)} &= X_n - \frac{\sqrt{h}}{2}\sigma\Sigma_a(X_n)\hat{\chi}_1. \end{aligned}$$

The notation coincides with that used in the definition of MT2 and W2Ito1 methods. The improved stability region can be found in Figure 2.2.1b. The next improvement of the stability region is achieved by modifying the Milstein-Tretyakov term of the noise integrator. We note that the term $\frac{q^2}{2}(\xi_n^2 - 1)$ from (2.2.5) results in the term q^4 in $\mathbb{E}[R_{11}^2], \mathbb{E}[R_{12}^2], \mathbb{E}[R_{11}R_{12}]$ from (2.2.4). We decrease the significance of this term by multiplying it by $1 + \frac{p}{2}$ which goes to 0 as p approaches -2. This is achieved by replacing X_n used in the Milstein-Tretyakov term by $X_n + \frac{h}{2}F(\overline{X}_n)$. The updated method is

$$\begin{aligned} X_{n+1} &= X_n + hF(\overline{X}_n) + \hat{\Phi}_h^{\Sigma}(X_n, X_n + \frac{h}{4}F(\overline{X}_{n-1}), X_n + \frac{h}{2}F(\overline{X}_n)), \\ \overline{X}_n &= X_n + \frac{1}{2}\sqrt{h}\sigma\Sigma(X_n)\xi_n, \quad \text{with } \overline{X}_{-1} = X_0, \end{aligned} \tag{2.2.7}$$

with the noise integrator being one of the following options:

1. Modified weak order 2 method from [4]

$$\Phi_h^{\Sigma}(X_n, X_n^{(1)}, X_n^{(2)}) = X_n + \frac{1}{2}\sum_{a=1}^{d}\Big(\sigma\Sigma_a\big(X_n + \sigma\Sigma(X_n^{(2)})J_a\big) - \sigma\Sigma_a\big(X_n - \sigma\Sigma(X_n^{(2)})J_a\big)\Big)$$
$$+ \frac{\sigma\sqrt{h}}{2}\Big(\sigma\Sigma\big(X_n^{(1)} + \sqrt{\frac{h}{2}}\sigma\Sigma(X_n^{(2)})\chi\big) + \sigma\Sigma\big(X_n^{(1)} - \sqrt{\frac{h}{2}}\sigma\Sigma(X_n^{(2)})\chi\big)\Big)\xi_n.$$

2. Modified W2Ito1 method from [76]

$$\Phi_h^{\Sigma}(X_n, X_n^{(1)}, X_n^{(2)}) = X_n + \sqrt{h}\sum_{a=1}^{d}\big(-\sigma\Sigma_a(X_n) + \sigma\Sigma_a(K_1^{(a)}) + \sigma\Sigma_a(K_2^{(a)})\big)\xi_{n,a}$$
$$+ 2\sqrt{h}\sum_{a=1}^{d}\big(\sigma\Sigma_a(X_n) - \sigma\Sigma_a(K_2^{(a)})\big)\hat{J}_{a,a},$$
$$K_1^{(a)} = X_n^{(1)} + \frac{\sqrt{h}}{2}\sigma\Sigma_a(X_n^{(2)})\hat{\chi}_1 + \sqrt{h}\sum_{\substack{b=1\\ b\neq a}}^{d}\sigma\Sigma_b(X_n^{(2)})\hat{J}_{a,b},$$
$$K_2^{(a)} = X_n - \frac{\sqrt{h}}{2}\sigma\Sigma_a(X_n^{(2)})\hat{\chi}_1.$$

Further experiments that modified the coefficient $\frac{1}{2}$ of $F(\overline{X}_n)$ confirmed that $X_n + \frac{h}{2}F(\overline{X}_n)$ is the optimal choice for $X_n^{(2)}$. The updated stability region is shown in Figure 2.2.1c. However, in our broader experiments, modifications (2.2.6) and (2.2.7) did not lead to a significant improvement in stability compared to the original version (2.0.1). As a result, we opted to use for simplicity the original version (2.0.1) in our numerical simulations.

2.3 Numerical experiments

We present experiments that confirm the convergence order two of PVD-2 for sampling the invariant measure. We explore several one and two-dimensional problems as well as higher dimensional problems to emphasize that the method converges regardless of dimensionality. In the following, we fix $\sigma = 1$.

In our experiments, we compare the performance of PVD-2, given by (2.0.1), against the following methods: Euler-Maruyama (EM), Leimkuhler-Matthews with drift correction (LMd) (referred to as Hummer-Leimkuhler-Matthews in some sources) [64], the Strang splitting between Runge-Kutta 4 and W2Ito1 noise integrator (RK4[W2Ito1]) [76], and Leimkuhler-Matthews with time rescaling (LMt) [64]. Properties of these methods are summarised in Table 2.3.1, below.

Note that LMd does not converge for general variable diffusion in dimensions larger than $d = 1$. The method RK4[W2Ito1] uses Strang splitting, doubling the number of force evaluations, giving a total of $4 \times 2 = 8$. The same order could be achieved with fewer F evaluations, however, our aim is to compare against a highly-accurate integrator as a challenging baseline. Method LMt uses a constant stepsize h in a transformed time variable $\tau(t)$. For plots, we display error curves with an effective step

Method	**Weak Order**	**Sampling Order**	**# F Eval.**	**# Σ Eval.**
EM	1	1	1	1
LMd (dim. $d = 1$)	1	1	1	1
RK4[W2Ito1]	2	2	8	3
LMt	1	2	1	1
PVD-2[W2Ito1]	1 (expected)	2 (expected)	1	3
PVD-2[MT2]	1 (expected)	2 (expected)	1	5

Table 2.3.1: Summary of method characteristics, including the number of F and Σ evaluations per step. *Sampling order* is the order of sampling of the invariant measure, i.e. p in Definition 1.2.4.

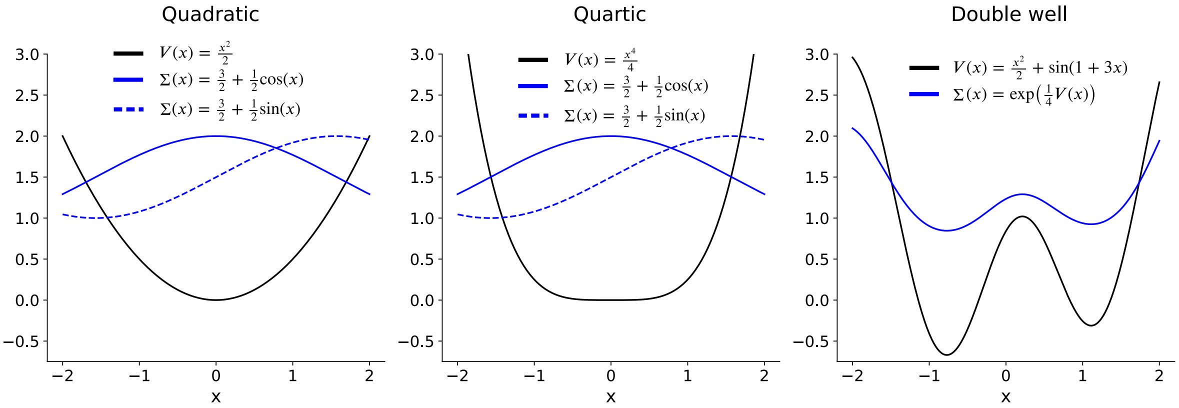


Figure 2.3.1: The potentials and diffusion coefficients used in one-dimensional experiments.

size $h' = h\left\langle \frac{dt}{d\tau} \right\rangle$, where $\langle \cdot \rangle$ denotes a trajectory average. The number of Σ evaluations for PVD-2[W2Ito1] and PVD-2[MT2] comes directly from the number of Σ evaluations for the noise integrator method, see the introduction of the new method in the beginning of Section 2.

2.3.1 One dimension

We consider three potentials of increasing complexity, namely the quadratic potential $V(x) = x^2/2$, the quartic potential $V(x) = x^4/4$ and the asymmetric double-well potential $V(x) = x^2/2 + \sin(1 + 3x)$. For the first two potentials we consider cosine diffusion $\Sigma(x) = \frac{3}{2} + \frac{1}{2}\cos(x)$ and sine diffusion $\Sigma(x) = \frac{3}{2} + \frac{1}{2}\sin(x)$. For the double-well, we consider diffusion of the form $\Sigma(x) = \exp(\frac{1}{4}V(x))$. These potentials and diffusion coefficients are illustrated in Figure 2.3.1. Under mild conditions on $V(x)$, diffusion of the form $\Sigma(x) \propto \exp(2\sigma^{-2}V(x))$ for $\sigma > 0$ is known to be nearly optimal diffusion for enhancing the crossing rate between metastable wells [46, 64]. Note that $f = -\nabla V$ is globally Lipschitz for the quadratic and double-well potentials, but not for the quartic potential, making it an interesting test case.

For computing the L_1 error, we divide the subset $[-5, 5]$ of the x-domain into $M = 30$ bins and for a fixed T compute the mean error:

$$\text{Error}(h, T) := \frac{1}{M}\sum_{i=1}^{M} |\omega_i - \hat{\omega}_i(h, T)|, \tag{2.3.1}$$

where ω_i is the exact occupancy probability of the i^{th} interval and $\hat{\omega}_i$ is the empirical

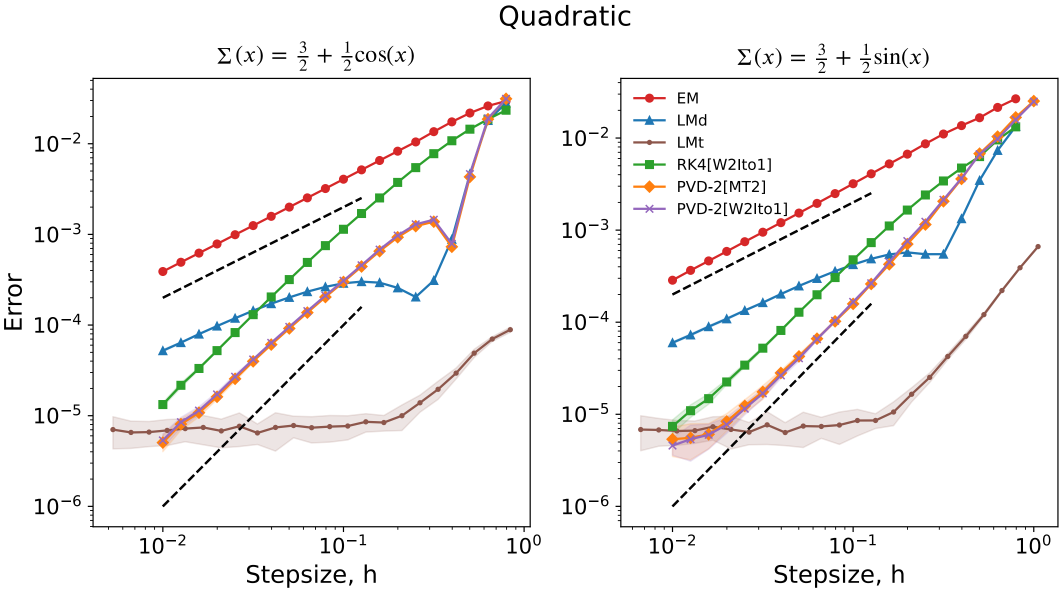


Figure 2.3.2: Convergence for sampling the invariant measure in a quadratic potential $V(x) = x^2/2$. Left: diffusion $\Sigma(x) = \frac{3}{2} + \frac{1}{2}\cos(x)$. Right: diffusion $\Sigma(x) = \frac{3}{2} + \frac{1}{2}\sin(x)$.

estimate when running trajectories with fixed stepsize h up to a final time T. In all experiments, we set $T = 5 \times 10^7$ and ran each integrator using time steps starting from 10^{-2}, increasing by a factor of $10^{0.1}$ at each step until the method became unstable. Results for the quadratic, quartic and double-well are shown in Figures 2.3.2, 2.3.3 and 2.3.4, respectively. For all curves, we also display an estimate of the Monte-Carlo error in the bias (shaded areas), considering the standard deviation of 10 independent trajectories.

Both variants of PVD-2 consistently achieve second-order convergence across various environments, including challenging non-globally Lipschitz cases like the quartic potential. They also yield lower errors than RK4[W2Ito1], using only one force evaluation per step versus RK4[W2Ito1]'s eight. Nevertheless, in one-dimension, time-rescaling combined with the Leimkuhler-Matthews (LMt) consistently results in the lowest error for any given stepsize. This highlights the importance of transforming a multiplicative noise to additive whenever possible [64]. Note, however, that LMt can only be applied in the multiplicative setting for isotropic diffusion. For this reason, we exclude this method from our higher-dimensional benchmarks.

2.3.2 Two dimensions

We consider a 4-well potential given by

$$V(x_1, x_2) = \sqrt{\frac{17}{16} - 2x_1^2 + x_1^4} + \sqrt{\frac{17}{16} - 2x_2^2 + x_2^4}, \tag{2.3.2}$$

along with four diffusion tensors of increasing complexity:

(A) *Constant*: A constant, anisotropic diffusion:

$$\Sigma_A(x_1, x_2) = \begin{bmatrix} 2 & 0 \\ 0 & \frac{3}{2} \end{bmatrix}.$$

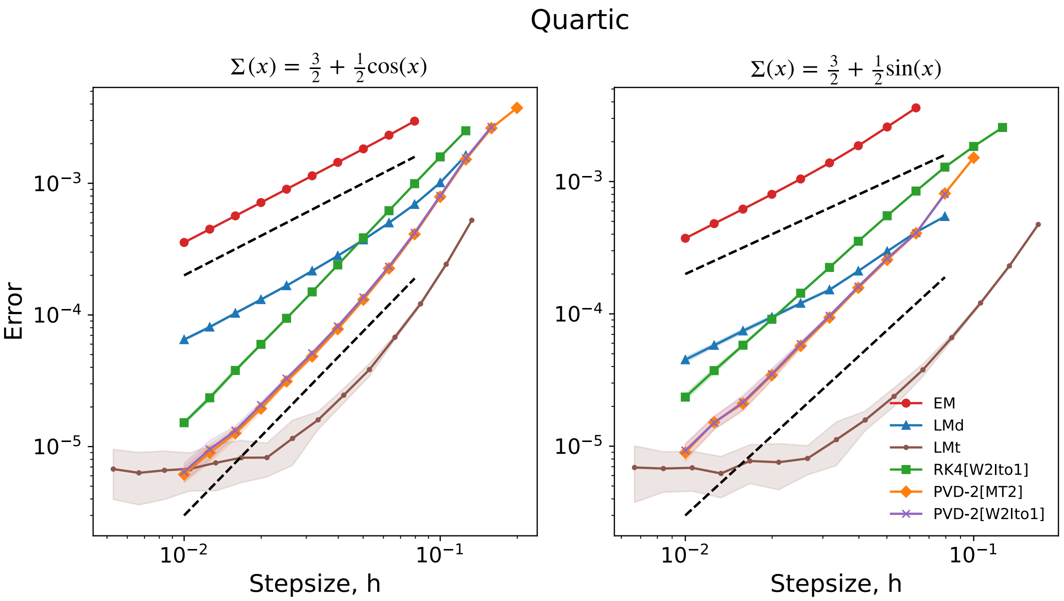


Figure 2.3.3: Convergence for sampling the invariant measure in a quartic potential $V(x) = x^4/4$. Left: diffusion $\Sigma(x) = \frac{3}{2} + \frac{1}{2}\cos(x)$. Right: diffusion $\Sigma(x) = \frac{3}{2} + \frac{1}{2}\sin(x)$.

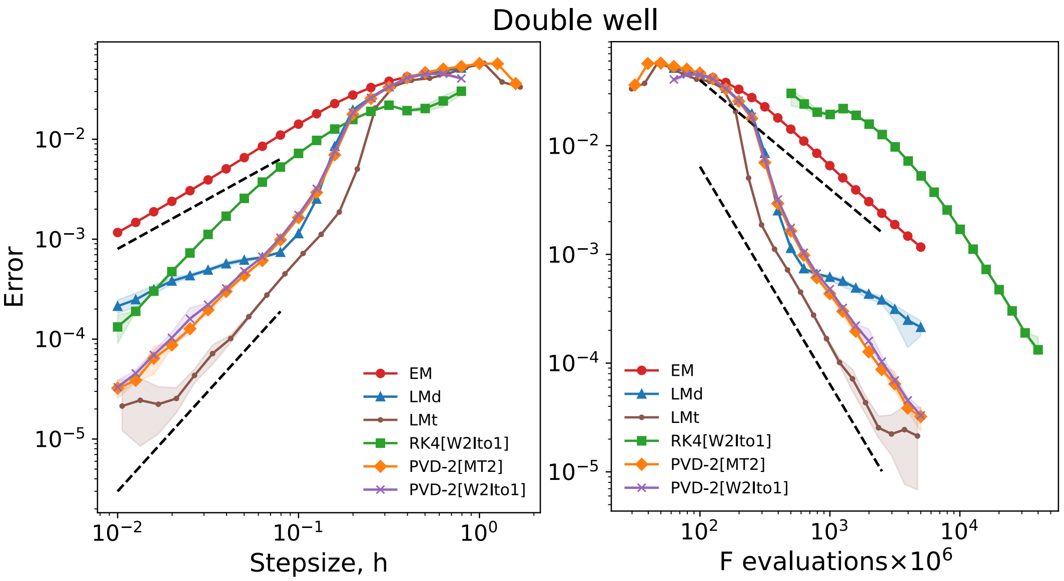


Figure 2.3.4: Convergence for sampling the invariant measure in a double-well potential $V(x) = \frac{x^2}{2} + \sin(1 + 3x)$ with diffusion $\Sigma(x) = \exp\left(\frac{1}{4}V(x)\right)$. The left figure shows the error convergence against stepsize and the right figure shows the error convergence against the number of F evaluations, a proxy for computational cost.

(**B**) *Isotropic I*: Non-homogeneous diffusion given by the Moro-Cardin tensor [55] which impedes convergence due to the low level of noise in the central high-potential region:

$$\Sigma_B(x_1, x_2) = \left(1 + A \exp\left(-\frac{\|x\|^2}{2\epsilon^2}\right)\right)^{-1} I, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

where $A = 5$ and $\epsilon = 0.3$.

(**C**) *Isotropic II*: Non-homogeneous diffusion which aids convergence due to the high level of noise in the central high-potential region:

$$\Sigma_C(x_1, x_2) = \left(1 + A \exp\left(-\frac{\|x\|^2}{2\epsilon^2}\right)\right) I, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

where $A = 1$ and $\epsilon = 0.3$.

(**D**) *Anisotropic*: An anisotropic diffusion given by

$$\Sigma_D(x_1, x_2) = I - \frac{xx^T}{2\|x\|^2 + 1}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

which can be written in terms of the planar angle $\theta = \arg(x)$ as

$$\Sigma_D(x_1, x_2) = I - \frac{\|x\|^2}{2\|x\|^2 + 1} \begin{bmatrix} \cos^2(\theta) & \cos(\theta)\sin(\theta) \\ \cos(\theta)\sin(\theta) & \sin^2(\theta) \end{bmatrix}.$$

The pre-factor of $\frac{\|x\|^2}{2\|x\|^2+1}$ ensures that θ-dependent component vanishes at $x = \mathbf{0}$, thus guaranteeing that $\Sigma_D(x_1, x_2)$ is everywhere smooth.

In Figure 2.3.5, we visualise how these diffusion tensors vary relative to the energy contours of (2.3.2). To measure convergence, we compute the L_1 error of the square-norm observable:

$$\text{Error}(h, N, T) := \left| O - \hat{O}(h, N, T) \right|,$$

where $O = \int (x_1^2 + x_2^2)\rho(x_1, x_2)dx_1dx_2$ is the exact square-norm average and $\hat{O}$ is the empirical estimate when running N trajectories with fixed stepsize h and averaging the value of the observable at time T. We fix $N = 10^5$ and $T = 30$ and ran each integrator with fixed step sizes $h \in \{10^{-2}, 10^{-1.9}, \dots, 10^{-0.1}, 10^{0.0}\}$. Results are shown in Figure 2.3.6.

Note that in all cases, PVD-2 is the best performing integrator for small stepsizes, within standard error. However, metastability is more severe in the quadruple-well compared to one-dimensional problems (Section 2.3.1). This is especially true for diffusion tensor Isotropic I (B). Here, diffusion vanishes over the central maximum, inhibiting well transitions. This highlights the loss of second-order convergence (for any method) in Figure 2.3.6(B); the simulation time is too short to observe complete sampling. In our numerical tests, increasing T did not improve these convergence rates, suggesting that the temporal convergence to equilibrium is very slow for the considered diffusion tensor.

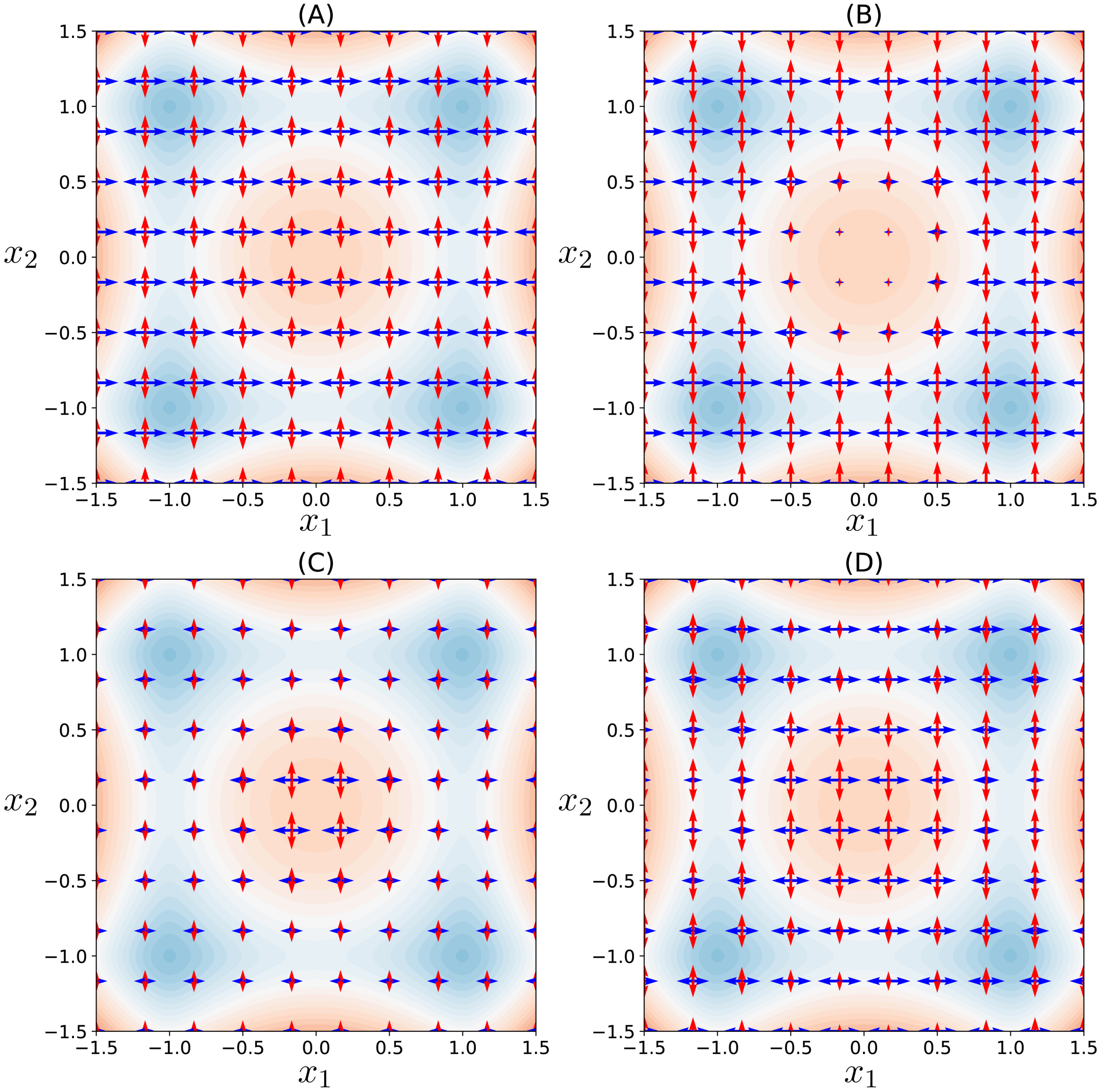


Figure 2.3.5: Contours of the quadruple well potential (2.3.2) are shown with the various diffusion tensor fields. Diffusion fields $\Sigma(x_1, x_2)$ are visualised by the magnitude of the expected noise increment in the x_1 (blue) and x_2 (red) directions shown at each grid point. For better comparison, the diffusion arrows in (B) and (D) are scaled by a factor of 2 compared to (A) and (C).

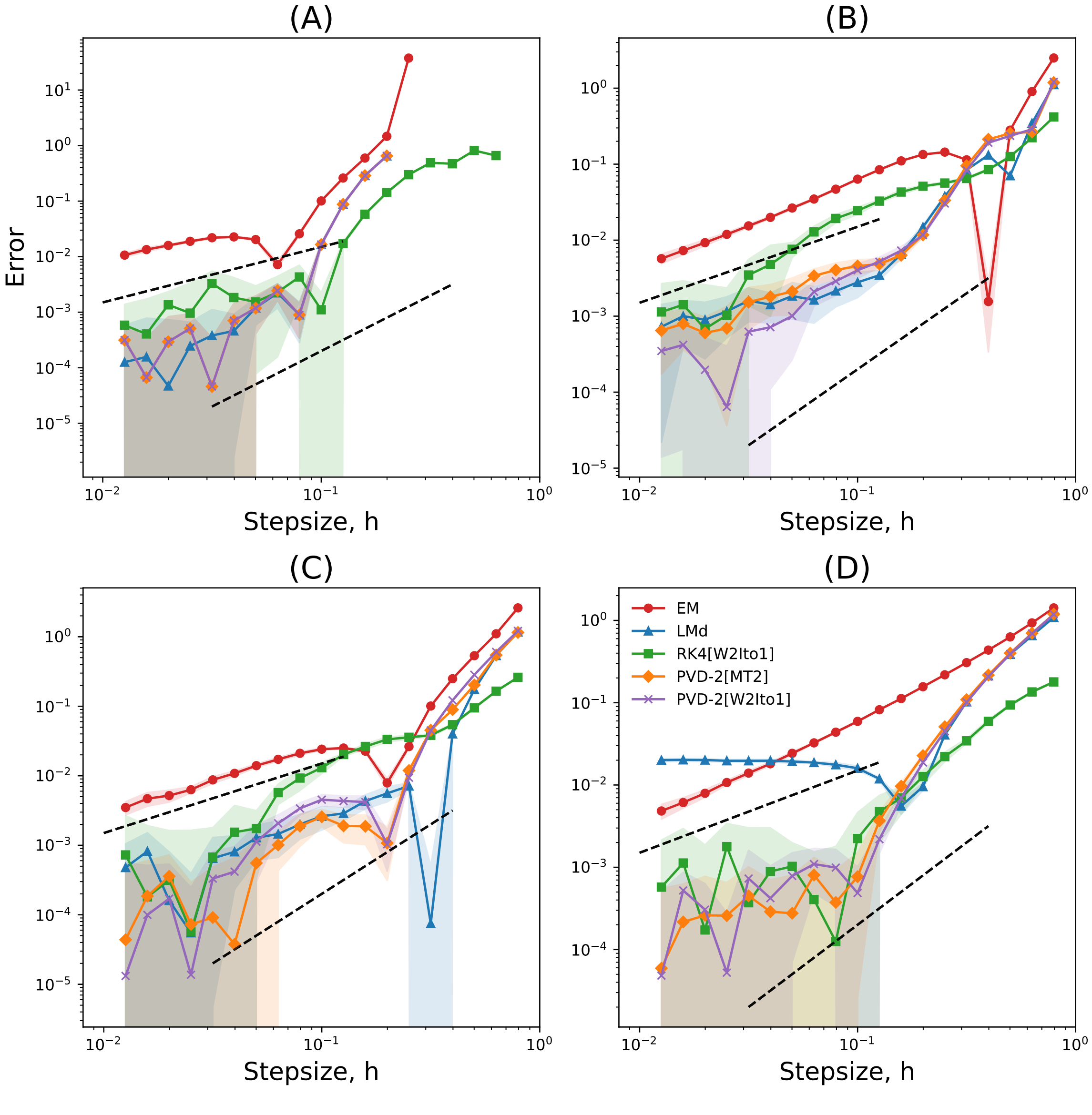


Figure 2.3.6: Convergence for sampling the invariant measure in the 2D quadruple-well potential for the four diffusion tensors depicted in Figure 2.3.5. Note that in (A), both variants of PVD-2 perform identically hence only PVD-2[W2Ito1] shows. The computations were performed on the Baobab cluster of the University of Geneva using the Julia programming language.

2.3.3 Higher dimensions

We consider the high-dimensional ring potential, which was used as a test problem in [72, Sect. 7]. For $x \in \mathbb{R}^d$, the potential is given by

$$V(x) = \frac{1}{2}k(1 - \|x\|)^2, \tag{2.3.3}$$

where $\|x\| := \sqrt{\sum_{i=1}^d x_i^2}$ is the L_2 norm of x. We use $k = 50$ and run experiments for $d = 10$ and $d = 100$. For the diffusion tensor, we use a rank-one update to the identity matrix which inhibits diffusion in the radial direction, given by

$$\Sigma(x) = I - \frac{xx^T}{2\|x\|^2}. \tag{2.3.4}$$

This diffusion tensor is non-smooth at $x = 0$. However, in high dimensions, and for large k, the invariant measure of the ring potential is highly concentrated near the unit sphere and there is thus negligible probability mass near this point. We therefore might still expect to observe second-order convergence when using PVD-2.

To measure convergence, we compute the L_1 error of the square-norm observable:

$$\text{Error}(h, T) := \left| O - \hat{O}(h, T) \right| ,$$

where $O = \int (\sum_{i=1}^d x_i^2)\rho(x)dx$ is the exact square-norm average and $\hat{O}$ is the empirical estimate when running trajectories with fixed stepsize h up to a final time T. For $d = 10$ experiments, we set $T = 10^6$. For $d = 100$, $T = 10^4$. We compare the performance of the same integrators as in Section 2.3.2, except instead of W2Ito1 we use MT2 with Runge-Kutta 4 Strang splitting (RK4[MT2]) [4]. In our low-dimensional experiments, W2Ito1 and MT2 performed very similarly, however MT2 is simpler to implement for large d.

In both high-dimensional experiments, we recovered a second order convergence curve for our new method. In moderate dimensions ($d = 10$), we observe that PVD-2[MT2] even outperforms RK4[MT2], whilst requiring only a single F evaluation per step, instead of eight. In high dimensions ($d = 100$), although RK4[MT2], it has much more limited stability that all other tested methods. In contrast, LMd does not converge in the multivariate setting.

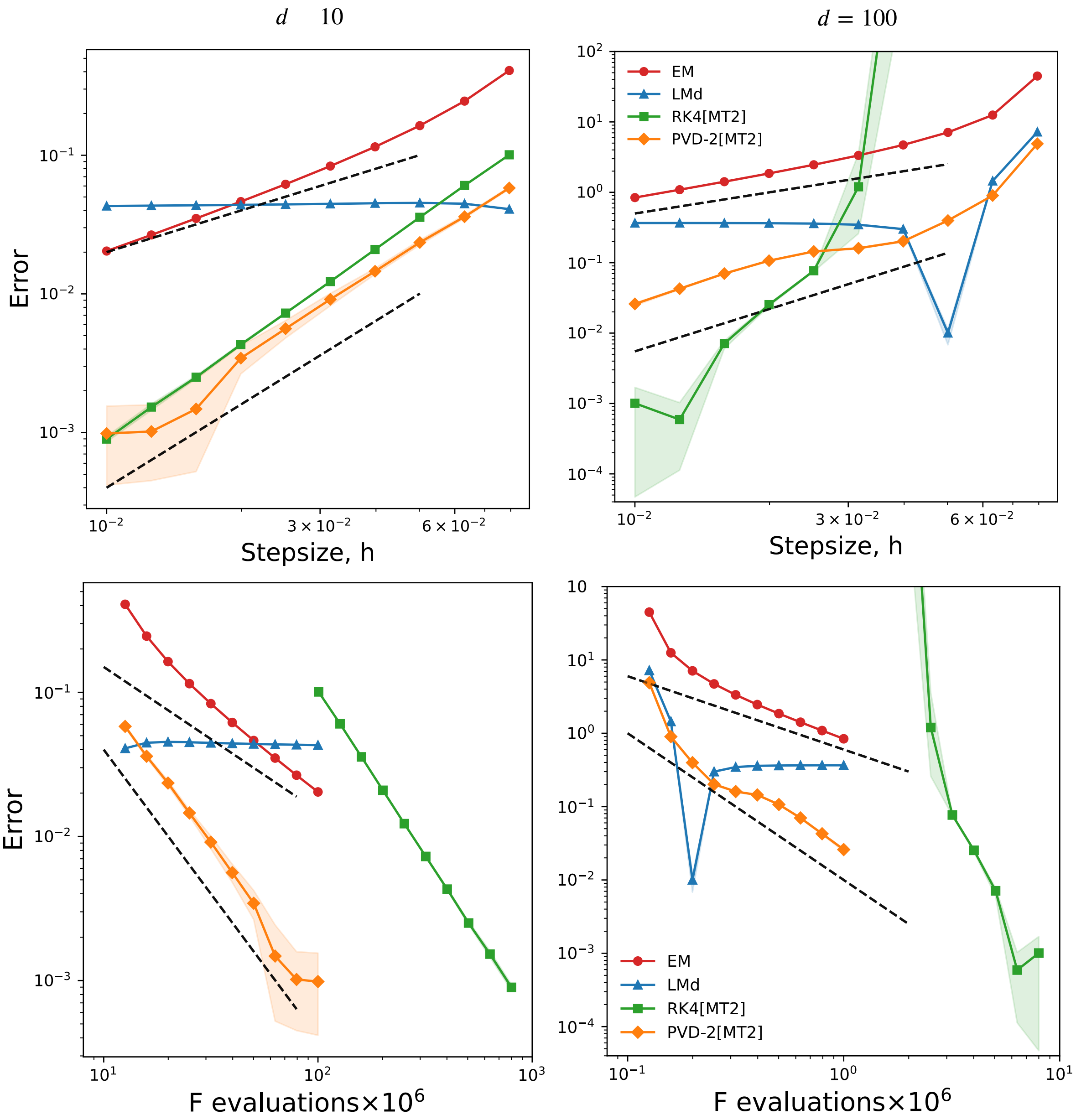


Figure 2.3.7: Convergence to the observable $\|x\|$ in the ring-potential (2.3.3) with diffusion tensor given by (2.3.4). Left: dimension $d = 10$. Right: dimension $d = 100$.

Chapter 3

Algebraic structure of exotic aromatic forests

In this chapter, we further generalize the class of forests by extending it to exotic aromatic forests in Section 3.2 and clumped forests in Section 3.3. Our focus is on the algebraic structures of grafted, exotic, aromatic, and clumped forests, aiming to apply these structures in the context of numerical analysis. This chapter consolidates the results from [7, 11] within a unified formalism.

In Section 3.1, we introduce a formal algorithm for generating order conditions for invariant measure sampling. Although the algorithm is developed for (1.1.1), we note that the theory also applies to (1.1.4), with the primary distinction being that the forest transformation, called ELI, introduced in Section 3.1, cannot be applied. Subsequently, we reduce the number of order conditions by examining their algebraic structure and demonstrating that they form a character with respect to the concatenation of forests. The results presented in Section 3.1 are published and can be found in [7].

In Section 3.2, we define the concept of aromas. Aromas are graphs that emerge when the divergence of a vector field and the inner product between two vector fields are incorporated into the framework of forests. Classical forests combined with aromas form aromatic forests, which are a powerful tool in geometric numerical integration. We extend the grafting product to aromatic forests, define the associated D-algebra structure, and investigate the Grossman-Larson Hopf algebroid structure that replaces the Hopf algebra structure from classical theory. Additionally, we examine how the introduction of aromas affects the structure of a pre-Hopf algebra, leading us to define the resulting structure as a pre-Hopf algebroid.

In Section 3.3, we introduce clumped forests, which generalize aromatic forests by associating each aroma with a rooted component of the forest. Clumped forests are a necessary component in defining the substitution law. This necessity arises from the fact that substitution is expressed using homomorphisms of D-algebras that map vertices to formal sums of aromatic trees. Additionally, since grafting is not linear over the ring of aromas in the right operand, we cannot define monomials of aromatic trees over this ring. As a result, the substitution law for aromatic forests is formulated using a coaction of clumped forests on aromatic forests.

In Section 3.4, we consolidate the results discussed throughout the chapter and extend them to the framework of exotic aromatic forests. A key property of exotic aromatic forests is that pairs of grafted vertices (referred to as *lianas*) can connect different connected components, forming components with multiple roots. This characteristic challenges many classical results, which often rely on the assumption that each component has a single root, rendering some results invalid or requiring alternative proofs. The results presented in Sections 3.2, 3.3, and 3.4 are part of [11], a collaboration with Adrien Laurent, and have been submitted for publication.

A fundamental object connecting the algebraic and combinatorial structures of forests to the numerical analysis of time integrators is the S-series. An S-series is a formal sum indexed by a class of forests, where each term corresponds to the differential operator associated with a given forest, with coefficients determined by the specific integrator. The primary focus is the S-series over exotic forests, which serves as a bridge to translate the algebraic results established in this chapter into numerical applications.

Recall that EF denotes the set of exotic forests introduced in Section 1.4 and $\mathcal{EF}$ denotes the corresponding vector space. Let $\overline{\mathcal{EF}}$ be the space of formal sums of the following form

$$\sum_{\pi \in EF} a(\pi)\pi, \quad \text{with } a \in \mathcal{EF}^*.$$

Recall the symmetry coefficient $\sigma(\pi) := |\mathrm{Aut}(\pi)|$. Let $\delta_\sigma : \mathcal{EF}^* \to \overline{\mathcal{EF}}$ be the isomorphism given by

$$\delta_\sigma(a) = \sum_{\pi \in EF} \frac{a(\pi)}{\sigma(\pi)} \pi.$$

We will abuse the notation and use δ_σ to denote analogous isomorphisms for other spaces. The space will be made clear from the context and will always be the domain of the functional to which δ_σ is applied.

Definition 3.0.1. S-series over exotic forests are defined as $S := \mathbb{F} \circ \delta_\sigma$, that is,

$$S(a) = \sum_{\pi \in EF} \frac{a(\pi)}{\sigma(\pi)} \mathbb{F}(\pi).$$

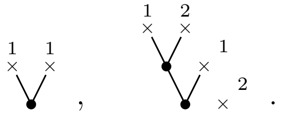
S-series over exotic forests are called *exotic S-series.* If $a \in \mathcal{EAT}^*$, that is, a is non-zero only on exotic aromatic trees, then $S(a)$ is called an *exotic B-series* and denoted by $B(a)$.

S-series over grafted forests F_g is defined analogously. We consider the following form of stochastic Runge-Kutta methods solving (1.1.1) with $F = -\nabla V$ denoting the drift term. Stochastic Runge-Kutta methods can be extended to solve (1.1.4) by adding an appropriate Σ term to the noise terms of the scheme.

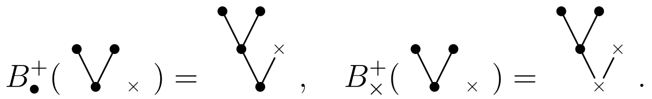
Definition 3.0.2. Let $a_{ij}, b_i, d_i^{(k)}$ be the coefficients defining the stochastic Runge-Kutta scheme, and $\xi_n^{(k)} \sim \mathcal{N}(0, I_d)$ be independent Gaussian random vectors. Then, the stochastic Runge-Kutta scheme has the form:

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

We shall assume for simplicity of the presentation that $l = 1$ which is sufficient to achieve weak order 2 or order 3 with respect to the invariant measure. We note that $l > 1$ is necessary in general to achieve high order [42]. The analysis in this paper extends naturally to the $l > 1$ case by considering grafted forests with decorated grafted vertices. Two grafted vertices can form a pair only if they are decorated by the same number. For example, for $l = 2$, we should consider grafted of the form



We recall the map $B_c^+ : F_D \to T_D$ for $c \in D$, which attaches all the roots of a given forest to a new vertex, thereby making it the root of the resulting tree. For example,

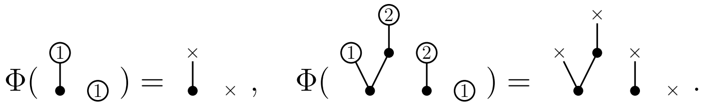


Proposition 3.0.3. *Let $S(a)$ be the S-series over grafted forests of a stochastic Runge-Kutta method with coefficients b_i, a_{ij}, d_i for $i, j = 1, \ldots, s$, then the map $a : F_g \to \mathbb{R}$ is defined as*

$$a(B_\bullet^+(\pi)) = \sum_{i=0}^{s} b_i a^{[i]}(\pi), \quad a^{[i]}(B_\bullet^+(\pi)) = \sum_{j=1}^{s} a_{ij} a^{[j]}(\pi), \quad a^{[i]}(B_\times^+(\pi)) = d_i,$$

with $a(\pi_1 \cdot \pi_2) = a(\pi_1)a(\pi_2)$ and $a^{[i]}(\pi_1 \cdot \pi_2) = a^{[i]}(\pi_1)a^{[i]}(\pi_2)$ for $\pi, \pi_1, \pi_2 \in F_g$.

We recall that a pair of grafted vertices is referred to as a *liana*. Let $\Phi : EF \to F_g$ denote the map that forgets the pairing between grafted vertices, that is, it sends all $k \in \mathbb{N}$ to $\times$ and keeps $\bullet$ unchanged. For example,



Proposition 3.0.4. *The expectation of an S-series over grafted forests $S(a)$ is the exotic S-series $S(a \circ \Phi)$, that is,*

$$\mathbb{E}\left[S(a)\right] = S(a \circ \Phi).$$

Proof. We use the definitions of S-series, as well as Theorem 1.4.12. □

We now present some numerical applications of the algebraic results developed throughout the chapter. Let e denote the functional $e : EF \to \mathbb{R}$ corresponding to the exact solution, such that $S(e) = \exp(h\mathcal{L})$. An integrator with an exotic S-series $S(a)$, defined as $\mathbb{E}[\phi(X_1)] = S(a)$, is said to have weak order q if $a(\pi) = e(\pi)$ for all $\pi \in EF$ with $|\pi| \leq q$. An integrator with at least weak order one is referred to as consistent.

Let us consider an equivalence relation between functionals over exotic forests, denoted by $\sim$, where $a \sim b$ for $a, b \in \mathcal{EF}^*$ if there exists a chain of transformations consisting of ELI and IBP (see Section 3.1) such that $\delta_\sigma(a) \to \delta_\sigma(b)$. A key property of the relation $\sim$ is presented below,

$$a \sim b \quad \Longrightarrow \quad \int_{\mathbb{R}^d} S(a)[\phi](x)\rho_\infty(x)dx = \int_{\mathbb{R}^d} S(b)[\phi](x)\rho_\infty(x)dx.$$

An integrator with an exotic S-series $S(a)$ is said to have order p for the invariant measure sampling if $a \sim \omega$ with $\omega(\pi) = 0$ for all $\pi \in EF$ with $|\pi| \leq p$.

Let ET denote the set of exotic trees, that is, the set of exotic forests with a single root. Corollary 3.1.9 of Theorem 3.1.8 introduced in Section 3.1 reduces the number of order conditions significantly by considering the value of ω over ET instead of EF.

Corollary 3.1.9. *Given an integrator satisfying the assumptions discussed in Chapter 1 with $\mathbb{E}[\phi(X_1)] = S(a)$, it has order p for the invariant measure sampling if $\omega(\tau) = 0$ for $\tau \in ET$ with ω defined as in* (3.1.1)*.*

All order conditions for invariant measure sampling up to order 3 can be found in Appendix .0.2. We note that the results presented below are obtained for S-series over exotic *aromatic* forests. Exotic aromatic forests are a generalization of exotic forests which is described in Section 3.2. The vector spaces of exotic aromatic forests and trees are denoted by $\mathcal{EAF}$ and $\mathcal{EAT}$, respectively.

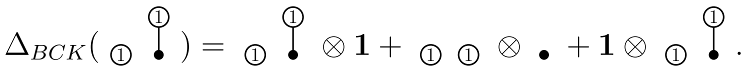
Let Φ_h^1 and Φ_h^2 be two integrators that share the same timestep while having independent noise terms. Theorem 3.4.3 from Section 3.4 establishes that the composition of these two integrators, each with an exotic S-series, results in an integrator that also possesses an exotic S-series of a specific form. We present the definition of the Butcher-Connes-Kreimer coproduct used in the statement of Theorem 3.4.3 and studied in Section 3.2.4.

Definition 3.4.2. The Butcher-Connes-Kreimer coproduct on $\mathcal{EAF}$ is defined as

$$\Delta_{BCK}(\pi) := \sum_{\pi_0 \subset \pi} \pi \setminus \pi_0 \otimes \pi_0,$$

where the sum runs over all rooted exotic aromatic subforests $\pi_0 \in \mathcal{EAF}$ of π such that $\pi \setminus \pi_0 \in \mathcal{EAF}$ and there are no edges going from π_0 to $\pi \setminus \pi_0$ in π.

An important property of Δ_{BCK} over exotic aromatic forests is that it keeps the vertices of a liana on the same side of the tensor product, for example,



Let $m_{\mathbb{R}}$ denote the multiplication map over $\mathbb{R}$.

Theorem 3.4.3. *Consider two independent integrators Φ_h^1 and Φ_h^2 with exotic S-series $S(a_1)$ and $S(a_2)$, then the composition of Φ_h^1 and Φ_h^2 has the following S-series*

$$\mathbb{E}[\phi((\Phi_h^2 \circ \Phi_h^1)(x))] = S(a_1 * a_2)[\phi](x).$$

with $a_1 * a_2 = m_{\mathbb{R}} \circ (a_1 \otimes a_2) \circ \Delta_{BCK}$ where Δ_{BCK} is the Butcher-Connes-Kreimer coproduct over exotic forests.

Let $S_F(a)$ denote the exotic S-series of a numerical integrator solving an SDE with drift F. The drift can be chosen to be a B-series $B_F(b)$, where $b \in EAT^*$ is a functional that is non-zero only on exotic aromatic trees. Theorem 3.4.5 asserts that applying a numerical integrator to an SDE with drift $S_F(b_0)$ is equivalent to applying a numerical integrator with an exotic S-series of a specific form to the SDE with drift F.

The description of the substitution law for exotic aromatic S-series relies on the coaction $\Delta_{CEM} : \mathcal{EAF} \to \mathcal{CEF}_1 \otimes \mathcal{EAF}$ with $\mathcal{CEF}_1 := \mathcal{S}(\mathcal{EAT})$ which is studied in detail in Sections 3.3 and 3.4.

Definition 3.4.4. Let the CEM coaction $\Delta_{CEM} : \mathcal{EAF} \to \mathcal{CEF}_1 \otimes \mathcal{EAF}$ over exotic aromatic forests be defined as

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

where $P(\pi)$ is the set of partitions of π with a partition $p \in P(\pi)$ being a set of exotic aromatic subtrees of π that covers all black vertices of π. When written in the left operand of the tensor product, p is interpreted as a monomial in $\mathcal{CEF}_1$. The exotic aromatic forests $\pi/_p$ denotes the forest obtained by contracting each element of p into a black vertex. If the forest $\pi \in \mathcal{EAF}$ doesn't have valid subforests $p \in \mathcal{CEF}_1$, then $\Delta_{CEM}(\pi) = \mathbf{1} \otimes \pi$. For details see the proof of Theorem 3.4.6.

Theorem 3.4.5. *Let $a \in \mathcal{EAF}^*$, $b \in \mathcal{EAT}^*$, then,*

$$S_{B_F(b)}(a) = S_F(b_c \star a), \quad \text{with } b_c \star a = m_{\mathbb{R}} \circ (b_c \otimes a) \circ \Delta_{CEM},$$

where b_c is the character of $\mathcal{CEF}_1$ that extends b.

We use Theorem 3.4.5, along with Theorem 3.1.8 introduced in Section 3.1, to extend the powerful techniques of backward error analysis and the modified equation—originally developed for the geometric numerical integration of ODEs—to the SDE context for invariant measure sampling. These techniques are formally presented in Theorems 3.0.5 and 3.0.7, using concepts introduced in Sections 3.3 and 3.4. Notably, the statements of Theorems 3.0.5 and 3.0.7 rely on the map A (Section 3.1) to be defined in a specific way that reduces all exotic forests to monomials of exotic trees which is known to be possible for overdamped Langevin equations with additive noise, i.e., problem (1.1.1), but not for more general problems.

The goal of backward error analysis is to find a modified vector field written formally as an exotic B-series,

$$h\widetilde{F} = B(b) = hF + h^2F_1 + h^3F_2 + \dots, \quad b\colon \mathcal{ET} \to \mathbb{R}, \quad b(\bullet) = 1,$$

for some vector fields F_1, F_2, ... that typically write as polynomials in the coordinates of F and its partial derivatives, such that the invariant measure of the ergodic integrator with S-series $S(a)$ coincides with the invariant measure of the modified dynamics of (1.1.1) with $F = -\nabla V$ replaced by $\tilde{F}$. The coefficient map $b \in \mathcal{ET}^*$ is the solution to the substitution $b_c \star e \sim a$ where $\star$ is the substitution law described in Theorem 3.4.5.

It is known [28, 2] that there exists a modified vector field for large classes of methods, such as stochastic Runge-Kutta methods. The calculations are tedious and were rewritten with exotic series in [41, 7]. There is, however, no proof that the calculations can be carried out up to any order in these works as there is no reason in general why the modified vector field could be written as an exotic B-series. A geometric justification of the importance of writing the modified vector field as an exotic B-series is given in [40]: it enforces that $\widetilde{F}$ is invariant with respect to orthogonal changes of coordinates, which is a natural property in the stochastic context. We provide here a simple and natural algebraic criterion, satisfied by large classes of methods, for the description of integrators that have a modified vector field in the form of an exotic B-series. In addition, we give the first explicit expression of the modified vector field $\widetilde{F}$ relying on the map A defined in Section 3.1. This shows in particular that exotic series are a powerful tool for the stochastic backward error analysis.

Theorem 3.0.5 (Backward error analysis)**.** *Consider a consistent method with the exotic S-series $S(a)$ for solving equation* (1.1.1)*. Assume that a is a character of $(\mathcal{EF},\cdot)$. Then, there exists a modified vector field $h\widetilde{F}=B(b)$ that can be written as an exotic B-series with a coefficient map $b\colon \mathcal{ET}\to\mathbb{R}$ satisfying $b_c\star e\sim a$, and given by*

$$b=\delta_\bullet+A\Big(\sum_{k=0}^{\infty}(-1)^k A^k_{\tilde{\star}e}(a-e)\Big)\big|_{\mathcal{ET}},$$

where $A_{\tilde{\star}e}:\mathcal{EF}^*\to\mathcal{EF}^*$ satisfies $A_{\tilde{\star}e}(x)=A(x)\tilde{\star}e$ and

$$a\tilde{\star}e=(a\otimes e)\circ\tilde{\Delta}_{CEM},\quad \textit{and}\quad \tilde{\Delta}_{CEM}(\pi)=\Delta_{CEM}(\pi)-\bullet\otimes\pi-\pi\otimes\bullet,$$

for $a\in\mathcal{EF}^*$, $\pi\in EF$ such that $|\pi|>1$.

Proof. After initializing $b_0=\delta_\bullet$, we construct recursively the coefficient map sequence (b_n) by

$$b_n=b_{n-1}+A(a-b_{n-1,c}\star e)\big|_{\mathcal{ET}},\quad b_{n-1,c}\star e=m_{\mathbb{R}}\circ(b_{n-1,c}\otimes e)\circ\Delta_{CEM},$$

with $b_{n-1,c}:=\exp^{\odot}(b_{n-1})$ and the coproduct $\Delta_{CEM}:\mathcal{EF}\to\mathcal{S}(\mathcal{ET})\otimes\mathcal{EF}$. Assume $b_{n-1}(\bullet)=1$, since $A^*(\bullet)=\bullet-$ ① ① and the method associated to a is consistent, we find

$b_n(\bullet)=b_{n-1}(\bullet)+(a-b_{n-1,c}\star e)(\bullet-$ ① ① $)=1+a(\bullet)-a($ ① ① $)-e(\bullet)+e($ ① ① $)=1,$

thus we obtain that $b_n(\bullet)=1$ for any n by induction. For all $\tau\in\mathcal{ET}$ such that $|\tau|>1$, using the reduced coproduct yields

$$\begin{aligned}b_n(\tau)&=b_{n-1}(\tau)+A(a)(\tau)-(b_{n-1}\otimes e)(A^*(\tau)\otimes\bullet)\\&\quad-(b_{n-1}\otimes e)(\bullet\otimes A^*(\tau))-A(b_{n-1,c}\tilde{\star}e)(\tau)\\&=b_{n-1}(\tau)-A(b_{n-1})(\tau)+A(a-e-b_{n-1,c}\tilde{\star}e)(\tau)\\&=A(a-e+b_{n-1,c}\tilde{\star}e)(\tau),\end{aligned}$$

where we recall that the only exotic tree τ for which $|\tau|\leq 1$ is $\tau=\bullet$ and we used $A(b_{n-1})=b_{n-1}$. The first values are

$$\begin{aligned}b_1&=\delta_\bullet+A(a-e)\big|_{\mathcal{ET}},\\b_2&=\delta_\bullet+A(a-e)\big|_{\mathcal{ET}}-A(A(a-e)\tilde{\star}e)\big|_{\mathcal{ET}},\\&\vdots\\b_n&=\delta_\bullet+A\big(\sum_{k=0}^{n-1}(-1)^kA^k_{\tilde{\star}e}(a-e)\big)\big|_{\mathcal{ET}},\quad\text{where } A_{\tilde{\star}e}(x)=A(x)\tilde{\star}e.\end{aligned}$$

Since $|A^*(\tau)|_e\leq|\tau|_e$ where $|\tau|_e$ is the number of edges of τ, and $\tilde{\Delta}^n_{CEM}(\tau)=0$ if $n\geq|\tau|_e$, we have for all $n\geq|\tau|_e$,

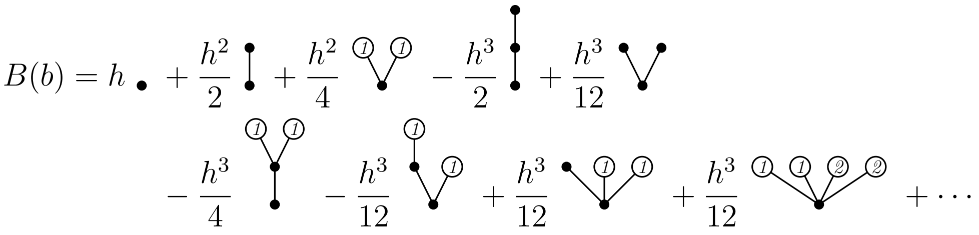
$$b_n(\tau)=\delta_\bullet+A\big(\sum_{k=0}^{|\tau|_e-1}(-1)^kA^k_{\tilde{\star}e}(a-e)\big)(\tau),$$

so that the sequence (b_n) converges to the desired coefficient map b by stationarity. □

Example 3.0.6. *Recall the Euler-Maruyama method for the Langevin equation* (1.1.1)*,*

$$X_1 = X_0 - h\nabla V(X_0) + \sqrt{h}\sigma\xi_1, \quad \text{with } \xi_1 \sim \mathcal{N}(0, I_d).$$

The first terms of the modified vector field $h\widetilde{F} = B(b)$ given by Theorem 3.0.5 for the Euler-Maruyama method are



Note that removing the stochastic terms (that are, the trees with lianas) does not yield the modified vector field for the Euler method with the standard deterministic backward error analysis (see [34, Chap. IX]). Indeed, high order for the invariant measure does not imply high order in the weak or strong sense [2], hence the modified equations and order conditions in the deterministic sense or weak sense are not the same as for the invariant measure sampling, as highlighted in [2, 3, 41, 43, 72].

Consider now a consistent integrator with S-series $S(a)$. Similar to backward error analysis, we are interested in finding a modified vector field $h\widetilde{F} = B(b)$ with $b \in \mathcal{ET}^*$ and $b(\bullet) = 1$ such that $b_c \star a \sim \delta_{\mathbf{1}}$, that is, the integrator applied to the modified equation (1.1.1) with $F = -\nabla V$ replaced by $\widetilde{F}$ is exact. This technique allows in particular to increase the order of a numerical method when the partial derivatives of F are not costly to evaluate (see, for instance, in the deterministic setting [19, 34, 22]). A general expansion of the modified vector field is presented in the $\mathbb{T}^d$ case in [2, 41], but it is not an exotic B-series in general and it is not unique. Following [7], we propose a simple criterion to obtain the existence of a modified vector field in the form of an exotic B-series for which we also provide an explicit expression.

Theorem 3.0.7 (Modified equations)**.** *Consider a consistent method with the exotic S-series $S(a)$ for solving equation* (1.1.1)*. Assume a is a character of $(\mathcal{EF}, \cdot)$. Then, there exists a modified vector field $h\widetilde{F} = B(b)$ that can be written as an exotic B-series with the coefficient map $b\colon \mathcal{ET} \to \mathbb{R}$ satisfying $b(\bullet) = 1$, $b_c \star a \sim \delta_{\mathbf{1}}$, and given by*

$$b = \delta_\bullet - A\Big(\sum_{k=0}^{\infty}(-1)^k A^k_{\tilde{\star}a}(a)\Big).$$

Proof. We introduce the sequence (in the spirit of the works [2, 41])

$$b_n = b_{n-1} - A(b_{n-1,c} \star a), \quad b_0 = \delta_\bullet.$$

Similarly to the proof of Theorem 3.0.5, we show by induction that $b_n(\bullet) = 1$ and we obtain $b_n = -A(a + b_{n-1,c}\tilde{\star}a)$ using the reduced coproduct. The sequence (b_n) takes the following first values

$$\begin{aligned}
b_1 &= \delta_\bullet - A(a),\\
b_2 &= \delta_\bullet - A(a) + A(A(a)\tilde{\star}a),\\
&\vdots\\
b_n &= \delta_\bullet - A\Big(\sum_{k=0}^{n-1}(-1)^k A_{\tilde{\star}a}^k(a)\Big), \quad \text{where } A_{\tilde{\star}a}(x) = A(x)\tilde{\star}a,
\end{aligned}$$

and (b_n) converges to $b = \delta_\bullet - A\big(\sum_{k=0}^{\infty}(-1)^k A_{\tilde{\star}a}^k(a)\big)$ by stationarity. □

Any method of Runge-Kutta type has a coefficient map that is a character, so that Theorem 3.0.7 applies and there exists a modified vector field that can be written as an exotic B-series. See [41, Sec. 5.1] for examples.

3.1 Order conditions for invariant measure sampling

We consider numerical integrators that can be expanded using B-series over grafted trees, for example, integrators of Runge-Kutta type. We recall that for such integrators the differential operators $\mathcal{A}_j$ from Theorem 1.2.5 have the form

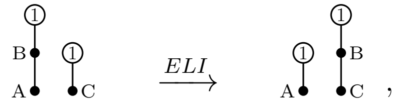
$$\mathcal{A}_j = \sum_{\pi \in EF_j} \frac{a(\pi)}{\sigma(\pi)} \mathbb{F}(\pi)[-],$$

where EF_j is the subset of exotic forests of size j. Thus, Theorem 1.2.5 states that a numerical method $y_0 + B(a)(y_0)$ is of order p with respect to the invariant measure, if

$$\int_{\mathbb{R}^d} S_{<p}(a)[\phi]\rho_\infty dx = (I \circ \delta_{\sigma,<p})(a) = 0,$$

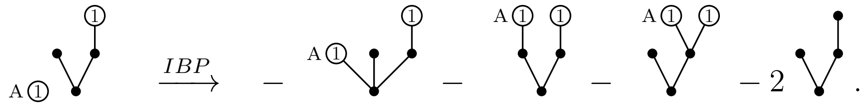
where $I(\pi) = \int_{\mathbb{R}^d} \mathbb{F}(\pi)[\phi]\rho_\infty dx$ and $\delta_{\sigma,<p}(a)$ is the sum over all exotic forests up to size p with coefficients given by $a : EF \to \mathbb{R}$ normalized by σ. We obtain order conditions with respect to the invariant measure by modifying the differential operators that make up $S_{<p}(a)$ in a way that does not change the value of the integral. This translates into two transformations applied to the corresponding exotic forests:

1. Edge-liana inversion (ELI), which moves the liana down the tree along an edge,



   where we note that ELI uses the fact that the exotic forests here are used to denote differential operators, which means that it assumes there is an "invisible" edge starting at the roots,

2. Integration by parts (IBP), which takes a grafted root, connects it to all other vertices with coefficient −1, and adds a term with coefficient −2 in which the grafted root is removed and the paired grafted vertex is colored black, for example,



More details on ELI and IBP can be found in [41] in Section 4.2 and Proposition 4.7. Proposition 3.1.1 allows us to use ELI and IBP to obtain order conditions. Proposition is proven for ELI using the fact that the vector field F of (1.1.1) is the gradient of a potential, that is, $F = -\nabla V$, and for IBP using the integration by parts process on the integral.

Proposition 3.1.1. *[41] Let $\pi_1, \hat{\pi}_1, \pi_2, \hat{\pi}_2 \in \mathcal{EF}$ such that $\pi_1 \xrightarrow{ELI} \hat{\pi}_1$ and $\pi_2 \xrightarrow{IBP} \hat{\pi}_2$, then*

$$I(\pi_1) = I(\hat{\pi}_1) \quad \textit{and} \quad I(\pi_2) = I(\hat{\pi}_2),$$

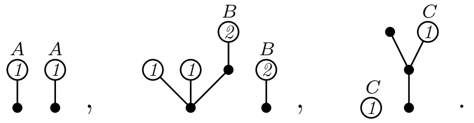
where $\phi \in C_P^\infty$ from the definition of I is a test function.

Definition 3.1.2. Let a connecting liana be a liana $\alpha_e^{-1}(k) = \{v_1, v_2\}$ in π for some $k \in \mathbb{N}$ such that v_1 and v_2 are in different connected components of π and are both leaves.

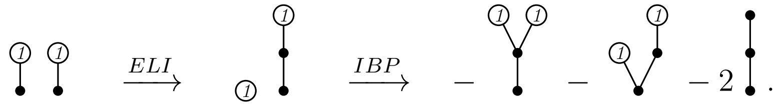
We define the term *connecting liana* and build an algorithm by composing ELI and IBP such that the exotic forests obtained by the algorithm have no connecting lianas.

Definition 3.1.3. Let a connecting liana be a liana $\alpha_e^{-1}(k) = \{v_1, v_2\}$ in π for some $k \in \mathbb{N}$ such that v_1 and v_2 are in different connected components of π.

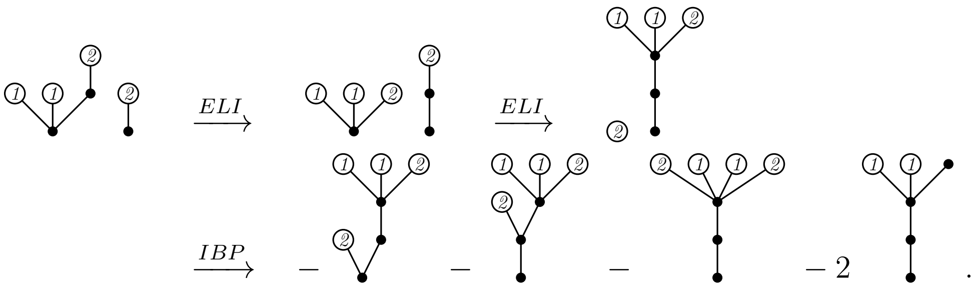
Example 3.1.4. *Connecting lianas are labeled in the exotic forests below:*



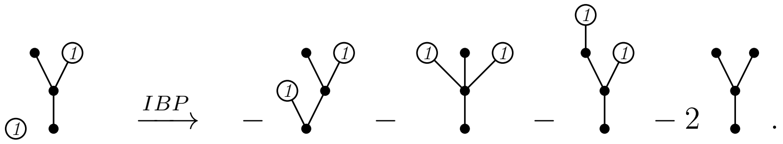
To eliminate the connecting liana A, we first apply ELI to bring one end of the liana to the root level, followed by IBP to remove the connecting liana. The resulting exotic forests are:



To eliminate the connecting liana B, we apply ELI twice, followed by IBP, resulting in:

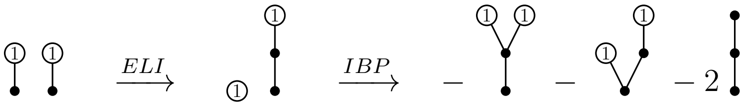


Finally, to eliminate the connecting liana C, we directly apply IBP, yielding:

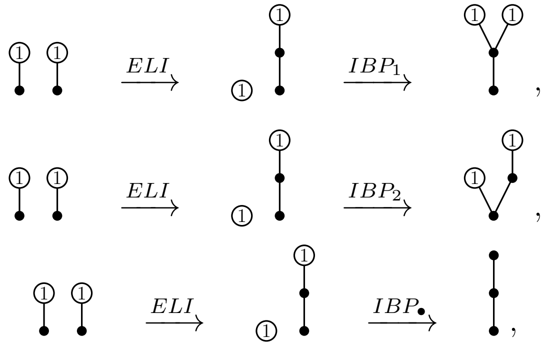


The compositions of ELI and IBP listed in Example 3.1.4 are called transformation chains and denoted by $\pi \to \hat{\pi}$ where $\pi \in EF$ and $\hat{\pi} \in \mathcal{S}(\mathcal{ET})$, where $\mathcal{S}(\mathcal{ET})$ is the vector space of exotic forests without connecting lianas which we write as the symmetric algebra of exotic trees.

To simplify the analysis of the algorithm that we introduce, let us consider the space of labeled exotic forests denoted by $\mathcal{EF}_L$. We use the labeling to split the transformation chains into labeled transformation chains (LTCs) that have labeled exotic forests as terms. This means that the IBP transformation applied to π is split into IBP_v transformations for $v \in V(\pi)$ and $IBP_\bullet$ for the term where the grafted root is removed and the remaining grafted vertex becomes black. For example, the transformation chain



is split into the following labeled transformation chains:



where we exclude the coefficients from the LTC and handle them separately. We note that ELI is not affected.

Let us denote by $\Psi : \mathcal{EF} \to \mathcal{EF}_L$ any injection of $\mathcal{EF}$ into $\mathcal{EF}_L$ which labels the vertices of exotic forests according to some rules. Let $\Phi : \mathcal{EF}_L \to \mathcal{EF}$ be the linear map that forgets the labeling. We note that $\Phi \circ \Psi = \mathrm{id}$ and $\Psi \circ \Phi$ is a relabeling. Let $A : \mathcal{EF} \to \mathcal{S}(\mathcal{ET})$ be a linear map defined as

$$A := \Phi \circ A_L \circ \Psi, \quad \text{where } A_L(\pi) := \sum_{\pi \to \hat{\pi}} C(\pi \to \hat{\pi})\hat{\pi},$$

where the sum is taken over all labeled transformation chains (LTCs) starting at π which are generated recursively by Algorithm 1 and the coefficient $C(\pi \to \hat{\pi})$ is defined as

$$C(\pi \to \hat{\pi}) := (-1)^{|\pi \to \hat{\pi}|_{IBP_v}} (-2)^{|\pi \to \hat{\pi}|_{IBP_\bullet}},$$

where $|\pi \to \hat{\pi}|_{IBP_v}$ is the number of IBP_v transformations for $v \in V(\tilde{\pi})$ for any intermediate $\tilde{\pi}$ and $|\pi \to \hat{\pi}|_{IBP_\bullet}$ is the number of $IBP_\bullet$ transformations.

Let us assume a total order on the vertices of any labeled exotic forest. We require the total order to respect the concatenation product, that is,

$$v_1 \leq v_2 \text{ in } \pi_1 \cdot \pi_2 \quad \text{if} \quad v_1 \leq v_2 \text{ in } \pi_1, \quad \text{for } v_1, v_2 \in V(\pi_1).$$

Such order can be obtained by extending the results from [78]. Let a minimal connecting liana be the connecting liana $\{v_1, v_2\}$ such that v_1 has the shortest path to the root. If there are multiple such lianas, choose the liana with smallest v_1 according to the total order of vertices. If there are multiple lianas with equal v_1, choose the liana with the smallest v_2.

Algorithm 1: Generate the set of all LTCs that start with $\pi_1 \in EF_L$

```
Input: $\pi_1 \in EF_L$
Output: The set of LTCs $\{\pi_1 \to \pi_n\}$ with $\pi_n \in \mathcal{S}(\mathcal{ET})_L$.
Step 1: If $\pi_1$ has no connecting lianas, then return the singleton set $\{\pi_1 \to \pi_1\}$. Else, let $l = \{v_1, v_2\}$ be the minimal connecting liana of $\pi_1$.
Step 2: If the grafted vertex $v_1$ is a root, then let $\{\pi_2^{(i)}\}_{i=1}^N$ with $N = |V(\pi_1)| + 1$ be the set of forests obtained by applying $IBP_v$ and $IBP_\bullet$ to $\pi_1$ with respect to $l$.
Step 3: If $v_1$ is not a root, then let $\{\pi_2\}$ be the singleton set containing the forest obtained by applying ELI moving $l$ towards the root.
Step 4: For each $\pi_2 \in \{\pi_2^{(i)}\}_{i=1}^N$, apply Algorithm 1 to $\pi_2$. Merge all the resulting sets $\{\pi_2^{(i)} \to \pi_n\}$ of LTCs and prepend $\pi_1$ to each LTC. Return the resulting set $\{\pi_1 \to \pi_n\} = \{\pi_1 \to \pi_2^{(i)} \to \pi_n\}$.
```

Proposition 3.1.5. *Algorithm 1 ends in a finite number of steps.*

Proof. The algorithm is guaranteed to end because every application of the IBP decreases the number of roots, which means that IBP can be applied only a finite number of times. The application of ELI does not change the number of roots. ELI moves a liana towards the root which can be done a finite number of times. Note that the minimal connecting liana will stay minimal after the application of ELI. □

Let $\langle \cdot, \cdot \rangle$ be the orthonormal inner product, that is, for $\pi_1, \pi_2 \in EF$, we have

$$\langle \pi_1, \pi_2 \rangle := \begin{cases} 1 & \text{, if } \pi_1 = \pi_2, \\ 0 & \text{, otherwise.} \end{cases}$$

Let $\langle \cdot, \cdot \rangle_\sigma$ be the renormalized inner product, that is, $\langle \pi_1, \pi_2 \rangle_\sigma := \sigma(\pi_1)\langle \pi_1, \pi_2 \rangle$. We note that both inner products are equal on the space of labeled exotic forests $\mathcal{EF}_L$. Due to Algorithm 1, the maps A_L and A are well-defined and we are ready to obtain the order conditions with respect to the invariant measure. The order conditions are denoted by $\omega(\pi) = 0$ with $\pi \in \mathcal{S}(\mathcal{ET})$ where

$$\omega(\pi) := (a \circ A^*)(\pi)\,, \tag{3.1.1}$$

with A^* being the adjoint of A with respect to the inner product $\langle \cdot, \cdot \rangle_\sigma$. Due to Theorem 1.2.5 and Proposition 3.1.1, the conditions $\omega(\pi) = 0$ for all $\pi \in \mathcal{S}(\mathcal{ET}), |\pi| < p$, imply the order p with respect to the invariant measure, since

$$\int_{\mathbb{R}^d} S_{<p}(a)[\phi]\rho_\infty dx = (I \circ \delta_{\sigma,<p})(a) = (I \circ A \circ \delta_{\sigma,<p})(a) = 0.$$

Let $\Delta_\sigma : \mathcal{EF} \to \mathcal{EF} \otimes \mathcal{EF}$ denote the dual of the concatenation product with respect to the inner product $\langle \cdot, \cdot \rangle_\sigma$. The explicit formula for Δ_σ is the following

$$\Delta_\sigma(\frac{\pi}{\sigma(\pi)}) = \sum_{\pi_1 \cdot \pi_2 = \pi} \frac{\pi_1}{\sigma(\pi_1)} \otimes \frac{\pi_2}{\sigma(\pi_2)}.$$

We can see that this formula is true, since,

$$\langle \pi_1 \cdot \pi_2, \pi \rangle_\sigma = \langle \pi_1 \otimes \pi_2, \Delta_\sigma(\pi) \rangle_\sigma = \sigma(\pi), \quad \text{if } \pi_1 \cdot \pi_2 = \pi.$$

Let us also consider the dual of the concatenation product on the space $\mathcal{EF}_L$ of labeled exotic forests, $\Delta : \mathcal{EF}_L \to \mathcal{EF}_L \otimes \mathcal{EF}_L$. We prove Lemma 3.1.6 as an intermediate result.

Lemma 3.1.6. *Let $\pi, \hat{\pi} \in EF_L$ be labeled exotic forests. We define the sets S_1 and S_2 as*

$$S_1 := \{(\pi \to \hat{\pi}, (\hat{\pi}_1, \hat{\pi}_2)) \ : \ \hat{\pi}_1 \cdot \hat{\pi}_2 = \hat{\pi}\},$$
$$S_2 := \{((\pi_1, \pi_2), \pi_1 \to \hat{\pi}_1, \pi_2 \to \hat{\pi}_2) \ : \ \pi_1 \cdot \pi_2 = \pi\},$$

then, $S_1 \cong S_2$.

Proof. Let us take a tuple $(\pi \to \hat{\pi}, (\hat{\pi}_1, \hat{\pi}_2)) \in S_1$. It contains an LTC $\pi \to \hat{\pi}$ and a splitting of $\hat{\pi}$ into $\hat{\pi}_1$ and $\hat{\pi}_2$. Since LTC keeps the labels of vertices when it acts on them, we can split π into π_1 and π_2 by following the labeling of $\hat{\pi}_1$ and $\hat{\pi}_2$. This also gives us a splitting of the LTC $\pi \to \hat{\pi}$ into $\pi_1 \to \hat{\pi}_1$ and $\pi_2 \to \hat{\pi}_2$. That is, we get a tuple $((\pi_1, \pi_2), \pi_1 \to \hat{\pi}_1, \pi_2 \to \hat{\pi}_2)$ which is an element of S_2.

Let us take a tuple $((\pi_1, \pi_2), \pi_1 \to \hat{\pi}_1, \pi_2 \to \hat{\pi}_2) \in S_2$ that contains a splitting of π into π_1 and π_2, and two LTCs $\pi_1 \to \hat{\pi}_1$ and $\pi_2 \to \hat{\pi}_2$. We can combine $\pi_1 \to \hat{\pi}_1$ and $\pi_2 \to \hat{\pi}_2$ by concatenating all intermediate labeled exotic forests into one LTC $\pi \to \hat{\pi}$. This is possible since the total order of vertices respects the concatenation product and the two LTC have distinct labels because π_1 and π_2 are a splitting of one exotic forest. By combining the two LTCs, we also get an exotic forest $\hat{\pi}$ that has $\hat{\pi}_1$ and $\hat{\pi}_2$ as splitting.

This finishes the proof. □

Proposition 3.1.7. *The following identities are true:*

1. $\Delta_\sigma \circ \Phi = (\Phi \otimes \Phi) \circ \Delta$,
2. $\Delta \circ A_L = (A_L \otimes A_L) \circ \Delta$,
3. $\Delta_\sigma \circ A = (A \otimes A) \circ \Delta_\sigma$.

Proof. We first prove identities (1) and (2) and use them to prove identity (3).

Step 1) Let us write the labeling of an exotic forests explicitly as α and take $(\pi, \alpha) \in EF_L$, then, we have

$$((\Phi \otimes \Phi) \circ \Delta)(\pi, \alpha) = \sum_{\substack{\pi_1 \cdot \pi_2 = \pi \\ \alpha_1 \sqcup \alpha_2 = \alpha}} \pi_1 \otimes \pi_2$$

We note the splittings of α as $\alpha_1 \sqcup \alpha_2$ with α_i being the decoration of π_i for $i = 1, 2$. Every splitting of α corresponds to a decorated exotic forest (π, β) with $\beta : V(\pi) \to \{1, 2\}$ such that $\beta^{-1}(1) = \pi_1$ and $\beta^{-1}(2) = \pi_2$, therefore,

$$|\{(\alpha_1, \alpha_2) \ : \ \alpha_1 \sqcup \alpha_2 = \alpha\}| = |\{\beta : V(\pi) \to \{1, 2\} \ : \ \beta^{-1}(i) = \pi_i\}| = |B|.$$

We see that $|B| = p(\pi, \beta, \emptyset)$ and, from Proposition 1.4.11, we know that

$$p(\pi, \beta, \emptyset) = \frac{\sigma(\pi)}{\sigma(\pi, \beta)} = \frac{\sigma(\pi)}{\sigma(\pi_1)\sigma(\pi_2)}.$$

This implies that, using the formula for Δ_σ, we have

$$\begin{aligned}((\Phi \otimes \Phi) \circ \Delta)(\pi, \alpha) &= \sigma(\pi) \sum_{\pi_1 \cdot \pi_2 = \pi} \frac{\pi_1}{\sigma(\pi_1)} \otimes \frac{\pi_2}{\sigma(\pi_2)} \\ &= \sigma(\pi)\Delta_\sigma(\frac{\pi}{\sigma(\pi)}) = (\Delta_\sigma \circ \Phi)(\pi, \alpha).\end{aligned}$$

This proves identity (1).

Step 2) We follow the definition of A_L and use the property $C(\pi \to \hat{\pi}) = C(\pi_1 \to \hat{\pi}_1)C(\pi_2 \to \hat{\pi}_2)$ where the LTC $\pi \to \hat{\pi}$ splits into the LTCs $\pi_1 \to \hat{\pi}_1$ and $\pi_2 \to \hat{\pi}_2$. We start by using the definitions of A_L and Δ and obtain:

$$\Delta \circ A_L = \sum_{\substack{\pi \to \hat{\pi} \\ \hat{\pi}_1 \cdot \hat{\pi}_2 = \hat{\pi}}} C(\pi \to \hat{\pi})(\hat{\pi}_1 \otimes \hat{\pi}_2).$$

Then, we use Lemma 3.1.6 and group the terms to get

$$\Delta \circ A_L = \sum_{\pi_1 \cdot \pi_2 = \pi} \Big(\sum_{\pi_1 \to \hat{\pi}_1} C(\pi_1 \to \hat{\pi}_1)\hat{\pi}_1 \Big) \otimes \Big(\sum_{\pi_2 \to \hat{\pi}_2} C(\pi_2 \to \hat{\pi}_2)\hat{\pi}_2 \Big).$$

We use the definitions of A_L and Δ to conclude the proof of identity (2).

Step 3) We use the definition of A and identites (1) and (2) to show that

$$\Delta_\sigma \circ A = (\Phi \otimes \Phi) \circ (A_L \otimes A_L) \circ \Delta \circ \Psi.$$

We note that the definition of A accepts any injection Ψ, therefore, we can insert a relabeling $\Psi \circ \Phi$ to obtain

$$\Delta_\sigma \circ A = (\Phi \circ \Phi) \circ (A_L \otimes A_L) \circ (\Psi \otimes \Psi) \circ (\Phi \otimes \Phi) \circ \Delta \circ \Psi,$$

which proves identity (3) using the definition of A, identity (1), and the property $\Phi \circ \Psi = \mathrm{id}$. □

Theorem 3.1.8. *Let us apply Algorithm 1 to an exotic S-series $S(a)$ with a being a character of $(\mathcal{EF}, \cdot)$, then, the map ω defined as (3.1.1) is a character of $(\mathcal{EF}, \cdot)$, that is,*

$$\omega(\pi_1 \cdot \pi_2) = \omega(\pi_1)\omega(\pi_2), \quad \text{for } \pi_1, \pi_2 \in \mathcal{EF}. \tag{3.1.2}$$

Proof. We use the definition of ω and Δ_σ to have

$$\omega(\pi_1 \cdot \pi_2) = \langle (A \circ \delta_\sigma)(a), \pi_1 \cdot \pi_2 \rangle_\sigma = \langle (\Delta_\sigma \circ A \circ \delta_\sigma)(a), \pi_1 \otimes \pi_2 \rangle_\sigma$$

let us use identity (2) from Proposition 3.1.7

$$= \langle ((A \otimes A) \circ \Delta_\sigma \circ \delta_\sigma)(a), \pi_1 \otimes \pi_2 \rangle_\sigma$$

we use the explicit formula for Δ_σ to obtain

$$= \langle (A \circ \delta_\sigma)(a), \pi_1 \rangle_\sigma \langle (A \circ \delta_\sigma)(a), \pi_2 \rangle_\sigma = \omega(\pi_1)\omega(\pi_2).$$

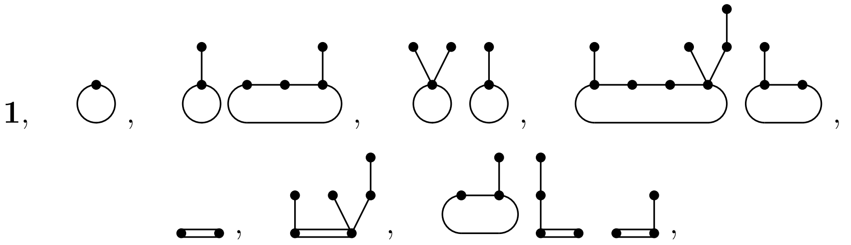
and this finishes the proof. □

We use the fact that the map ω is a character of $(\mathcal{EF}, \cdot)$ to reduce the number of order condtition for invariant measure sampling. We recall that given a numerical integrator $X_1 := \Phi_h(X_0)$ with $\mathbb{E}[\phi(X_1)] = S(a)[\phi]$, it is of order p for the invariant measure sampling if $\omega(\pi) = 0$ for all $\pi \in \mathcal{S}(\mathcal{ET})$ with $\omega := a \circ A^*$ where A^* is the adjoint of the map A with respect to the inner product $\langle -, - \rangle_\sigma$.

Corollary 3.1.9. *Given an integrator satisfying the assumptions discussed in Chapter 1 with $\mathbb{E}[\phi(X_1)] = S(a)[\phi]$, it has order p for the invariant measure sampling if $\omega(\tau) = 0$ for $\tau \in ET$.*

3.2 Composition law

We extend the framework of B-series and S-series by extending the tree and forest formalisms. We consider the sets of aromatic trees $AT := A \times T$ where A is the set of aromas, that is, graphs in which every vertex has exactly one outgoing edge. They were introduced independently in [23] and [37] for the study of volume-preserving integrators. The definition of aromas is extended in [43] to include stolons[1], that is, pairs of trees with linked roots which are used to represent inner products two vector fields. The set A also includes the empty graph and some of its elements are

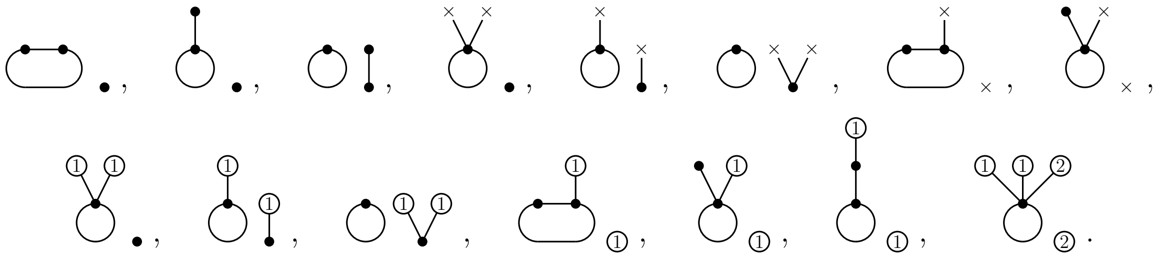


where all the edges are oriented towards the bottom and anticlockwise. The corresponding vector spaces are denoted by $\mathcal{A}$ and $\mathcal{AT}$, respectively. The set of aromatic forests AF is defined as $AF := A \times F$ and includes the empty forest $\mathbf{1}$. The corresponding vector space is denoted by $\mathcal{AF}$.

Aromas play a significant role in geometric numerical integration, serving to represent inner products and the divergence of a vector field, as will be demonstrated in the discussion of the $\mathbb{F}$ map.

We define decorated aromatic forests AF_D, grafted aromatic forests AF_g, and exotic aromatic forests EAF with the corresponding vector spaces $\mathcal{AF}_D, \mathcal{AF}_g, \mathcal{EAF}$ analogously to F_D, F_g, EF.

For example, some grafted aromatic trees and exotic aromatic trees are listed below



We extend the definition of the map $\mathbb{F}$ to include aromas which induces the notion of S-series over aromatic forests following the Definition 3.0.1.

[1] In botany, stolons are horizontal connections that link the base of two plants, allowing a plant to clone itself. Strawberry plants are an example of plants with stolons.

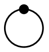
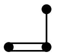
Definition 3.2.1. On aromas ω which have a cycle, choose an edge $e = (u, v)$ of the cycle. Let $\omega \setminus e$ denote the tree obtained by removing the edge e from the aroma ω. Let $\omega \setminus_i e$ denote the tree $\omega \setminus e$ with the vertex v being decorated by i. Let $\mathbb{F}(\overset{i}{\bullet}) := \partial_i f$, then, $\mathbb{F}(\omega)$ is defined as

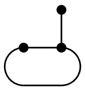
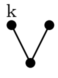
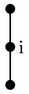
$$\mathbb{F}(\omega) := \sum_{i=1}^{d} \mathbb{F}(\omega \setminus_i e)^i.$$

On armoas given by a stolon ${}_{\tau}{=}_{\gamma}$ where τ and γ are trees, $\mathbb{F}$ is defined as

$$\mathbb{F}({}_{\tau}{=}_{\gamma}) := \langle \mathbb{F}(\tau), \mathbb{F}(\gamma) \rangle = \sum_{i=1}^{d} \mathbb{F}(\tau)^i \mathbb{F}(\gamma)^i.$$

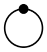
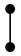
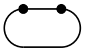
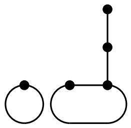
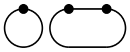
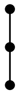
For example,

$\mathbb{F}($  $) = \sum_{i=1}^{d} \partial_i f^i,$ $\mathbb{F}($  $) = \sum_{i,j=1}^{d} f^i f^j \partial_j f^i$

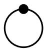
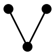
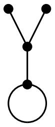
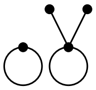
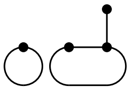
$\mathbb{F}($  $) = \sum_{i=1}^{d} \mathbb{F}($  $)^k = \sum_{i=1}^{d} \mathbb{F}($  $)^i = \sum_{i,j,k=1}^{d} \partial_k f^i f^j \partial_{ij} f^k$

3.2.1 D-algebra of decorated aromatic forests

Let $\curvearrowright: \mathcal{AT}_D \otimes \mathcal{AT}_D \to \mathcal{AT}_D$ denote the grafting product over decorated aromatic trees. The grafting product $\tau \curvearrowright \gamma$ is the sum over all ways to attach the root of τ to a vertex of γ, for example,

  $\curvearrowright$  $\bullet = 2$  $\bullet +$   .

Divergence of an aromatic tree is defined to be a map div : $\mathcal{AT}_D \to \mathcal{A}_D$ such that $\text{div}(\tau)$ is a sum over all ways to attach the root of τ to one of its vertices, for example,

div( ) =  +  + 2  .

Let MAT_D denote the set of marked decorated aromatic trees (v, τ) with $v \in V(\tau)$ being the marked vertex of τ. Let $\mathcal{MAT}_D$ denote the corresponding vector space. Marked decorated aromatic trees $(v, \tau) \in MAT_D$ define $\mathcal{A}_D$-linear endomorphisms $(v, \tau) : \mathcal{AT}_D \to \mathcal{AT}_D$ by $(v, \tau)(\gamma) = \gamma \curvearrowright_v \tau$ where the product $\curvearrowright_v$ attaches the root of the left operand to the vertex v of the right operand. Let the map $d : \mathcal{AT}_D \to \mathcal{MAT}_D$ be an injection defined as

$$d\tau := \sum_{v \in V(\tau)} (v, \tau),$$

let $\curvearrowright^m: \mathcal{AT}_D \otimes \mathcal{MAT}_D \to \mathcal{MAT}_D$ denote the action of $\mathcal{AT}_D$ on $\mathcal{MAT}_D$ defined as

$$\gamma \curvearrowright^m (v, \tau) := (v, \gamma \curvearrowright \tau).$$

Then, let $El_{\mathcal{A}_D}(\mathcal{AT}_D)$ be the algebra of $\mathcal{A}_D$-linear endomorphisms generated by

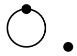
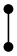
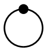
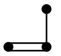
$$\big((\gamma_1 \curvearrowright^m) \circ \cdots \circ (\gamma_n \curvearrowright^m)\big)(d\tau), \quad \text{for } \tau, \gamma_1, \dots, \gamma_n \in \mathcal{AT}_D.$$

The pair $(\mathcal{AT}_D, \mathcal{A}_D)$ is the tracial pre-Lie-Rinehart algebra generated by the set D, that is, it satisfies the following properties:

1. $\mathcal{A}_D$ is a unital commutative algebra with concatenation product $\cdot : \mathcal{A}_D \otimes \mathcal{A}_D \to \mathcal{A}_D$,
2. $\mathcal{AT}_D$ is an $\mathcal{A}_D$-module with a pre-Lie product $\curvearrowright : \mathcal{AT}_D \otimes \mathcal{AT}_D \to \mathcal{AT}_D$, that is,
$$\tau \curvearrowright (\gamma \curvearrowright \nu) - (\tau \curvearrowright \gamma) \curvearrowright \nu = \gamma \curvearrowright (\tau \curvearrowright \nu) - (\gamma \curvearrowright \tau) \curvearrowright \nu,$$
for $\tau, \gamma \in \mathcal{AT}_D, \nu \in \mathcal{AT}_D \sqcup \mathcal{A}_D$,
3. for any $\tau \in \mathcal{AT}_D$, the map $\tau \curvearrowright - : \mathcal{A}_D \to \mathcal{A}_D$ is a derivation and the Leibniz rule holds,
$$\tau \curvearrowright (\omega \cdot \nu) = (\tau \curvearrowright \omega) \cdot \nu + \omega \cdot (\tau \curvearrowright \nu),$$
for $\omega \in \mathcal{A}_D$ and $\nu \in \mathcal{A}_D \sqcup \mathcal{AT}_D$.
4. there exists a map $t : El_{\mathcal{A}_D}(\mathcal{AT}_D) \to \mathcal{A}_D$ called a trace such that $t(\tilde{\tau} \circ \tilde{\gamma}) = t(\tilde{\gamma} \circ \tilde{\tau})$ and $t(\tau \curvearrowright^m \tilde{\gamma}) = \tau \curvearrowright t(\tilde{\gamma})$ with $\tau \in \mathcal{AT}_D, \tilde{\tau}, \tilde{\gamma} \in El_{\mathcal{A}_D}(\mathcal{AT}_D)$ and $\circ$ denotes the composition of endomorphisms. The divergence is then defined as $\operatorname{div}(\tau) := t(d\tau)$.

More details can be found in [31] where it is proven that $(\mathcal{AT}_D, \mathcal{A}_D)$ is a free tracial pre-Lie-Rinehart algebra. We extend the structure of the tracial pre-Lie-Rinehart algebra by considering a symmetric $\mathcal{A}_D$-bilinear form $\langle -, - \rangle : \mathcal{AT}_D \otimes_{\mathcal{A}_D} \mathcal{AT}_D \to \mathcal{A}_D$ with the Leibniz rule:

$$\tau \curvearrowright \langle \gamma, \nu \rangle = \langle \tau \curvearrowright \gamma, \nu \rangle + \langle \gamma, \tau \curvearrowright \nu \rangle, \quad \text{for } \tau, \gamma, \nu \in \mathcal{AT}_D.$$

The aroma $\langle \gamma, \nu \rangle \in \mathcal{A}_D$ is called a stolon and is denoted by a horizontal double edge that connects the roots of the corresponding trees, for example, $\langle$ ,  $\rangle =$  . The resulting algebra $(\mathcal{AT}_D, \mathcal{A}_D)$ is called a *tracial stolonic pre-Lie-Rinehart algebra.*

Definition 3.2.2. Let (A_1, R_1) and (A_2, R_2) be tracial stolonic pre-Lie-Rinehart algebras. A map $\varphi : (A_1, R_1) \to (A_2, R_2)$ is a tracial stolonic pre-Lie-Rinehart homomorphism if

$$\begin{aligned} \varphi(x \curvearrowright y) &= \varphi(x) \curvearrowright \varphi(y), \quad \text{for } x, y \in A_1, \\ \varphi(\langle x, y \rangle) &= \langle \varphi(x), \varphi(y) \rangle, \\ t(\varphi(\tilde{x})) &= \varphi(t(\tilde{x})), \quad \text{for } \tilde{x} \in El_{R_1}(A_1). \end{aligned}$$

Proposition 3.2.3. *The tracial stolonic pre-Lie-Rinehart algebra $(\mathcal{AT}_D, \mathcal{A}_D)$ is a free tracial stolonic pre-Lie-Rinehart algebra. That is, given a tracial stolonic pre-Lie-Rinehart algebra (A, R) and a set map $D \to A$, it can be extended uniquely to a tracial stolonic pre-Lie-Rinehart homomorphism $\varphi : (\mathcal{AT}_D, \mathcal{A}_D) \to (A, R)$.*

Proof. Let (A, R) be a tracial stolonic pre-Lie-Rinehart algebra. The pair $(\mathcal{AT}_D, \mathcal{A}_D)$ is a free tracial pre-Lie-Rinehart algebra as is shown in [31], therefore, given a map $D \to A$ from the set of generators D, it can be extended uniquely to a tracial pre-Lie-Rinehart homomorphism $\varphi : (\mathcal{AT}_D, \mathcal{A}_D) \to (A, R)$. It remains to show that the map

φ can be defined uniquely over the stolons forming a tracial stolonic pre-Lie-Rinehart homomorphism.

The space $\mathcal{AT}_D\hat{\otimes}_{\mathcal{A}_D}\mathcal{AT}_D$ is the commutative tensor product over $\mathcal{A}_D$ defined using the $\mathcal{A}_D$-module structure of $\mathcal{AT}_D$. We note that $\langle -,-\rangle : \mathcal{AT}_D\hat{\otimes}_{\mathcal{A}_D}\mathcal{AT}_D \to \mathcal{A}_D$ is injective and covers all stolons in $\mathcal{A}_D$. Therefore, we define $\varphi(\omega)$ with ω being a stolon as

$$\varphi(\omega) := \langle -,-\rangle_R \circ (\varphi\otimes\varphi)\circ\langle -,-\rangle^{-1},$$

where $\langle -,-\rangle_R : A\hat{\otimes}_R A \to R$ is the inner product of the tracial stolonic pre-Lie-Rinehart algebra (A,R). The map φ is the unique tracial stolonic pre-Lie-Rinehart homomorphism extending a given map $D\to A$ and the statement is proved. □

The classical Guin-Oudom process [61] extends uniquely a pre-Lie product over a vector space V to a product over the symmetric algebra $\mathcal{S}(V)$. We generalize and use this process to extend uniquely the pre-Lie-Rinehart product $\curvearrowright$ over a $\mathcal{A}_D$-module $\mathcal{AT}_D$ to the symmetric algebra $\mathcal{AF}_D := \mathcal{S}_{\mathcal{A}_D}(\mathcal{AT}_D)$ of decorated aromatic forests.

Proposition 3.2.4. *There exists a unique extension of the $\curvearrowright$ product to $\mathcal{AF}_D$ such that*

(i) $\omega\mathbf{1}\curvearrowright\nu = \omega\nu,$ *for* $\omega\in\mathcal{A}_D, \nu\in\mathcal{A}_D\sqcup\mathcal{AT}_D$,

(ii) $(\tau\cdot\pi)\curvearrowright\nu = \tau\curvearrowright(\pi\curvearrowright\nu)-(\tau\curvearrowright\pi)\curvearrowright\nu,$ *for* $\tau\in\mathcal{AT}_D, \pi\in\mathcal{AF}_D$,

(iii) $\pi\curvearrowright(\mu_1\cdot\mu_2) = \sum_{(\pi)}(\pi_{(1)}\curvearrowright\mu_1)\cdot(\pi_{(2)}\curvearrowright\mu_2),$ *for* $\mu_1,\mu_2\in\mathcal{AF}_D$,

with deshuffle coproduct $\Delta_{\mathcal{A}_D}(\pi)=\sum_{(\pi)}\pi_{(1)}\otimes_{\mathcal{A}_D}\pi_{(2)}$.

We follow the structure of the proof of Proposition 2.7 of [61] which proves an analogous statement for $\mathcal{S}(\mathcal{AT}_D)$. We check that the relations $(i),(ii),(iii)$ are well-defined over $S_{\mathcal{A}_D}(\mathcal{AT}_D)$.

Proof. It follows from (i), (iii), and the coassociativity of $\Delta_{\mathcal{A}_D}$ that

$$\tau\curvearrowright\mathbf{1}=0, \quad \text{and} \quad \tau\curvearrowright(\pi_1\cdots\pi_n)=\sum_{k=1}^{n}\pi_1\cdots(\tau\curvearrowright\pi_k)\cdots\pi_n.$$

The relation (ii) is well-defined with respect to the choice of τ using the Lemma 2.5 of [61] which is based on induction on the length of the monomial and the pre-Lie relation. This means that $(\tau\cdot\pi)\curvearrowright\nu$ is well-defined for $\tau\cdot\pi\in\mathcal{S}(\mathcal{AT}_D)$ using (i) and (ii). Let $\mathcal{J}$ be an ideal of $\mathcal{S}(\mathcal{AT}_D)$

$$\mathcal{J}:=\langle(\omega\tau\cdot\gamma-\tau\cdot\omega\gamma)\cdot\pi : \omega\in\mathcal{A}_D, \tau,\gamma\in\mathcal{AT}_D, \pi\in\mathcal{S}(\mathcal{AT}_D)\rangle.$$

It remains to show that $\mathcal{J}\curvearrowright\nu=0$ which follows from the property $(\omega\tau\cdot\pi)\curvearrowright\nu = \omega(\tau\cdot\pi\curvearrowright\nu)$, with $\omega\in\mathcal{A}_D$, proved by induction on the length of the monomial π. The initial step is shown below for $\gamma\in\mathcal{AT}_D$ using the $\mathcal{A}_D$-linearity in the left operand of $\curvearrowright$:

$$(\omega\tau\cdot\gamma)\curvearrowright\nu=\omega\tau\curvearrowright(\gamma\curvearrowright\nu)-(\omega\tau\curvearrowright\gamma)\curvearrowright\nu=\omega(\tau\cdot\gamma\curvearrowright\nu).$$

Assume the property to be true for monomials shorter than $\pi = \tau_1 \cdots \tau_n \in \mathcal{S}(\mathcal{AT}_D)$ and recall that

$$\omega\tau \curvearrowright \pi = \sum_{k=1}^{n} \omega(\tau \curvearrowright \tau_k) \cdot \pi_{\hat{k}}, \quad \text{where } \pi_{\hat{k}} := \tau_1 \cdots \tau_{k-1}\tau_{k+1} \cdots \tau_n.$$

Then, by induction, we have,

$$\begin{aligned}(\omega\tau \curvearrowright \pi) \curvearrowright \nu &= \big(\sum_{k=1}^{n} \omega(\tau \curvearrowright \tau_k) \cdot \pi_{\hat{k}}\big) \curvearrowright \nu \\ &= \omega\big(\sum_{k=1}^{n} (\tau \curvearrowright \tau_k) \cdot \pi_{\hat{k}}\big) \curvearrowright \nu \\ &= \omega((\tau \curvearrowright \pi) \curvearrowright \nu).\end{aligned}$$

This allows us to prove the inductive step:

$$(\omega\tau \cdot \pi) \curvearrowright \nu = \omega\tau \curvearrowright (\pi \curvearrowright \nu) - (\omega\tau \curvearrowright \pi) \curvearrowright \nu = \omega(\tau \cdot \pi \curvearrowright \nu).$$

Therefore, relations (i) and (ii) extend $\curvearrowright$ to $S_{\mathcal{A}_D}(\mathcal{AT}_D) \otimes (\mathcal{A}_D \oplus \mathcal{AT}_D) \to (\mathcal{A}_D \oplus \mathcal{AT}_D)$. Due to the Leibniz rule, the property $(\omega\pi) \curvearrowright \nu = \omega(\pi \curvearrowright \nu)$, the cocommutativity and coassociativity of $\Delta_{\mathcal{A}_D}$, the relation (iii) is well-defined. Therefore, it defines $\curvearrowright$ on $\mathcal{AF}_D = S_{\mathcal{A}_D}(\mathcal{AT}_D)$. □

Definition 3.2.5. Let $(A, \cdot)$ be a unital commutative graded algebra with unit $\mathbf{1}$. Let A be equipped with a non-associative product $\curvearrowright$ and let $x \in A_1$ and $a, b \in A$ satisfy the relation

$$x \curvearrowright (a \cdot b) = (x \curvearrowright a) \cdot b + a \cdot (x \curvearrowright b).$$

The triple $(A, \cdot, \curvearrowright)$ is a commutative D-algebra if the following identities are satisfied

$$\begin{aligned}\mathbf{1} \curvearrowright a &= a, \\ a \curvearrowright x &\in A_1, \\ (\omega \cdot a) \curvearrowright b &= \omega \cdot (a \curvearrowright b), \\ (x \cdot a) \curvearrowright b &= x \curvearrowright (a \curvearrowright b) - (x \curvearrowright a) \curvearrowright b,\end{aligned}$$

for $\omega \in A_0,\ x \in A_1,\ a, b \in A$. It is called tracial if there exists a trace $t : El_{A_0}(A_1) \to A_0$.

For a tracial commutative D-algebra A, the pair (A_1, A_0) is a tracial pre-Lie-Rinehart algebra. This implies that $(\mathcal{AF}_D, \curvearrowright)$ is a free tracial commutative D-algebra due to the fact that $(\mathcal{AT}_D, \mathcal{A}_D)$ is a free tracial pre-Lie-Rinehart algebra and $(\mathcal{AF}_D, \curvearrowright)$ is obtained uniquely using the Guin-Oudom process (Proposition 3.2.4). We extend the structure of the D-algebra $(\mathcal{AF}_D, \curvearrowright)$ with the $\mathcal{A}_D$-bilinear form $\langle -, - \rangle : \mathcal{AT}_D \otimes_{\mathcal{A}_D} \mathcal{AT}_D \to \mathcal{A}_D$ and see that it is free using Proposition 3.2.3.

A map $\varphi : A \to A'$ between two tracial commutative D-algebras A and A' is a D-algebra morphism if $\varphi(A_1) \subset A'_1$ and

$$\begin{aligned}\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b), &\quad \varphi(a \curvearrowright b) = \varphi(a) \curvearrowright \varphi(b), \\ \varphi(\langle x, y \rangle) = \langle \varphi(x), \varphi(y) \rangle, &\quad \varphi(t(\tilde{x})) = t(\varphi(\tilde{x})),\end{aligned}$$

for $a, b \in A$, $x, y \in A_1$, $\tilde{x} \in El_{A_0}(A_1)$.

Example 3.2.6. *Let $\mathcal{X}$ be the space of vector fields $\mathbb{R}^d \to \mathbb{R}^d$. The symmetric algebra $S_{\mathcal{C}^\infty(\mathbb{R}^d)}(\mathcal{X})$ over the ring of $\mathcal{C}^\infty(\mathbb{R}^d)$ maps is a commutative tracial D-algebra and represents the space of differential operators in $\mathbb{R}^d$. The non-associative product is given by the directional derivation, for example, let $f, g, h : \mathbb{R}^d \to \mathbb{R}^d$, then,*

$$(fg)[h] = \sum_{i,j=1}^{d} f^i g^j h_{ij}, \quad \textit{where } h_{ij} := \frac{\partial^2 h}{\partial x_i \partial x_j}.$$

Divergence of a vector field is $\operatorname{div}(f) = \sum_{i=1}^{d} f^i_i$ *and the bilinear product is the inner product, i.e.,* $\langle f, g\rangle = \sum_{i=1}^{d} f^i g^i$.

3.2.2 Grossman-Larson Hopf algebroid

We observe that the algebra of endomorphisms generated by $\pi \curvearrowright - : \mathcal{AF}_D \to \mathcal{AF}_D$ for $\pi \in \mathcal{AF}_D$ is isomorphic to the algebra $(\mathcal{AF}_D, \diamond)$, where $\diamond$ denotes the Grossman-Larson product. This product is defined by the property

$$\pi_1 \curvearrowright (\pi_2 \curvearrowright -) = (\pi_1 \diamond \pi_2) \curvearrowright -, \quad \pi_1, \pi_2 \in \mathcal{AF}_D, \tag{3.2.1}$$

consistent with the definition introduced in Chapter 1. Furthermore, this formulation of the Grossman-Larson product implies its associativity.

We also note that the algebra $(\mathcal{AF}_D, \diamond)$ is defined over $\mathbb{R}$, whereas the coalgebra $(\mathcal{AF}_D, \Delta_{\mathcal{A}_D})$ is defined over $\mathcal{A}_D$. This distinction implies that the algebraic and coalgebraic structures do not form a bialgebra but rather a bialgebroid. The concept of a bialgebroid has several non-equivalent definitions, often referred to by different names. In this discussion, we focus on the definitions provided in [6, 49, 54].

In [6], notions of left and right bialgebroids are introduced. The concept of a bialgebroid presented in [54], referred to as an $\mathcal{A}/\mathbb{R}$-bialgebra, corresponds to the left bialgebroid from [6] and appears to be a special case of the bialgebroid defined in [49]. The key difference lies in the requirement in [54] and [6] that the image of the coproduct resides in a subspace of $\mathcal{AF}_D \otimes_{\mathcal{A}_D} \mathcal{AF}_D$ where component-wise multiplication is well-defined.

Definition 3.2.7 (Left bialgebroid)**.** Let R and A be two associative algebras with R being commutative. Let $\iota : R \to A$ be an algebra inclusion which induces a bimodule structure, $r \cdot a \cdot r' := \iota(rr')a$, which we use to define the tensor product of A with itself over R, $A \otimes_R A$. A is endowed with a coalgebra structure over R with counit $\epsilon : A \to R$ and the coproduct $\Delta : A \to A \otimes_R A$ which corestricts to the subspace

$$A\,{}_R{\times} A := \{\sum_i a_i \otimes_R a_i' \;:\; \sum_i a_i\iota(r) \otimes_R a_i' = \sum_i a_i \otimes_R a_i'\iota(r), \quad \text{for } r \in R\},$$

and is an algebra homomorphism, that is, $\Delta(aa') = \Delta(a)\Delta(a')$. Counit ϵ satisfies the property $\epsilon(aa') = \epsilon(a(\iota \circ \epsilon)(a'))$. Then, A is a left bialgebroid over R.

We note that the component-wise multiplication is not well-defined on $A \otimes_R A$, however, it is well-defined on a subspace which, for all $\sum_i a_i \otimes_R a_i'$, $\sum_j b_j \otimes b_j' \in A \otimes_R A$ and $r \in R$, satisfies

$$\sum_{i,j} a_i\iota(r)b_j \otimes_R a_i'b_j' = a_ib_j \otimes_R a_i'\iota(r)b_j', \tag{3.2.2}$$

Using the associativity of the product, the requirement (3.2.2) is reduced to

$$\sum_i a_i \iota(r) \otimes_R a_i' = \sum_i a_i \otimes_R a_i' \iota(r),$$

found in the definition of $A\,{}_R{\times} A$. We also note, that the counit ϵ is not an algebra homomorphism.

Definition 3.2.8 (Right bialgebroid)**.** Let R and A be two associative algebras with R being commutative. Let $\iota : R \to A$ be an inclusion which induces a bimodule structure, $r \cdot a \cdot r' := a\iota(rr')$, which we use to define the tensor product of A with itself over R, $A \otimes_R A$. A is endowed with a coalgebra structure over R with counit $\epsilon : A \to R$ and the coproduct $\Delta : A \to A \otimes_R A$ which corestricts to the subspace

$$A \times_R A := \{\sum_i a_i \otimes_R a_i' \ : \ \sum_i \iota(r) a_i \otimes_R a_i' = \sum_i a_i \otimes_R \iota(r) a_i', \quad \text{for } r \in R\},$$

and is an algebra homomorphism, that is, $\Delta(aa') = \Delta(a)\Delta(a')$. Counit ϵ satisfies the property $\epsilon(aa') = \epsilon((\iota \circ \epsilon)(a)a')$.Then, A is a right bialgebroid over R.

Proposition 3.2.9 demonstrates that $\mathcal{AF}_D$, equipped with the Grossman-Larson product and a deshuffle coproduct over $\mathcal{A}_D$, forms a left bialgebroid over $\mathcal{A}_D$.

Proposition 3.2.9. *Let $(\mathcal{AF}_D, \diamond)$ denote the Grossman-Larson algebra over $\mathbb{R}$, and let $\iota : \mathcal{A}_D \to \mathcal{AF}_D$ be the inclusion map defined by $\iota(\omega) = \mathbf{1}\omega$ for $\omega \in \mathcal{A}_D$. Define $\mathcal{AF}_D$ as a $\mathcal{A}_D$-bimodule via*

$$\omega \cdot \pi \cdot \omega' = \mathbf{1}\omega \diamond \mathbf{1}\omega' \diamond \pi = \omega\omega'\pi.$$

The deshuffle coproduct $\Delta_{\mathcal{A}_D} : \mathcal{AF}_D \to \mathcal{AF}_D \otimes_{\mathcal{A}_D} \mathcal{AF}_D$ corestricts to $\mathcal{AF}\,{}_{\mathcal{A}_D}{\times}\mathcal{AF}$ and is an algebra homomorphism. Furthermore, the counit $\epsilon_{\mathcal{A}_D} : \mathcal{AF}_D \to \mathcal{A}_D$, defined by $\epsilon_{\mathcal{A}_D}(\mathbf{1}\omega) := \omega$ for $\omega \in \mathcal{A}_D$ and 0 elsewhere, satisfies the property

$$\epsilon_{\mathcal{A}_D}(\pi \diamond \pi') = \epsilon_{\mathcal{A}_D}(\pi \diamond (\iota \circ \epsilon_{\mathcal{A}_D})(\pi')).$$

This structure defines a left bialgebroid, referred to as the left Grossman-Larson bialgebroid and denoted by B_{GL}^L.

Proof. Let us show that the deshuffle coproduct corestricts to $\mathcal{AF}\,{}_{\mathcal{A}_D}{\times}\mathcal{AF}$, that is, for $\pi \in \mathcal{AF}_D$, $\omega \in \mathcal{A}_D$, and $\Delta_{\mathcal{A}_D}(\pi) = \sum_{(\pi)} \pi_{(1)} \otimes \pi_{(2)}$, we have

$$\begin{aligned}\sum_{(\pi)} \pi_{(1)} \diamond \omega \otimes_{\mathcal{A}_D} \pi_{(2)} &= \sum_{(\pi)} \pi_{(1)}(\pi_{(2)} \curvearrowright \omega) \otimes_{\mathcal{A}_D} \pi_{(3)} \\ &= \sum_{(\pi)} \pi_{(1)} \otimes_{\mathcal{A}_D} (\pi_{(2)} \curvearrowright \omega)\pi_{(3)} = \sum_{(\pi)} \pi_{(1)} \otimes_{\mathcal{A}_D} \pi_{(2)} \diamond \omega,\end{aligned}$$

where we use the definition of the Grossman-Larson product from Chapter 1, and the coassociativity and cocommutativity of the deshuffle coproduct. The fact that $\Delta_{\mathcal{A}_D}$ is an algebra homomorphism, that is, for $\pi, \pi' \in \mathcal{AF}_D$, we have

$$\sum_{(\pi \diamond \pi')} (\pi \diamond \pi')_{(1)} \otimes_{\mathcal{A}_D} (\pi \diamond \pi')_{(2)} = \sum_{(\pi),(\pi')} \pi_{(1)} \diamond \pi'_{(1)} \otimes_{\mathcal{A}_D} \pi_{(2)} \diamond \pi'_{(2)},$$

is proven using the definition of the Grossman-Larson product (3.2.1) together with the relation (*iii*) from Proposition 3.2.4. To prove the compatibility of $\epsilon_{\mathcal{A}_D}$ with the product, we note that $\epsilon_{\mathcal{A}_D}(\pi_1 \diamond \pi_2)$ is non-zero if and only if $\pi_2 \in \mathcal{A}_D$ and is equal to the aromas obtained by grafting all trees of π_1 onto π_2 in all possible ways. □

Proposition 3.2.9 can be extended to any commutative D-algebra, provided the associativity of the Grossman-Larson product is established similarly to [61, Lemma 2.10]. Moreover, by excluding all aromas (i.e., setting $\mathcal{A}_D = \{\mathbf{1}\}$), the left bialgebroid reduces to a graded connected bialgebra, which is also a Hopf algebra.

Remark 3.2.10. *We say that B^L_{GL} is cocomplete since $B^L_{GL} = \bigcup_{n=0}^{\infty} B^L_{GL,n}$ and is a free $\mathcal{A}_D$-module. An analog of the Cartier-Milnor-Moore theorem is proven in [54] which states that a cocomplete and graded projective left bialgebroid is the universal enveloping Lie-Rinehart algebra of the Lie-Rinehart algebra of its primitive elements. The freeness of the $\mathcal{A}_D$-module B^L_{GL} implies its graded projectiveness, therefore, the left bialgebroid B^L_{GL} is the universal enveloping Lie-Rinehart algebra of the pre-Lie-Rinehart algebra $(\mathcal{AT}_D, \mathcal{A}_D)$.*

Remark 3.2.11. *If we replace the coalgebra structure of B^L_{GL} by the deshuffle coproduct $\Delta : \mathcal{AF}_D \to \mathcal{AF}_D \otimes \mathcal{AF}_D$ and $\epsilon : \mathcal{AF}_D \to \mathbb{R}$, then we get a Grossman-Larson Hopf algebra dual (up to the symmetry coefficients) to the Hopf algebra mentioned in [5, Thm. 4.4].*

The notions of a Hopf algebroid presented in [6] and [49] are different and none seems to be more general than the other, as is discussed in detail in [6, Section 4.6.1]. We show that B^L_{GL} defines a Hopf algebroid as introduced in [49] by defining the antipode S_{GL} which satisfies the necessary assumptions.

Proposition 3.2.12. *Let $S_{GL} : (\mathcal{AF}_D, \diamond) \to (\mathcal{AF}_D, \diamond)$ be the algebra anti-isomorphism defined as*

(i) $S_{GL}(\omega) := \omega, \quad S_{GL}(\omega\pi) := S_{GL}(\pi) \diamond \omega, \quad$ *for* $\omega \in \mathcal{A}_D, \pi \in \mathcal{AF}_D,$

(ii) $S_{GL}(\tau) := -\tau, \quad S_{GL}(\tau\pi) := -S_{GL}(\pi) \diamond \tau - S_{GL}(\tau \curvearrowright \pi), \quad$ *for* $\tau \in \mathcal{T}_D.$

Then, $H_{GL} := (B_{GL}, S_{GL})$ is the Grossman-Larson Hopf algebroid as defined in [49] with S_{GL} being called an antipode and satisfying the following conditions where $\pi \in \mathcal{AF}_D$ and $\Delta_{\mathcal{A}_D}(\pi) = \sum_{(\pi)} \pi_{(1)} \otimes_{\mathcal{A}_D} \pi_{(2)}$,

(1) $\sum_{(\pi)} S_{GL}(\pi_{(1)}) \diamond \pi_{(2)} = \mathbf{1}\epsilon_{\mathcal{A}_D}(S_{GL}(\pi)),$

(2) $\sum_{(\pi)} \hat{\pi}_{(1)} \diamond S_{GL}(\hat{\pi}_{(2)}) = \mathbf{1}\epsilon_{\mathcal{A}_D}(\pi),$ *with* $\gamma(\pi_{(1)} \otimes_{\mathcal{A}_D} \pi_{(2)}) = \hat{\pi}_{(1)} \otimes \hat{\pi}_{(2)},$

where γ is the section of the projection $P : \mathcal{AF}_D \otimes \mathcal{AF}_D \to \mathcal{AF}_D \otimes_{\mathcal{A}_D} \mathcal{AF}_D$ that places all aromas on the left side of the tensor product, that is, $\hat{\pi}_{(2)}$ has no aromas. Moreover, $S^2_{GL} = id$.

Proof. It can be seen that S_{GL} defined this way is an anti-isomorphism due to the associativity of the Grossman-Larson product and the definition which can be rewritten as

$$S_{GL}(\omega \diamond \pi) = S_{GL}(\pi) \diamond S_{GL}(\omega), \quad S_{GL}(\tau \diamond \pi) = S_{GL}(\pi) \diamond S_{GL}(\tau).$$

Let us prove that S_{GL} satisfies the condition (1) and (2). We start with (1) by noting that $S_{GL}(-) \diamond - : \mathcal{AF}_D \otimes_{\mathcal{A}_D} \mathcal{AF}_D \to \mathcal{AF}_D$ is well-defined since

$$S_{GL}(\omega\pi_{(1)}) \diamond \pi_{(2)} = S_{GL}(\pi_{(1)}) \diamond \omega \diamond \pi_{(2)} = S_{GL}(\pi_{(1)}) \diamond \omega\pi_{(2)}.$$

We see that (1) is satisfied for $\pi = \omega \in \mathcal{A}_D$. Let us check that (1) is satisfied for $\pi = \omega\tau \in \mathcal{AT}_D$:

$$S_{GL}(\tau) \diamond \omega + S_{GL}(\mathbf{1}) \diamond \omega\tau = -\tau\omega - \tau \curvearrowright \omega + \omega\tau = S_{GL}(\tau) \curvearrowright \omega = \mathbf{1}\epsilon_{\mathcal{A}_D}(S_{GL}(\omega\tau)),$$

where $\tau \in \mathcal{T}_D$. Let $\tau \in \mathcal{T}_D, \pi \in \mathcal{F}_D, \omega \in \mathcal{A}_D$, then, $\tau\pi\omega = \tau \diamond \pi\omega - \tau \curvearrowright \pi\omega$. We use induction on the length of π and assume that the condition (1) is satisfied for $\tau \curvearrowright \pi\omega$. We check that the left-hand side of (1) applied to $\tau \diamond \pi\omega$ is 0:

$$\begin{aligned}\sum_{(\tau\diamond\pi\omega)} S_{GL}((\tau \diamond \pi\omega)_{(1)}) \diamond (\tau \diamond \pi\omega)_{(2)} &= \sum_{(\tau),(\pi\omega)} S_{GL}(\tau_{(1)} \diamond \pi_{(1)}) \diamond \tau_{(2)} \diamond \pi_{(2)}\omega \\ &= \sum_{(\pi\omega)} S_{GL}(\pi_{(1)}) \diamond \big(\sum_{(\tau)} S_{GL}(\tau_{(1)}) \diamond \tau_{(2)}\big) \diamond \pi_{(2)}\omega = 0,\end{aligned}$$

since $\sum_{(\tau)} S_{GL}(\tau_{(1)}) \diamond \tau_{(2)} = 0$. This implies that left-hand side of (1) applied to $\tau\pi\omega$ is

$$\begin{aligned}\sum_{(\tau\pi\omega)} S_{GL}((\tau\pi\omega)_{(1)}) \diamond (\tau\pi\omega)_{(2)} &= -\mathbf{1}\epsilon_{\mathcal{A}_D}(S_{GL}(\tau \curvearrowright \pi\omega)) \\ &= \mathbf{1}\epsilon_{\mathcal{A}_D}(S_{GL}(\tau\pi\omega)) + \mathbf{1}\epsilon_{\mathcal{A}_D}(S_{GL}(\pi\omega) \diamond \tau) \\ &= \mathbf{1}\epsilon_{\mathcal{A}_D}(S_{GL}(\tau\pi\omega)).\end{aligned}$$

This proves that S_{GL} satisfies the condition (1). To prove condition (2), we recall that the Grossman-Larson product is $\mathcal{A}_D$-linear in its left operand and that γ places all aromas on the left side of the tensor product. This implies that both sides of the condition (2) are $\mathcal{A}_D$-linear and the condition is reduced to the analogous condition over the Grossman-Larson Hopf algebra over $\mathcal{F}_D$. This proves that S_{GL} satisfies the condition (2).

We prove the property $S_\diamond^2 = \text{id}$ by induction on the number of trees in $\pi \in \mathcal{AF}_D$. Assume $\pi \in \mathcal{F}_D$, if $\pi = \tau \in \mathcal{T}_D$, then, $S_\diamond^2(\tau) = -S_\diamond(-\tau) = \tau$. Assume the statement is true for all monomials shorter than $\pi = \tau\pi'$ for $\tau \in \mathcal{T}_D$ and $\pi' \in \mathcal{F}_D$, then,

$$S_{GL}^2(\tau\pi') = S_{GL}(-S_{GL}(\pi') \diamond \tau) - \tau \curvearrowright \pi' = \tau \diamond \pi' - \tau \curvearrowright \pi' = \tau\pi'.$$

Let us now consider $\omega\pi \in \mathcal{AF}_D$ for $\omega \in \mathcal{A}_D$ and $\pi \in \mathcal{F}_D$. We have,

$$S_{GL}^2(\omega\pi) = S_{GL}(S_{GL}(\pi) \diamond \omega) = \omega S_{GL}^2(\pi) = \omega\pi,$$

and the statement is proved. □

We now demonstrate that $\mathcal{AF}_D$, equipped with the Grossman-Larson product and an appropriate coalgebra structure, forms a right bialgebroid. Furthermore, we show that the triple $(B_{GL}^L, B_{GL}^R, S_{GL})$ constitutes a Hopf algebroid as defined in [6]. To distinguish between the coalgebra structures of B_{GL}^L and B_{GL}^R, we denote the coproduct and counit of B_{GL}^L by

$$\Delta_{\mathcal{A}_D}^L : \mathcal{AF}_D \to \mathcal{AF}_D \otimes_{\mathcal{A}_D^{\curvearrowright}} \mathcal{AF}_D, \quad \text{and } \epsilon_{\mathcal{A}_D}^L : \mathcal{AF}_D \to \mathcal{A}_D,$$

respectively, and those of B^R_{GL} by

$$\Delta^R_{\mathcal{A}_D} : \mathcal{AF}_D \to \mathcal{AF}_D \otimes_{\mathcal{A}_D^{\curvearrowright}} \mathcal{AF}_D, \quad \text{and } \epsilon^R_{\mathcal{A}_D} : \mathcal{AF}_D \to \mathcal{A}_D.$$

Using the fact that $S^2_{GL} = \text{id}$, established in Proposition 3.2.12, we deduce using [6, Prop. 4.4] that S_{GL} is an anti-isomorphism between the left and right bialgebroids. This gives us a unique way to construct the right bialgebroid B^R_{GL}.

Proposition 3.2.13. *Let $(\mathcal{AF}_D, \diamond)$ be the Grossman-Larson algebra over $\mathbb{R}$ with inclusion $\iota : \mathcal{A}_D \to \mathcal{AF}_D$ defined as $\iota(\omega) = \mathbf{1}\omega$ for $\omega \in \mathcal{A}_D$. The $\mathcal{A}_D$-bimodule structure is defined as $\omega \cdot \pi \cdot \omega' := \pi \diamond \iota(\omega\omega')$ for $\omega, \omega' \in \mathcal{A}_D$ and $\pi \in \mathcal{AF}_D$. Let the coproduct $\Delta^R_{\mathcal{A}_D} : \mathcal{AF}_D \to \mathcal{AF}_D \otimes_{\mathcal{A}_D^{\curvearrowright}} \mathcal{AF}_D$ be defined, for $\pi \in \mathcal{F}_D, \omega \in \mathcal{A}_D$, as*

$$\Delta^R_{\mathcal{A}_D}(\pi) := \Delta^L_{\mathcal{A}_D}(\pi), \quad \Delta^R_{\mathcal{A}_D}(\omega\pi) := \omega\mathbf{1} \diamond \Delta^R_{\mathcal{A}_D}(\pi),$$

where we identify $\mathcal{F}_D \otimes_{\mathcal{A}_D^{\curvearrowright}} \mathcal{F}_D$ and $\mathcal{F}_D \otimes_{\mathcal{A}_D^{\curvearrowright}} \mathcal{F}_D$. Let the counit be defined as

$$\epsilon^R_{\mathcal{A}_D}(\omega\pi) := S_{GL}(\pi) \curvearrowright \omega.$$

Then, this structure defines a right bialgebroid, referred to as the right Grossman-Larson bialgebroid and denoted by B^R_{GL}. Moreover, the triple $(B^L_{GL}, B^R_{GL}, S_{GL})$ forms a Hopf algebroid, that is, the following compatibility conditions are satisfied:

(1) $\epsilon^L_{\mathcal{A}_D} \circ \iota = id_{\mathcal{A}_D}$ *and* $\epsilon^R_{\mathcal{A}_D} \circ \iota = id_{\mathcal{A}_D}$,

(2) $(\Delta^L_{\mathcal{A}_D} \otimes_{\mathcal{A}_D^{\curvearrowright}} id) \circ \Delta^R_{\mathcal{A}_D} = (id \otimes_{\mathcal{A}_D^{\curvearrowright}} \Delta^R_{\mathcal{A}_D}) \circ \Delta^L_{\mathcal{A}_D}$,

(3) $(\Delta^R_{\mathcal{A}_D} \otimes_{\mathcal{A}_D^{\curvearrowright}} id) \circ \Delta^L_{\mathcal{A}_D} = (id \otimes_{\mathcal{A}_D^{\curvearrowright}} \Delta^L_{\mathcal{A}_D}) \circ \Delta^R_{\mathcal{A}_D}$

(4) $S_{GL}(\iota(\omega) \diamond \pi \diamond \iota(\omega')) = \iota(\omega') \diamond S_{GL}(\pi) \diamond \iota(\omega)$,

(5) $\diamond \circ (S_{GL} \otimes_{\mathcal{A}_D^{\curvearrowright}} id) \circ \Delta^L_{\mathcal{A}_D} = \iota \circ \epsilon^R_{\mathcal{A}_D}$ *and* $\diamond \circ (id \otimes_{\mathcal{A}_D^{\curvearrowright}} S_{GL}) \circ \Delta^R_{\mathcal{A}_D} = \iota \circ \epsilon^L_{\mathcal{A}_D}$

Proof. We use the fact that S_{GL} is an anti-isomorphism between the left and right bialgebroids to define the right bialgebroid B^R_{GL} with $\epsilon^R_{\mathcal{A}_D} = \epsilon^L_{\mathcal{A}_D} \circ S_{GL}$ and

$$\Delta^R_{\mathcal{A}_D} := (S_{GL} \otimes S_{GL}) \circ \Delta^L_{\mathcal{A}_D} \circ S_{GL}.$$

The definition of S_{GL} from Proposition 3.2.12 implies

$$\epsilon^R_{\mathcal{A}_D}(\omega\pi) = \epsilon^R_{\mathcal{A}_D}(\omega\mathbf{1} \diamond \pi) = \epsilon^L_{\mathcal{A}_D}(S_{GL}(\pi) \diamond \omega\mathbf{1}) = S_{GL}(\pi) \curvearrowright \omega.$$

It is also straightforward to check that $\Delta^R_{\mathcal{A}_D}(\tau) = \Delta^L_{\mathcal{A}_D}(\tau)$ for $\tau \in \mathcal{T}_D$. We note that $\Delta^L_{\mathcal{A}_D}$ is an algebra homomorphism and S_{GL} is an algebra anti-homomorphism which is applied twice. This implies that $\Delta^R_{\mathcal{A}_D}$ is an algebra homomorphism. This property defines $\Delta^R_{\mathcal{A}_D}$ uniquely as

$$\Delta^R_{\mathcal{A}_D}(\tau\pi) = \Delta^R_{\mathcal{A}_D}(\tau) \diamond \Delta^R_{\mathcal{A}_D}(\pi) - \Delta^R_{\mathcal{A}_D}(\tau \curvearrowright \pi).$$

This proves that $\Delta^R_{\mathcal{A}_D}(\pi) = \Delta^L_{\mathcal{A}_D}(\pi)$ for all $\pi \in \mathcal{F}_D$ where we identify $\mathcal{F}_D \otimes_{\mathcal{A}_D^{\curvearrowright}} \mathcal{F}_D$ and $\mathcal{F}_D \otimes_{\mathcal{A}_D^{\curvearrowright}} \mathcal{F}_D$. Analogously, for $\omega \in \mathcal{A}_D$, we have

$$\Delta^R_{\mathcal{A}_D}(\omega\pi) = \Delta^R_{\mathcal{A}_D}(\omega\mathbf{1} \diamond \pi) = (\mathbf{1} \otimes \omega\mathbf{1}) \diamond \Delta^R_{\mathcal{A}_D}(\pi) = \omega\mathbf{1} \diamond \Delta^R_{\mathcal{A}_D}(\pi).$$

The compatibility conditions of the Hopf algebroid follow by the construction of B^R_{GL} and from Proposition 3.2.12. □

3.2.3 Pre-Hopf algebroid of decorated aromatic forests

The space of decorated aromatic forests with commutative product $\cdot$ and deshuffle coproduct forms a Hopf algebra $H := (\mathcal{AF}_D, \cdot, \mathbf{1}, \Delta_{\mathcal{A}_D}, \epsilon_{\mathcal{A}_D}, S)$. The Hopf algebra H together with the product $\curvearrowright$: $H \otimes H \to H$ forms a pre-Hopf algebroid which is a generalization of the pre-Hopf algebra [47] that satisfies the following conditions for $\pi, \mu, \eta \in H$, a map $\beta : H \to H$, and a section γ of the projection $P : H \otimes H \to H \otimes_{\mathcal{A}_D} H$ with $\gamma(\pi_{(1)} \otimes_{\mathcal{A}_D} \pi_{(2)}) = \hat{\pi}_{(1)} \otimes \hat{\pi}_{(2)}$:

(1) $\pi \curvearrowright (\mu \cdot \eta) = (\pi_{(1)} \curvearrowright \mu) \cdot (\pi_{(2)} \curvearrowright \eta)$,

(2) $\pi \curvearrowright (\mu \curvearrowright \eta) = (\pi_{(1)} \cdot (\pi_{(2)} \curvearrowright \mu)) \curvearrowright \eta$,

(3) $\hat{\pi}_{(1)} \curvearrowright (\beta(\hat{\pi}_{(2)}) \curvearrowright -) = \epsilon_{\mathcal{A}_D}(\pi)\mathrm{id}$,

(4) $\beta(\pi_{(1)}) \curvearrowright (\pi_{(2)} \curvearrowright -) = \epsilon_{\mathcal{A}_D}(\beta(\pi))\mathrm{id}$.

Conditions (1) and (2) follow from the definition of the D-algebra and the conditions (3) and (4) are satisfied for β and γ being the anitipode S_{GL} and γ from Proposition 3.2.12. We present an alternative proof for (8) of [47, Lemma 2.3].

Lemma 3.2.14. *For all $\pi, \mu \in H$, we have $\pi \curvearrowright S(\mu) = S(\pi \curvearrowright \mu)$.*

Proof. Consider $\mu = \tau \in \mathcal{AT}_D$, then $\pi \curvearrowright \tau \in \mathcal{AT}_D$ and $\pi \curvearrowright S(\tau) = -\pi \curvearrowright \tau = S(\pi \curvearrowright \tau)$. Assume the statement is true for all monomials shorter than $\mu = \tau_1 \cdots \tau_n$ for $\tau_i \in \mathcal{AT}_D$, then,

$$\begin{aligned} S(\pi \curvearrowright \tau_1 \cdots \tau_n) &= -\sum_{(\pi)} (\pi_{(1)} \curvearrowright \tau_1) S(\pi_{(2)} \curvearrowright \tau_2 \cdots \tau_n) \\ &= -\sum_{(\pi)} (\pi_{(1)} \curvearrowright \tau_1)(\pi_{(2)} \curvearrowright S(\tau_2 \cdots \tau_n)) \\ &= -\pi \curvearrowright (\tau_1 S(\tau_2 \cdots \tau_n)) = \pi \curvearrowright S(\mu). \end{aligned}$$

This finishes the proof. □

We check that the subadjacent Hopf algebra with antipode $\hat{S}_{GL}$ defined in [47, Thm. 2.4] corresponds to the Grossman-Larson Hopf algebroid by showing that the antipodes coincide.

Lemma 3.2.15. *Let $\hat{S}_{GL}$ be the antipode defined in [47, Thm. 2.4], that is,*

$$\hat{S}_{GL}(\pi) := \sum_{(\pi)} \hat{S}_{GL}(\pi_{(1)}) \curvearrowright S(\pi_{(2)}), \quad \textit{for } \pi \in H.$$

Then, $\hat{S}_{GL} = S_{GL}$ where S_{GL} is defined in Proposition 3.2.12.

Proof. We see that $\hat{S}_{GL}(\omega) = \omega$ and $\hat{S}_{GL}(\tau) = -\tau$ for $\omega \in \mathcal{A}_D$ and $\tau \in \mathcal{T}_D$. We check by induction and using the fact that S_{GL} is a coalgebra homomorphism that $\hat{S}_{GL}(\omega\pi) = S_{GL}(\omega\pi)$ for $\pi \in \mathcal{AF}_D$:

$$\begin{aligned} \hat{S}_{GL}(\omega\pi) &= \sum_{(\pi)} S_{GL}(\pi_{(1)}) \curvearrowright S(\omega\pi_{(2)}) = \sum_{(\pi)^2} (S_{GL}(\pi_{(1)}) \curvearrowright \omega)(S_{GL}(\pi_{(2)}) \curvearrowright S(\pi_{(3)})) \\ &= \sum_{(\pi)} (S_{GL}(\pi_{(1)}) \curvearrowright \omega) S_{GL}(\pi_{(2)}) = S_{GL}(\pi) \diamond \omega. \end{aligned}$$

We use the same properties to show that $\hat{S}_{GL}(\tau\pi) = S_{GL}(\tau\pi)$:

$$\begin{aligned}\hat{S}_{GL}(\tau\pi) &= S_{GL}(\pi_{(1)}) \curvearrowright S(\tau\pi_{(2)}) + S_{GL}(\tau\pi_{(1)}) \curvearrowright S(\pi_{(2)})\\ &= -S_{GL}(\pi_{(1)}) \curvearrowright \tau S(\pi_{(2)}) - (S_{GL}(\pi_{(1)}) \diamond \tau) \curvearrowright S(\pi_{(2)}) - S_{GL}(\tau \curvearrowright \pi_{(1)}) \curvearrowright S(\pi_{(2)})\\ &= -S_{GL}(\pi) \diamond \tau - S_{GL}(\pi_{(1)}) \curvearrowright S(\tau \curvearrowright \pi_{(2)}) - S_{GL}(\tau \curvearrowright \pi_{(1)}) \curvearrowright S(\pi_{(2)})\\ &= -S_{GL}(\pi) \diamond \tau - S_{GL}(\tau \curvearrowright \pi),\end{aligned}$$

where we omit writting the sums to simplify the notation. Therefore, $\hat{S}_{GL} = S_{GL}$ following the definition from Proposition 3.2.12. □

3.2.4 Butcher-Connes-Kreimer coproduct

Let us recall the definition of the inner product $\langle -,-\rangle_\sigma$ on the space of decorated aromatic forests $\mathcal{AF}_D$. Let $\langle -,-\rangle$ be the inner product with respect to which AF_D forms an orthonormal basis of $\mathcal{AF}_D$. Then, $\langle \pi,\eta\rangle_\sigma := \sigma(\pi)\langle \pi,\eta\rangle$. Let us define the Butcher-Connes-Kreimer coproduct that is the adjoint of the Grossman-Larson product with respect to the inner product $\langle -,-\rangle_\sigma$ as is proven in Propostion 3.2.17.

Definition 3.2.16. The Butcher-Connes-Kreimer coproduct on AF_D is defined as

$$\Delta_{BCK}(\pi,\alpha) := \sum_{\pi_0 \subset \pi} (\pi \setminus \pi_0, \alpha|_{\pi\setminus\pi_0}) \otimes (\pi_0, \alpha|_{\pi_0}),$$

where the sum runs over all rooted subforests $\pi_0 \in AF$ of π such that $\pi \setminus \pi_0 \in AF$ and there are no edges going from π_0 to $\pi \setminus \pi_0$ in π.

We prove the relation between the Grossman-Larson product and the Butcher-Connes-Kreimer coproduct.

Proposition 3.2.17. *The Butcher-Connes-Kreimer coproduct Δ_{BCK} over decorated aromatic forests is the adjoint of the Grossman-Larson product $\diamond$ with respect to the inner product $\langle -,-\rangle_\sigma$, that is, $\langle \pi \diamond \eta, \gamma\rangle_\sigma = \langle \pi \otimes \eta, \Delta_{BCK}(\gamma)\rangle_\sigma$.*

Proof. We recall that $\langle \pi,\eta\rangle_\sigma := \sigma(\pi)\langle \pi,\eta\rangle$ where $\langle -,-\rangle$ is the inner product with respect to which the set of decorated aromatic forests AF_D forms an orthonormal basis of $\mathcal{AF}_D$.

We note that an aromatic forest can be decorated by multiple sets by taking their Cartesian product, for example, $(\pi,\alpha,\alpha_{\mathbb{N}}) := (\pi, \alpha\times\alpha_{\mathbb{N}})$ where $\alpha\times\alpha_{\mathbb{N}} : V(\pi) \to D\times\mathbb{N}$. We define a set of labeled decorated aromatic forests AF_{DL} and the corresponding space $\mathcal{AF}_{DL}$. We note that labels, unlike decorations, are required to be in bijection with the vertices. The elements (π,α,α_l) of AF_{DL} are aromatic forests $\pi \in AF$ decorated by $\alpha : V(\pi) \to D$ for some set D and $\alpha_l : V(\pi) \to \mathbb{N}$ such that α_l is an injection.

Let us define the Grossman-Larson product and BCK coproduct on the space $\mathcal{AF}_{DL}$ by considering the space $\mathcal{AF}_{D\mathbb{N}}$ of elements $(\pi,\alpha,\alpha_{\mathbb{N}})$ where $(\pi,\alpha) \in AF_D$ and $\alpha_{\mathbb{N}} : V(\pi) \to \mathbb{N}$. The algebra $(\mathcal{AF}_{DL},\diamond)$ is obtained by taking a quotient of $(\mathcal{AF}_{D\mathbb{N}},\diamond)$ over an appropriate ideal, while the coalgebra $(\mathcal{AF}_{DL},\Delta_{BCK})$ is an appropriate subcoalgebra of $(\mathcal{AF}_{D\mathbb{N}},\Delta_{BCK})$. We note that the Grossman-Larson product and the BCK coproduct on the space $\mathcal{AF}_{DL}$ are adjoint to each other with respect to the inner product $\langle -,-\rangle$.

Let us denote by $\widetilde{\mathcal{AF}}_{DL}$ the space of formal sums of the form $\sum_{\pi\in AF_{DL}} a(\pi)\pi$ with $a \in \mathcal{AF}^*_{DL}$. Let $\varphi : (\mathcal{AF}_D, \diamond) \to (\widetilde{\mathcal{AF}}_{DL}, \diamond)$ be an algebra injection and $\hat{\varphi} : (\mathcal{AF}_{DL}, \Delta_{BCK}) \to (\mathcal{AF}_D, \Delta_{BCK})$ be a coalgebra surjection defined as

$$\varphi(\pi,\alpha) := \sum_{\alpha_l} (\pi,\alpha,\alpha_l), \quad \hat{\varphi}(\pi,\alpha,\alpha_l) := (\pi,\alpha),$$

where the sum is over all possible bijections $\alpha_l : V(\pi) \to \mathbb{N}$. We notice that

$$\langle \varphi(\pi,\alpha), (\pi,\alpha,\alpha_l)\rangle = \sigma(\pi,\alpha), \quad \text{and} \quad \langle (\pi,\alpha), \hat{\varphi}(\pi,\alpha,\alpha_l)\rangle = 1,$$

where we use Proposition 1.4.11 to see that $p(\pi,\alpha,\alpha\times\alpha_l) = \sigma(\pi,\alpha)$. This implies that $\varphi(\pi,\alpha,\alpha_l) = \sigma(\pi,\alpha)\hat{\varphi}^*(\pi,\alpha,\alpha_l)$. Let $\pi,\eta,\gamma \in \mathcal{AF}_D$ and $\tilde{\gamma} \in \mathcal{AF}_{DL}$ such that $\gamma = \hat{\varphi}(\tilde{\gamma})$, then, using the definition of $\langle -,-\rangle_\sigma$, we have,

$$\langle \pi\otimes\eta, \Delta_{BCK}(\gamma)\rangle_\sigma = \sigma(\pi)\sigma(\eta)\langle \pi\otimes\eta, \Delta_{BCK}(\hat{\varphi}(\tilde{\gamma}))\rangle$$

we use the fact that $\tilde{\varphi}$ is a coalgebra homomorphism,

$$\begin{aligned}\langle \pi\otimes\eta, \Delta_{BCK}(\gamma)\rangle_\sigma &= \sigma(\pi)\sigma(\eta)\langle \pi\otimes\eta, (\hat{\varphi}\otimes\hat{\varphi})\circ\Delta_{BCK}(\tilde{\gamma})\rangle \\ &= \sigma(\pi)\sigma(\eta)\langle \hat{\varphi}^*(\pi)\otimes\hat{\varphi}^*(\eta), \Delta_{BCK}(\tilde{\gamma})\rangle\end{aligned}$$

we use $\varphi(\pi,\alpha,\alpha_l) = \sigma(\pi,\alpha)\hat{\varphi}^*(\pi,\alpha,\alpha_l)$ as,

$$\langle \pi\otimes\eta, \Delta_{BCK}(\gamma)\rangle_\sigma = \langle \varphi(\pi)\otimes\varphi(\eta), \Delta_{BCK}(\tilde{\gamma})\rangle$$

the BCK coproduct and Grossman-Larson product are adjoint on $\mathcal{AF}_{DL}$,

$$\langle \pi\otimes\eta, \Delta_{BCK}(\gamma)\rangle_\sigma = \langle \varphi(\pi)\diamond\varphi(\eta), \tilde{\gamma}\rangle$$

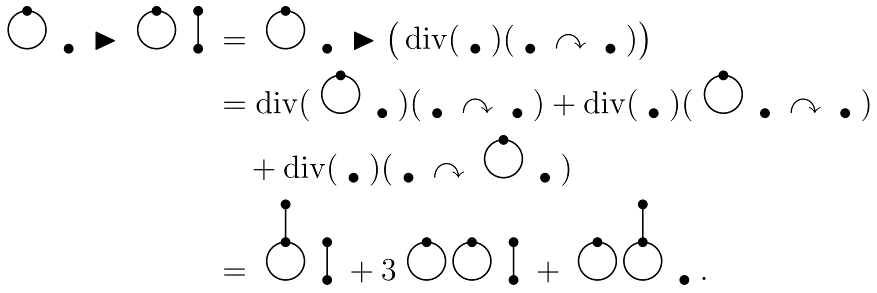
recall that φ is an algebra homomorphism,

$$\langle \pi\otimes\eta, \Delta_{BCK}(\gamma)\rangle_\sigma = \langle \varphi(\pi\diamond\eta), \tilde{\gamma}\rangle = \langle \pi\diamond\eta, \varphi^*(\tilde{\gamma})\rangle = \langle \pi\diamond\eta, \gamma\rangle_\sigma,$$

and the statement is proved. □

3.3 Substitution law

In this section, we introduce the necessary algebroic tools for the description of the substitution law of exotic aromatic S-series. Similarly to Section 3.2 in which we start with the grafting pre-Lie product, in this section, we start with the insertion pre-Lie product ▶: $\mathcal{AT}\otimes\mathcal{AT}\to\mathcal{AT}$ defined by inserting the left operand into a vertex of the right operand in all possible ways, for example,



If for grafting product the extension to decorated aromatic trees is natural and doesn't require additional discussion, the same cannot be said about the insertion product.

3.3.1 Clumped forests

We define decorated clumped forests as a symmetric algebra $\mathcal{CF}_D = \mathcal{S}(\mathcal{AT}_D)$ over $\mathbb{R}$. We use the Guin-Oudom process [61] to define the product $\curvearrowright$ on $\mathcal{CF}_D$. The commutative D-algebra $(\mathcal{CF}_D, \curvearrowright)$ that we obtain in this way has $\mathcal{CF}_{D,0} = \{\mathbf{1}\}$ and is in many ways similar to the commutative D-algebra of classical forests $(\mathcal{F}_D, \curvearrowright)$. Decorated clumped forests have a convenient algebraic structure described in the following result.

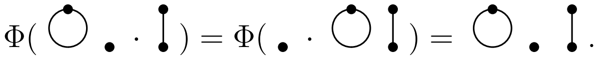
Theorem 3.3.1. *The left Grossman-Larson bialgebroid $(\mathcal{CF}_D, \diamond, \mathbf{1}, \Delta, \epsilon)$ is a Hopf algebra, dual up to the symmetry to the Butcher-Connes-Kreimer Hopf algebra over clumped forests.*

Proof. Since $\mathcal{CF}_{D,0} = \{\mathbf{1}\}$, the left bialgebroid structure reduces to a graded connected bialgebra, that is, to a Hopf algebra. Its duality to the corresponding Butcher-Connes-Kreimer Hopf algebra over clumped forests can be seen by following the proof for classical forests from [36]. □

We recall that the map δ_σ sends a functional $a \in \mathcal{CF}_D^*$ (or $a \in \mathcal{AF}_D^*$) to a formal sum with coefficients given by the functional and renormalized by the symmetry σ, that is, $\sum_\pi \frac{a(\pi)}{\sigma(\pi)}\pi$ for $\pi \in \mathcal{CF}_D$ (or $\pi \in \mathcal{AF}_D$). We note that since the concatenation product and deshuffle coproduct are adjoint with respect to the inner product $\langle -, - \rangle_\sigma$, we have

$$\delta_\sigma(a) \cdot \delta_\sigma(b) = \delta_\sigma(a \odot b) \quad \text{with } a \odot b := m_{\mathbb{R}} \circ (a \otimes b) \circ \Delta. \tag{3.3.1}$$

Let the map $\Phi : (\mathcal{CF}_D, \curvearrowright) \to (\mathcal{AF}_D, \curvearrowright)$ be a commutative D-algebra morphism that "forgets" the clumping, for example,



We define $\Phi^* : \mathcal{AF}_D \to \mathcal{CF}_D$ as

$$(\Phi \circ \delta_\sigma)(a) = \delta_\sigma(a \circ \Phi^*), \quad \text{with } a \in \mathcal{CF}_D^*,$$

in particular, Φ^* is the adjoint of Φ with respect to the $\langle -, - \rangle_\sigma$ inner product. Let us consider the exponential maps

$$\begin{aligned} \exp^{\cdot} &: \mathcal{AT}_D \to \mathcal{AF}_D, \quad \exp^{\odot} : \mathcal{AT}_D^* \to \mathcal{AF}_D^*, \\ \exp_C^{\cdot} &: \mathcal{AT}_D \to \mathcal{CF}_D, \quad \exp_C^{\odot} : \mathcal{AT}_D^* \to \mathcal{CF}_D^*. \end{aligned}$$

Using (3.3.1), we obtain for $a_0 \in \mathcal{AT}_D^*$ the following identities

$$\exp^{\cdot}(\delta_\sigma(a_0)) = \delta_\sigma(\exp^{\odot}(a_0)), \quad \exp_C^{\cdot}(\delta_\sigma(a_0)) = \delta_\sigma(\exp_C^{\odot}(a_0)),$$

with the functionals $\exp^{\odot}(a_0)$ and $\exp_C^{\odot}(a_0)$ being characterized in Propositon 3.3.2.

Proposition 3.3.2. *Let $a_0 \in \mathcal{AT}^*$ be an infinitisimal character and let*

$$a_e := \exp^{\odot}(a_0), \quad a_c := \exp_C^{\odot}(a_0).$$

Then, $a_c \in \mathcal{CF}_D^*$ is a character of $(\mathcal{CF}_D, \cdot)$ and $a_e = a_c \circ \Phi^*$ with $a_e \in \mathcal{AF}_D^*$.

Proof. We prove that $a_c := \exp_C^{\odot}(a_0)$ is a character of $\mathcal{CF}_D$ by considering a decorated clumped forest $\pi = \tau_1 \cdots \tau_n \in \mathcal{CF}_D$ where $\tau_i \in \mathcal{AT}_D$. Then,

$$\begin{aligned}\exp_C^{\odot}(a_0)(\pi) &= \frac{1}{n!}\cdot (a_0 \otimes \cdots \otimes a_0)(\Delta^{n-1}(\pi))\\ &= \frac{1}{n!}\cdot \sum_{\sigma\in S_n} a_0(\tau_{\sigma(1)})\cdots a_0(\tau_{\sigma(n)}) = a_0(\tau_1)\cdots a_0(\tau_n).\end{aligned}$$

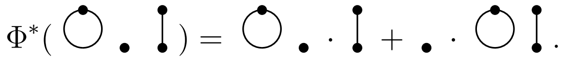
We prove $a_e = a_c \circ \Phi^*$ by using the identity $\exp^{\cdot}(\delta_\sigma(a_0)) = \Phi\big(\exp^{\cdot}_C(\delta_\sigma(a_0))\big)$. □

We define $\mathbb{F}$ over $\mathcal{CF}_D$ by $\mathbb{F} := \mathbb{F} \circ \Phi$ where we use the same notation for the morphism over clumped and aromatic forests. This way, we obtain S-series over decorated clumped forests. We note that, following the definition of Φ^*, any S-series $S(a)$ with $a \in \mathcal{CF}_D^*$ is identical to the S-series over decorated aromatic forests $S(a \circ \Phi^*)$. Moreover, given any functional $a \in \mathcal{AF}_D^*$, there exists a functional $a_C \in \mathcal{CF}_D^*$ such that $a = a_C \circ \Phi^*$, since Φ^* is injective. A possible definition of a_C is

$$a_C(\pi) = \frac{1}{n^m} a(\Phi(\pi)), \quad \text{for } \pi \in \mathcal{CF}_D,$$

where n is the number of rooted components and m is the number of aromas.

Proposition 3.3.3 presents a straightforward way to compute $\Phi^*(\pi)$ for a forest $\pi \in \mathcal{AF}_D$, for example,



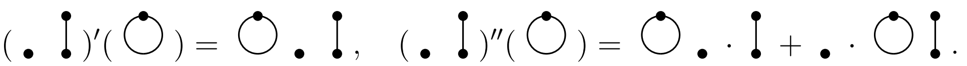
We note that a tree $\tau \in \mathcal{T}_D$ induces a map $\tau' : \mathcal{A}_D \to \mathcal{AT}_D$ with $\tau'(\omega) = \omega\tau$ for $\omega \in \mathcal{A}_D$. We can extend $(-)'$ to $\mathcal{F}_D$ in two possible ways, that is, for a $\pi \in \mathcal{F}_D$, we have two maps,

$$\pi' : \mathcal{A}_D \to \mathcal{AF}_D, \quad \pi'' : \mathcal{A}_D \to \mathcal{CF}_D,$$

defined as

$$(\pi\cdot\mu)'(\omega) = \omega\pi\mu, \quad (\pi\cdot\mu)''(\omega) = (\pi'' \odot \mu'')(\omega),$$

with $\pi'' \odot \mu'' := \cdot \circ (\pi'' \otimes \mu'') \circ \Delta$ where Δ is a $\mathbb{R}$-linear deshuffle coproduct. For example,



The following result presents a convenient method to compute Φ^*.

Proposition 3.3.3. *Given a decorated forest $\pi \in \mathcal{F}_D$ and $\omega \in \mathcal{A}_D$, we have the following identity*

$$\Phi^*(\pi'(\omega)) = \pi''(\omega).$$

Proof. We take the adjoints of π' and π'' with respect to the $\langle -,-\rangle_\sigma$ inner product and denote them by $\pi'^* : \mathcal{AF}_D \to \mathcal{A}_D$ and $\pi''^* : \mathcal{CF}_D \to \mathcal{A}_D$. Then, $\pi'^*(\omega\pi) = \sigma(\pi)\omega$ where $\omega \in \mathcal{A}_D$ and $\pi \in \mathcal{F}_D$. To prove the statement, we have to show that $\pi''^*(\pi_\omega) = \sigma(\pi)\omega$ where $\pi_\omega \in \mathcal{CF}_D$ is a decorated clumped forest that occurs as a term in $\pi''(\omega)$, that is, we need to show that $\pi'^*(\Phi(\pi_\omega)) = \pi''^*(\pi_\omega)$.

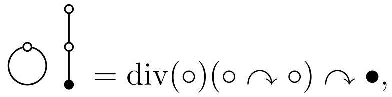
We note that for $\tau \in \mathcal{T}_D$, $\tau''^*(\omega) = \tau'^*(\omega) = \sigma(\tau)\omega$ and the statement is true. We use an inductive assumption and (3.3.1) to obtain

$$(\pi\cdot\mu)''^*(\eta_\omega) = (\cdot \circ (\pi''^* \otimes \mu''^*) \circ \Delta)(\eta_\omega) = \sigma(\pi)\sigma(\mu)|A|\omega,$$

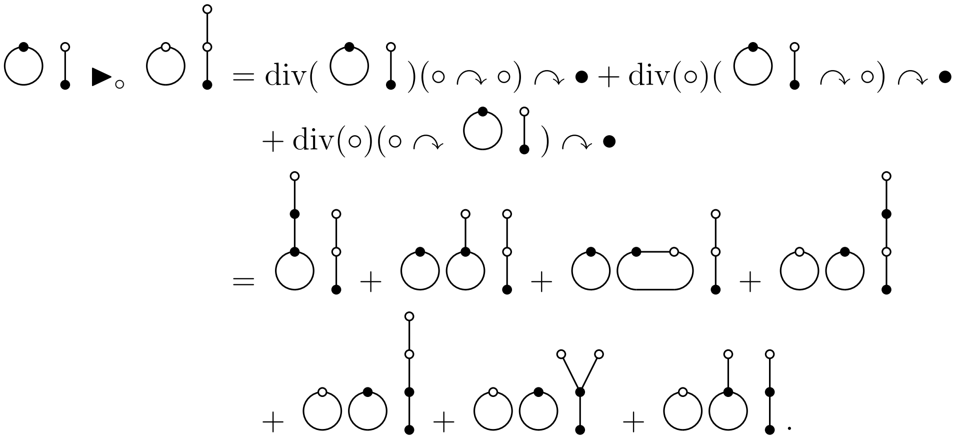
where A is the set of $\eta_{\omega,(1)} \otimes \eta_{\omega,(2)}$ such that the rooted components of $\eta_{\omega,(1)}$ and $\eta_{\omega,(2)}$ are isomorphic to π and μ, respectively. We note that $\sigma(\pi)\sigma(\mu)|A| = \sigma(\pi\cdot\mu)$ and the proof is finished. □

3.3.2 Insertion multi-pre-Lie algebra

We introduce the decorated insertion product $\blacktriangleright_d\colon \mathcal{AT}_D \otimes \mathcal{AT}_D \to \mathcal{AT}_D$ which inserts the aromatic tree from the left operand into the vertices decorated by $d \in D$ of the right operand in all possible ways. For example, we recall that



then,



The product $\blacktriangleright_d$ generalizes the insertion product studied in [70, 51, 69]. The family $(\mathcal{AT}_D, (\blacktriangleright_d)_{d\in D})$ is a multi-pre-Lie algebra [13, 32], that is, for $\tau, \gamma, \nu \in \mathcal{AT}_D$ and $d, e \in D$ we have

$$\tau \blacktriangleright_d (\gamma \blacktriangleright_e \nu) - (\tau \blacktriangleright_d \gamma) \blacktriangleright_e \nu = \gamma \blacktriangleright_e (\tau \blacktriangleright_d \nu) - (\gamma \blacktriangleright_e \tau) \blacktriangleright_d \nu.$$

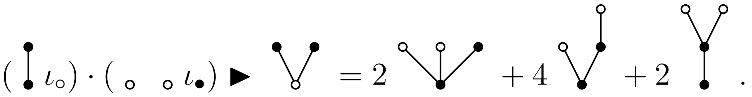
Let $\mathcal{AT}_D^{\oplus D} := \mathcal{AT}_D \otimes \mathbb{R}[D] = \bigoplus_{d\in D} \mathcal{AT}_D \iota_d$ where ι_d for $d \in D$ form the basis of $\mathbb{R}[D]$ and let us define the action $\blacktriangleright\colon \mathcal{AT}_D^{\oplus D} \otimes \mathcal{AT}_D \to \mathcal{AT}_D$ by

$$\tau\iota_d \blacktriangleright \gamma = \tau \blacktriangleright_d \gamma.$$

Let us consider $\mathcal{S}(\mathcal{AT}_D^{\oplus D})$ which becomes $\mathcal{CF}_D^{\otimes D} := \bigotimes_{d\in D} \mathcal{CF}_D \iota_d$ after we assume the identity $\pi\iota_d \cdot \mu\iota_d = (\pi \cdot \mu)\iota_d$. We check that the Guin-Oudom process for multi-pre-Lie products [32, Thm. 2.4] is well-defined and use it to define

$$\blacktriangleright\colon \mathcal{CF}_D^{\otimes D} \otimes \mathcal{CF}_D \to \mathcal{CF}_D.$$

For example, let $D := \{\bullet, \circ\}$, then,



We note that for $\pi, \mu \in \mathcal{CF}_D$, we have $\pi\iota_d \blacktriangleright \mu = 0$ if the number $|\pi|_{AT}$ of aromatic trees in π is greater than the number $|\mu|_d$ of vertices decorated by d in μ. We define a substitution action $\triangleright : \mathcal{CF}_D^{\otimes D} \otimes \mathcal{CF}_D \to \mathcal{CF}_D$ in the following way

$$(\otimes_{d\in D}\pi_d\iota_d) \triangleright \mu := \begin{cases} (\otimes_{d\in D}\pi_d\iota_d) \blacktriangleright \mu, & \text{if } |\pi_d|_{AT} = |\mu|_d \text{ for all } d \in D, \\ 0, & \text{otherwise.} \end{cases}$$

The substitution action substitutes all vertices in the right operand by the aromatic trees from the left operand. We recall that CF_D is a free D-algebra generated by the set D, therefore, given a map $\varphi : D \to \mathcal{AT}_D$, there exists a unique morphism $A_\varphi : \mathcal{CF}_D \to \mathcal{CF}_D$. The morphism A_φ can be written using the substitution action.

Lemma 3.3.4. *Given a morphism $A_\varphi : \mathcal{CF}_D \to \mathcal{CF}_D$ that acts on generators D as φ, we have*

$$A_\varphi(\pi) = \otimes_{d\in D} \exp^{\cdot}_C(\varphi(d)\iota_d) \triangleright \pi.$$

Proof. We use the definition of the exponential map $\exp^{\cdot}_C$ and the substitution action $\triangleright$ to obtain

$$\otimes_{d\in D} \exp^{\cdot}_C(\varphi(d)\iota_d) \triangleright \pi = \frac{1}{\prod_{d\in D} |\pi|_d!}(\otimes_{d\in D}\varphi(d)^{|\pi|_d}\iota_d) \blacktriangleright \pi = A_\varphi(\pi),$$

which proves the statement. □

3.3.3 Substitution law over clumped forests

Let $a^{\otimes D} := \otimes_{d\in D} a_d \iota_d$ with $a_d \in \mathcal{CF}^*_D$ be a functional over $\mathcal{CF}^{\otimes D}_D$ defined as

$$a^{\otimes D}(\otimes_{d\in D}\pi_d\iota_d) = \prod_{d\in D} a_d(\pi_d).$$

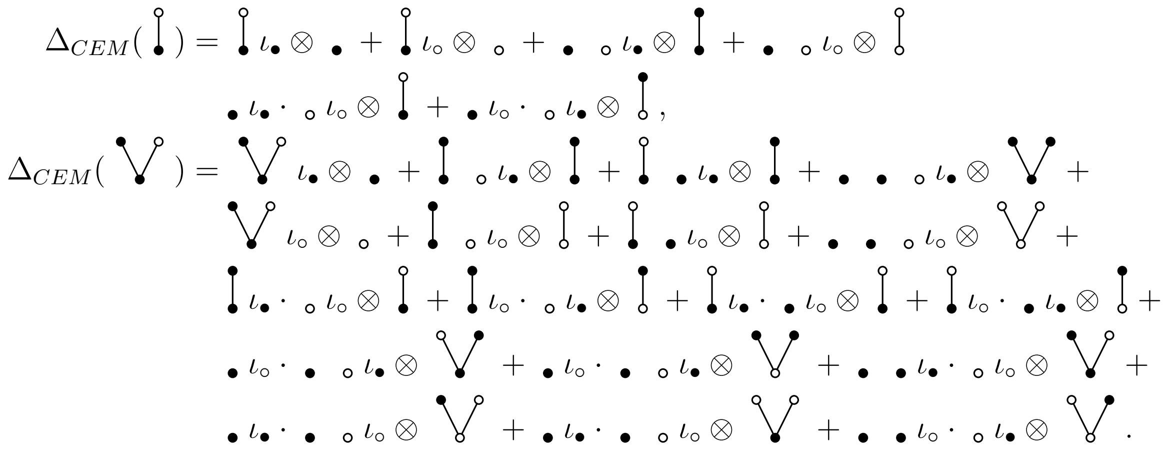
Definition 3.3.5. Define the CEM coaction $\Delta_{CEM} : \mathcal{CF}_D \to \mathcal{CF}^{\otimes D}_D \otimes \mathcal{CF}_D$ as

$$\Delta_{CEM}(\pi) := \sum_{p\in P(\pi)} \sum_{c:p\to D} c^{-1}(D) \otimes \pi/_{(p,c)},$$

where the sum is over the set of partitions $P(\pi)$ of π and the maps $c : p \to D$ which assign a decoration to each element of the partition p. A partition $p \in P(\pi)$ is a set of decorated aromatic subtrees of π that covers all vertices of π. When written in the left operand of the tensor product, $c^{-1}(D)$ is interpreted as a monomial $\otimes_{d\in D} c^{-1}(d)\iota_d$ in $\mathcal{CF}^{\otimes D}_D$. The exotic aromatic forest $\pi/_{(p,c)}$ denotes the forest obtained by contracting each element τ of p into a vertex of color $c(\tau)$.

For example, let $D = \{\bullet, \circ\}$, then,

$$\Delta_{CEM}(\bullet) = \bullet\,\iota_\bullet \otimes \bullet + \bullet\,\iota_\circ \otimes \circ\,,$$



$\Delta_{CEM}($
$) =$
$\iota_\bullet \otimes$ + $\iota_\circ \otimes$ + $\iota_\bullet \otimes$ + $\iota_\circ \otimes$
$\iota_\bullet \cdot$ $\iota_\circ \otimes$ + $\iota_\circ \cdot$ $\iota_\bullet \otimes$,

Proposition 3.3.6. *Let $a \in \mathcal{CF}_D^*$ and $b^{\otimes D} \in \mathcal{CF}_D^{\otimes D*}$, then,*

$$\delta_\sigma(b^{\otimes D} \star a) = \delta_\sigma(b^{\otimes D}) \triangleright \delta_\sigma(a),$$

where $b^{\otimes D} \star a = m_{\mathbb{R}} \circ (b^{\otimes D} \otimes a) \circ \Delta_{CEM}$.

Proof. We prove the statement in two steps. First, we prove the statement over the space of ordered clumped forests. Ordered clumped forest are forests in which all vertices are totally ordered. Next, we describe the relationships between the product and coproduct over ordered and non-ordered clumped forests and use them to finish the proof.

1. The space $\mathcal{CF}_{DO}$ is defined by assigning every decorated clumped forest π a total order over the vertices. We note that a forest $\pi \in \mathcal{CF}_D$ corresponds to $|\pi|!$ ordered forests in $\mathcal{CF}_{DO}$. The symmetry of any element of $\mathcal{CF}_{DO}$ is equal to 1. Let $\mathcal{CF}_{DO}^{\otimes D} := \bigotimes_{d\in D} \mathcal{CF}_{DO} \iota_d$.

We define $\triangleright^O : \mathcal{CF}_{DO}^{\otimes D} \otimes \mathcal{CF}_D \to \mathcal{CF}_{DO}$ and $\Delta_{CEM}^O : \mathcal{CF}_{DO} \to \mathcal{CF}_{DO}^{\otimes D} \otimes \mathcal{CF}_D$ to be the natural extensions of $\triangleright$ and Δ_{CEM}. The map $\triangleright^O$ substitutes the vertices decorated by d by the trees of $\pi_d \iota_d$ and chooses a total order between the vertices of π_d and π_l with $d \neq l$ in all possible ways. We note that there are $\left(\sum_{d\in D} |\pi_d|\right)!/\prod_{d\in D} |\pi_d|!$ ways to choose a total order between the vertices of π_d and π_l with $d \neq l$. Let $\pi \in \mathcal{CF}_{DO}^{\otimes D}$ and $\mu \in \mathcal{CF}_D$, then,

$$\pi \triangleright^O \mu = \sum_{\eta \in \mathcal{CF}_{DO}} N(\pi, \mu, \eta)\eta,$$

where $N(\pi, \mu, \eta)$ is the number of ways to substitute the vertices of μ by trees in π to obtain η. We note that there are $|\mathrm{Aut}(\mu)|$ ways to substitute the vertices of μ to obtain the same ordered forest η, therefore, $N(\pi, \mu, \eta) = \sigma(\mu)$. The symmetry is 1 for an ordered clumped forest, and, since all terms in $\Delta_{CEM}^O(\pi)$ have coefficient 1, we obtain

$$\delta_\sigma^O(b^{\otimes D}) \triangleright^O \delta_\sigma(a) = \delta_\sigma^O(b^{\otimes D} \star^O a), \quad \text{where } b^{\otimes D} \star^O a = m_{\mathbb{R}} \circ (b^{\otimes D} \otimes a) \circ \Delta_{CEM}^O,$$

for $b^{\otimes D} \in \mathcal{CF}_{DO}^{\otimes D*}$ and $a \in \mathcal{CF}_D^*$. The image of δ_σ^O is a formal sum over ordered clumped forests.

2. We define the map φ that forgets the ordering of the vertices and let

$$\hat{\varphi}(\pi) := \frac{\varphi(\pi)}{|\pi|!}, \quad \varphi^{\otimes D}(\otimes_{d\in D} \pi_d \iota_d) = \otimes_{d\in D} \varphi(\pi_d)\iota_d,$$

where $\varphi, \hat{\varphi} : \mathcal{CF}_{DO} \to \mathcal{CF}_D$ and $\varphi^{\otimes D}, \hat{\varphi}^{\otimes D} : \mathcal{CF}_{DO}^{\otimes D} \to \mathcal{CF}_D^{\otimes D}$. We have the following properties

$$\hat{\varphi} \circ \triangleright^O = \triangleright \circ (\hat{\varphi}^{\otimes D} \otimes \mathrm{id}), \quad (\varphi^{\otimes D} \otimes \mathrm{id}) \circ \Delta_{CEM}^O = \Delta_{CEM} \circ \varphi.$$

We also note that for a functional $b^{\otimes D} \in \mathcal{CF}_D^{\otimes D*}$, $\delta_\sigma(b^{\otimes D}) = \hat{\varphi}^{\otimes D}(\delta_\sigma^O(b^{\otimes D} \circ \varphi^{\otimes D}))$, therefore,

$$\begin{aligned}
\delta_\sigma(b^{\otimes D}) \triangleright \delta_\sigma(a) &= \hat{\varphi}^{\otimes D}\big(\delta_\sigma^O(b^{\otimes D} \circ \varphi^{\otimes D})\big) \triangleright \delta_\sigma(a) \\
&= \hat{\varphi}\big(\delta_\sigma^O(b^{\otimes D} \circ \varphi^{\otimes D}) \triangleright^O \delta_\sigma(a)\big) \\
&= \hat{\varphi}\big(\delta_\sigma^O((b^{\otimes D} \circ \varphi^{\otimes D}) \star^O a)\big) \\
&= \hat{\varphi}\big(\delta_\sigma^O((b^{\otimes D} \star a) \circ \varphi)\big) = \delta_\sigma(b^{\otimes D} \star a).
\end{aligned}$$

This proves that δ_σ is an algebra morphism. □

We extend the insertion action $\blacktriangleright\colon \mathcal{AT}_D^{\oplus D} \otimes \mathcal{AT}_D \to \mathcal{AT}_D$ to the insertion pre-Lie product $\blacktriangleright\colon \mathcal{AT}_D^{\oplus D} \otimes \mathcal{AT}_D^{\oplus D} \to \mathcal{AT}_D^{\oplus D}$ by

$$\tau\iota_d \blacktriangleright \gamma\iota_l = (\tau \blacktriangleright_d \gamma)\iota_l.$$

We recall that $\mathcal{CF}_D^{\otimes D} = \mathcal{S}(\mathcal{AT}_D^{\oplus D})$ which allows us to use the Guin-Oudom process and define $\blacktriangleright\colon \mathcal{CF}_D^{\otimes D} \otimes \mathcal{CF}_D^{\otimes D} \to \mathcal{CF}_D^{\otimes D}$ and the substitution product $\rhd : \mathcal{CF}_D^{\otimes D} \otimes \mathcal{CF}_D^{\otimes D} \to \mathcal{CF}_D^{\otimes D}$. The algebra $(\mathcal{CF}_D^{\otimes D}, \rhd)$ is endowed with the deshuffle coproduct Δ of $\mathcal{S}(\mathcal{AT}_D^{\oplus D})$. We present a generalization of the result obtained in [18] for classical forests.

Theorem 3.3.7. *The algebra $B_\rhd := (\mathcal{CF}_D^{\otimes D}, \rhd)$ endowed with the deshuffle coproduct Δ is a bialgebra dual to the CEM bialgebra $B_{CEM} := (\mathcal{CF}_D^{\otimes D}, \cdot, \Delta_{CEM})$ with respect to the $\langle -, - \rangle_\sigma$ inner product. Moreover, $H_{CEM} := B_{CEM}/\langle (1 - \overset{\mathrm{d}}{\bullet}\,)\iota_d \,:\, d \in D\rangle$ is a Hopf algebra dual to the Grossman-Larson Hopf algebra $H_\rhd$, which is an appropriate sub-bialgebra of $B_\rhd$ and a universal enveloping algebra of $(\mathcal{AT}_D^{\oplus D}, \blacktriangleright)$.*

Proof. Let us start by proving that $B_\rhd$ is a bialgebra. The unit is given by $u := \otimes_{d\in D} \exp^{\cdot}_C(\,\overset{\mathrm{d}}{\bullet}\ \iota_d)$ and counit is given by $\epsilon := \mathbf{1}^* \in \mathcal{CF}_D^{\otimes D *}$. It is straightforward to check that the counit is compatible with the product and unit. We use the property $\Delta(\exp^{\cdot}_C(\tau)) = \exp^{\cdot}_C(\tau) \otimes \exp^{\cdot}_C(\tau)$ to check that the unit is compatible with the coproduct, that is, $\Delta(u) = u \otimes u$. We use

$$\pi \rhd (- \cdot -) = \sum_{(\pi)} (\pi_{(1)} \rhd -) \cdot (\pi_{(2)} \rhd -)$$

and the associativity of $\rhd$ to obtain the identity

$$\sum_{(\pi),(\mu)} (\pi_{(1)} \rhd \mu_{(1)} \rhd -)(\pi_{(2)} \rhd \mu_{(2)} \rhd -) = \sum_{(\pi \rhd \mu)} ((\pi \rhd \mu)_{(1)} \rhd -)((\pi \rhd \mu)_{(2)} \rhd -).$$

Therefore, the compatibility of the product and coproduct is necessary. This proves that $B_\rhd$ is a bialgebra. Equation (3.3.1) and Proposition 3.3.6 show that B_{CEM} is the dual bialgebra of $B_\rhd$.

Similarly to [18], we obtain a Hopf algebra once we take the quotient of B_{CEM} by the ideal $\langle (\mathbf{1} - \overset{\mathrm{d}}{\bullet}\,)\iota_d \,:\, d \in D\rangle$. The Hopf algebra H_{CEM} thus obtained is dual to the Hopf algebra $H_\rhd$ which is isomorphic to the Grossman-Larson Hopf algebra obtained using Guin-Oudom process applied to the insertion pre-Lie product $\blacktriangleright$. The elements of $H_\rhd$ have the form

$$\otimes_{d\in D} \exp^{\cdot}_C(\,\overset{\mathrm{d}}{\bullet}\ \iota_d) \cdot \pi, \quad \text{for } \pi \in \mathcal{CF}_D^{\otimes D}.$$

Following Guin-Oudom, Grossman-Larson algebra is a universal enveloping algebra of the respective pre-Lie algebra. □

Theorem 3.3.8 (Substitution law)**.** *Let $a \in \mathcal{CF}_D^*$ and let $\{b_{0,d} \in \mathcal{AT}_D^* \,:\, d \in D\}$ be a set of infinitesimal characters. Let us consider a map $\varphi : D \to \mathcal{AT}_D$ with $\varphi(d) = \delta_\sigma(b_{0,d})$, let $\mathbb{F}(d) = f_d$ for $d \in D$, and let $S_\varphi = \mathbb{F} \circ A_\varphi \circ \delta_\sigma$, then,*

$$S_\varphi(a) = S(b_c^{\otimes D} \star a),$$

where $b_c^{\otimes D} = \otimes_{d\in D} b_{c,d}\iota_d$ with $b_{c,d}$ being a character of $\mathcal{CF}_D$ that extends $b_{0,d}$.

Proof. We use Lemma 3.3.4 and Proposition 3.3.2 to write A_φ as

$$A_\varphi(\pi) = \delta_\sigma(b_c^{\otimes D}) \triangleright \pi, \quad \text{where } b_c^{\otimes D} = \otimes_{d\in D} b_{c,d}\iota_d,$$

with $b_{c,d}$ being a character of $\mathcal{CF}_D$ that extends $b_{0,d}$. We use the fact that $S_\varphi = \mathbb{F}\circ A_\varphi \circ \delta_\sigma$ and Proposition 3.3.6 to obtain

$$S_\varphi(a) = \mathbb{F}\big(\delta_\sigma(b_c^{\otimes D}) \triangleright \delta_\sigma(a)\big) = \mathbb{F}\big(\delta_\sigma(b_c^{\otimes D} \star a)\big) = S(b_c^{\otimes D} \star a).$$

This finishes the proof. □

Analogously to [50], due to the relation between the D-algebra homomorphism $A_\varphi : \mathcal{CF}_D \to \mathcal{CF}_D$ and the corresponding map $b_c^{\otimes D}\star : \mathcal{CF}_D^* \to \mathcal{CF}_D^*$, we obtain Corollary 3.3.9.

Corollary 3.3.9 (Left distributivity)**.** *There is the following distributivity relation between the substitution law and the composition law:*

$$a_c^{\otimes D} \star (b * c) = (a_c^{\otimes D} \star b) * (a_c^{\otimes D} \star c),$$

for $a_c^{\otimes D} = \otimes_{d\in D} a_{c,d}\iota_d$ *where* $a_{c,d}$ *are characters of* $\mathcal{CF}_D$ *and* $b, c \in \mathcal{CF}_D^*$.

Proof. Let $\varphi(d) = \delta_\sigma(a_{0,d})$, then we have

$$\begin{aligned}\delta_\sigma(a_c^{\otimes D} \star (b * c)) &= A_\varphi(\delta_\sigma(b * c))\\ &= A_\varphi(\delta_\sigma(b)) \diamond A_\varphi(\delta_\sigma(c))\\ &= \delta_\sigma((a_c^{\otimes D} \star b) * (a_c^{\otimes D} \star c)).\end{aligned}$$

Since δ_σ is an isomorphism, the statement is proved. □

3.3.4 Substitution law over decorated aromatic forests

We extend the definition of $\blacktriangleright$ to decorated aromatic forests as $\blacktriangleright$: $\mathcal{CF}_D^{\otimes D} \otimes \mathcal{AF}_D \to \mathcal{AF}_D$ and note that Φ is a $\mathcal{CF}_D^{\otimes D}$-module morphism, that is,

$$\Phi(\pi \blacktriangleright \mu) = \pi \blacktriangleright \Phi(\mu).$$

We extend the substitution product $\triangleright : \mathcal{CF}_D^{\otimes D} \otimes \mathcal{AF}_D \to \mathcal{AF}_D$ and $\Delta_{CEM} : \mathcal{AF}_D \to \mathcal{CF}_D^{\otimes D} \otimes \mathcal{AF}_D$ and use Proposition 3.3.6 to see that Φ^* is a $\mathcal{CF}_D^{\otimes D}$-comodule morphism, that is,

$$\Delta_{CEM} \circ \Phi^* = (\mathrm{id} \otimes \Phi^*) \circ \Delta_{CEM}.$$

Using the freeness of the tracial commutative D-algebra $(\mathcal{AF}_D, \curvearrowright)$, we note that for every map $\varphi : D \to \mathcal{AT}_D$, there exists a corresponding $A_\varphi : \mathcal{AF}_D \to \mathcal{AF}_D$ with the following property

$$\Phi \circ A_\varphi = A_\varphi \circ \Phi,$$

where we use the same notation for the A_φ over decorated clumped and aromatic forests. We describe the substitution law for S-series over decorated aromatic forests by using the fact that any functional over decorated aromatic forests $a \in \mathcal{AF}_D^*$ can be written as an image of the map Φ^*, that is, there exists a functional $a_C \in \mathcal{CF}_D^*$ such that $a = a_C \circ \Phi^*$.

Theorem 3.3.10. *Using the notation from Theorem 3.3.8 and given $a \in \mathcal{AF}_D$, we have*

$$S_\varphi(a) = S(b_c^{\otimes D} \star a),$$

with convolution product $\star$ with respect to $\Delta_{CEM} : \mathcal{AF}_D \to \mathcal{CF}_D^{\otimes D} \otimes \mathcal{AF}_D$.

Proof. We note that given a functional $a \in \mathcal{AF}_D^*$, we can define a functional $a_C \in \mathcal{CF}_D^*$ such that $a = a_C \circ \Phi^*$, moreover, $S(a) = S(a_C)$. The statement follows from the following identities,

$$S_\varphi(a) = S_\varphi(a_C) = S(b_c^{\otimes D} \star a_C) = S((b_c^{\otimes D} \star a_C) \circ \Phi^*) = S(b_c^{\otimes D} \star (a_C \circ \Phi^*)) = S(b_c^{\otimes D} \star a),$$

with $b_c^{\otimes D} := \otimes_{d \in D} b_{c,d} \iota_d$. □

Corollary 3.3.9 is extended to functionals over decorated aromatic forests in a similar fashion.

3.4 Exotic aromatic forests

We construct the space $\mathcal{EAF}$ from the space of decorated aromatic forest $\mathcal{AF}_D$ and inherit a number of algebraic structures from it. There are two ways to do that: by taking an appropriate subspace or by taking a quotient. We use the first approach to define the algebras and the second approach to define coalgebras.

Let us consider decorated forests $\mathcal{AF}_D$ with $D = \{\bullet\} \cup \mathbb{N}$ and let $\overline{\mathcal{AF}_D}$ denote its completion with respect to the grading given by the number of vertices, that is, $\overline{\mathcal{AF}_D}$ is a space of formal sums over AF_D. We define the subspace $\mathcal{AF}_D^{\mathcal{E}}$ of $\mathcal{AF}_D$ corresponding to exotic aromatic forests spanned by

$$\varphi(\pi, \alpha_e) := \sum_{\alpha \in P(\alpha_e)} (\pi, \alpha), \tag{3.4.1}$$

with $\varphi : \mathcal{EAF} \to \mathcal{AF}_D^{\mathcal{E}}$ an isomorphism. $P(\alpha_e)$ is the set of decorations α with $\alpha^{-1}(\bullet) = \alpha_e^{-1}(\bullet)$ and

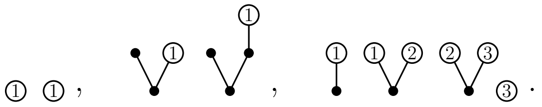
$$\alpha(v_1) = \alpha(v_2) \quad \text{if} \quad \alpha_e(v_1) = \alpha_e(v_2), \quad \text{for } v_1, v_2 \in V(\pi).$$

We note that $\mathcal{AF}_D^{\mathcal{E}}$ forms a subalgebra of $\mathcal{AF}_D$ with respect to the concatenation, grafting, and Grossman-Larson products which induces the corresponding algebraic structures over $\mathcal{EAF}$ through the isomorphism φ.

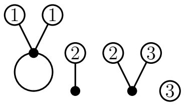
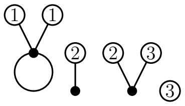
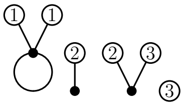
Let us now consider the subspace of $\mathcal{AF}_D$ orthogonal to $\mathcal{AF}_D^{\mathcal{E}}$ with respect to the inner product $\langle -, - \rangle_\sigma$. We denote it by $\mathcal{I}$ and we can see that it is a coideal in the coalgebra $(\mathcal{AF}_D, \Delta_{\mathcal{A}_D})$ using the fact that the deshuffle coproduct is adjoint to the concatenation product. Taking a quotient of $(\mathcal{AF}_D, \Delta_{\mathcal{A}_D})$ by $\mathcal{I}$ gives a definition of the coalgebra of exotic forests $(\mathcal{EF}, \Delta_{\mathcal{EA}})$ through the isomorphism

$$\psi(\pi, \alpha_e) := (\pi, \alpha_e) + \mathcal{I} \quad \in \mathcal{AF}_D/_{\mathcal{I}}. \tag{3.4.2}$$

An exotic aromatic forest is connected if it cannot be written as a concatenation of non-trivial exotic aromatic forests. This notion of connectedness coincides with the one found in [40]. We note that due to the pairings of the number vertices (that are also called *lianas*), exotic aromatic forests can contain connected components which have more than one root, which is a major difference with the standard Butcher trees and forests. For example, the following exotic aromatic forests are connected:



The $\mathcal{EA}$-module of primitive elements $\mathcal{PEAF} := \mathrm{Prim}(\mathcal{EAF}, \Delta_{\mathcal{EA}})$ is spanned by the connected exotic aromatic forests, for example,

$\Delta_{\mathcal{EA}}($  $) =$  $\otimes \mathbf{1} + \mathbf{1} \otimes$  .

To prove the compatibility between the algebraic and coalgebraic structures of $\mathcal{EAF}$, we consider the product $\tilde{\curvearrowright} : \mathcal{EAF} \otimes \mathcal{EAF} \to \mathcal{EAF}$ defined as

$$\pi \tilde{\curvearrowright} \eta = \pi \diamond \eta - \pi \cdot \eta, \quad \text{for } \pi \in \mathcal{PEAF}, \eta \in \mathcal{EAF},$$
$$(\pi \cdot \gamma) \tilde{\curvearrowright} \eta = \pi \tilde{\curvearrowright} (\gamma \tilde{\curvearrowright} \eta) - (\pi \tilde{\curvearrowright} \gamma) \tilde{\curvearrowright} \eta, \quad \text{for } \gamma \in \mathcal{EAF}.$$

Intuitively, the product $\tilde{\curvearrowright}$ is a modification of the Grossman-Larson product $\diamond$ in which we require each element of $\mathcal{PEAF}$ in the left operand to attach at least one root to a vertex of the right operand. We note the following property

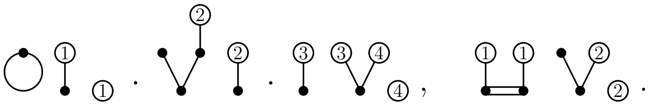
$$\pi \diamond \eta = \sum_{(\pi)} \pi_{(1)} \cdot (\pi_{(2)} \tilde{\curvearrowright} \eta), \quad \text{for } \pi, \eta \in \mathcal{EAF}, \tag{3.4.3}$$

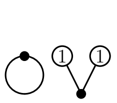
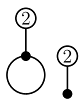
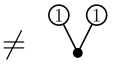
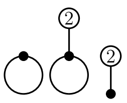
where $\Delta_{\mathcal{EA}}(\pi) = \sum_{(\pi)} \pi_{(1)} \otimes \pi_{(2)}$ is the deshuffle coproduct defined on $\mathcal{EAF}$.

We note that the space of exotic aromatic forests $\mathcal{EAF}$ can be defined as the symmetric algebra $S_{\mathcal{EA}}(\mathcal{PEAF})$ over the ring of exotic aromas. Analogously to Section 3.3.1, we define two possible extensions of the concept of clumped forests to the exotic context,

$$\mathcal{CEF} := \mathcal{S}(\mathcal{PEAF}), \quad \mathcal{CEF}_1 := \mathcal{S}(\mathcal{EAT}).$$

We note that both symmetric algebras are over the base field $\mathbb{R}$, meaning that the exotic aromas are attached to the rooted components. We recall that $\mathcal{EAT}$ is the space of exotic aromatic forests with one root and $\mathcal{CEF}_1 \subset \mathcal{CEF}$. Some elements of $\mathcal{CEF}$ are



We note that  $\cdot$  $\neq$  $\cdot$  in $\mathcal{CEF}$. Let $\Phi : \mathcal{CEF} \to \mathcal{EAF}$ denote the map that forgets the "clumping", that is,

$$\ker \Phi = \langle \omega\tau \cdot \gamma \cdot \pi - \tau \cdot \omega\gamma \cdot \pi \ : \ \omega \in \mathcal{EA}, \tau, \gamma \in \mathcal{EAT}, \pi \in \mathcal{CEF} \rangle.$$

Theorem 3.4.1. *The space of exotic aromatic forests forms a Grossman-Larson Hopf algebroid, and the space of clumped exotic forests forms a Grossman-Larson Hopf algebra. Moreover, $\Phi : \mathcal{CEF} \to \mathcal{EAF}$ is a surjective algebra morphism.*

Proof. We note that the algebra structure is obtained by considering a subalgebra of $(\mathcal{AF}_D, \diamond)$, while the coalgebra structure is obtained by taking a quotient of the coalgebra $(\mathcal{AF}_D, \Delta_{\mathcal{A}_D})$. This means that the algebra and coalgebra structures are not automatically compatible. We prove the compatibility condition

$$\Delta_{\mathcal{EA}}(\pi \diamond \eta) = \Delta_{\mathcal{EA}}(\pi) \diamond \Delta_{\mathcal{EA}}(\eta),$$

by induction on the number of connected components in π. Assume the compatibility is proven for all forests with the number of components less or equal to the number of components in π. Consider $\tilde{\pi} \cdot \pi$ with $\tilde{\pi}$ having one connected component, then,

$$\begin{aligned}\Delta_{\mathcal{EA}}((\tilde{\pi}\pi) \diamond \eta) &= \Delta_{\mathcal{EA}}(\tilde{\pi} \diamond (\pi \diamond \eta) - (\tilde{\pi} \,\tilde{\curvearrowright}\, \pi) \diamond \eta) \\ &= \Delta_{\mathcal{EA}}(\tilde{\pi}) \diamond \Delta_{\mathcal{EA}}(\pi) \diamond \Delta_{\mathcal{EA}}(\eta) - \Delta_{\mathcal{EA}}(\tilde{\pi} \,\tilde{\curvearrowright}\, \pi) \diamond \Delta_{\mathcal{EA}}(\eta) \\ &= \Delta_{\mathcal{EA}}(\tilde{\pi} \diamond \pi - \tilde{\pi} \,\tilde{\curvearrowright}\, \pi) \diamond \Delta_{\mathcal{EA}}(\eta) = \Delta_{\mathcal{EA}}(\tilde{\pi}\pi) \diamond \Delta_{\mathcal{EA}}(\eta),\end{aligned}$$

where we use the associativity of $\diamond$ and coassociativity of $\Delta_{\mathcal{EA}}$. It remains to show that

$$\Delta_{\mathcal{EA}}(\tilde{\pi} \diamond \eta) = \Delta_{\mathcal{EA}}(\tilde{\pi}) \diamond \Delta_{\mathcal{EA}}(\eta), \quad \text{for } \tilde{\pi} \in \mathcal{PEAF}, \eta \in \mathcal{EAF}.$$

We note that $\tilde{\pi}$ has multiple roots which, when grafted onto different connected components of η, connect them into a single component. We represent this by

$$\tilde{\pi} \diamond \eta = \sum_{S \in \eta} (\tilde{\pi} \hat{\curvearrowright} S) \cdot (\eta \setminus S),$$

where S is a possibly empty set of components of η and $\hat{\curvearrowright}$ is grafting product which attaches at least one root to each component of S, which means that the number of components in S is less or equal to the number of roots of $\tilde{\pi}$. If S is empty, then $\tilde{\pi} \hat{\curvearrowright} S = \tilde{\pi}$. Using this representation of $\tilde{\pi} \diamond \eta$, we have,

$$\begin{aligned}\Delta_{\mathcal{EA}}(\tilde{\pi} \diamond \eta) &= \Delta_{\mathcal{EA}}\big(\sum_{S \in \eta} (\tilde{\pi} \hat{\curvearrowright} S) \cdot (\eta \setminus S)\big) \\ &= \sum_{S \in \eta} (\tilde{\pi} \hat{\curvearrowright} S \otimes \mathbf{1} + \mathbf{1} \otimes \tilde{\pi} \hat{\curvearrowright} S) \cdot \Delta_{\mathcal{EA}}(\eta \setminus S) \\ &= \sum_{S \in \eta, (\eta \setminus S)} \tilde{\pi} \hat{\curvearrowright} S \cdot (\eta \setminus S)_{(1)} \otimes (\eta \setminus S)_{(2)} + (\eta \setminus S)_{(1)} \otimes \tilde{\pi} \hat{\curvearrowright} S \cdot (\eta \setminus S)_{(2)} \\ &= \sum_{(\eta)} (\tilde{\pi} \diamond \eta_{(1)} \otimes \eta_{(2)} + \eta_{(1)} \otimes \tilde{\pi} \diamond \eta_{(2)}) = \Delta_{\mathcal{EA}}(\tilde{\pi}) \diamond \Delta_{\mathcal{EA}}(\eta),\end{aligned}$$

and the compatibility condition is proven.

Following [48], the primitive elements $\mathcal{PEAF}$ endowed with $\tilde{\curvearrowright}$ form a pre-Lie algebra. We use the product $\tilde{\curvearrowright}$ to define the antipode S_{GL} for $\mathcal{EAF}$ by replacing all instances of $\curvearrowright$ in Proposition 3.2.12 by the product $\tilde{\curvearrowright}$, all instances of $\tau \in \mathcal{T}_D$ by $\tau \in \mathcal{PEAF}$, and $\omega \in \mathcal{A}_D$ by $\omega \in \mathcal{EA}$.

We obtain a bialgebra $(\mathcal{CEF}, \mathbf{1}, \diamond, \epsilon, \Delta)$ by noting that $\mathcal{CEF}$ can be obtained by taking a sub-D-algebra of D-algebra $\overline{\mathcal{CF}}_{\bullet_\mathbb{N}}$ which is a special case of $\mathcal{CF}_D$. The antipode S_{GL}^C for $\mathcal{CEF}$ is obtained the same way, but the trees $\tau \in \mathcal{T}_D$ are replaced by $\tau \in \mathcal{PEAF}$ and the identities (i) are ignored.

This proves that we have a Hopf algebroid and Hopf algebra structures over $\mathcal{EAF}$ and $\mathcal{CEF}$, respectively. We note that $\Phi(\pi \diamond \mu) = \Phi(\pi) \diamond \Phi(\mu)$ and $\Phi(\mathbf{1}) = \mathbf{1}$, so Φ is a surjective algebra morphism. □

We note that $(\mathcal{EAF}, \cdot, \Delta_{\mathcal{EA}}, \curvearrowright)$ is not a pre-Hopf algebra, as

$$\pi \curvearrowright (\pi_1 \cdot \pi_2) \neq (\pi_{(1)} \curvearrowright \pi_1) \cdot (\pi_{(2)} \curvearrowright \pi_2) \text{ in } \mathcal{EAF}.$$

Moreover, $(\mathcal{EAF}, \cdot, \Delta_{\mathcal{EA}}, \tilde{\curvearrowright})$ is not a pre-Hopf algebra for the same reason. However, $(\mathcal{EAF}, \cdot, \curvearrowright)$ is a D-subalgebra of $(\mathcal{AF}_D, \cdot, \curvearrowright)$ with the grading of the associative algebra $(\mathcal{EAF}, \cdot)$ given by the number of roots.

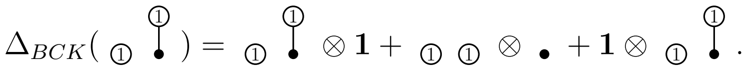
We also note that following the discussion in Section 3.2.4, $\mathcal{EAF}$ is endowed with a Butcher-Connes-Kreimer coproduct which is adjoint to the Grossman-Larson product with respect to the inner product $\langle -, - \rangle_\sigma$ with the Δ_{BCK} coproduct defined in Definition 3.4.2.

Definition 3.4.2. The Butcher-Connes-Kreimer coproduct on $\mathcal{EAF}$ is defined as

$$\Delta_{BCK}(\pi) := \sum_{\pi_0 \subset \pi} \pi \setminus \pi_0 \otimes \pi_0,$$

where the sum runs over all rooted exotic aromatic subforests $\pi_0 \in \mathcal{EAF}$ of π such that $\pi \setminus \pi_0 \in \mathcal{EAF}$ and there are no edges going from π_0 to $\pi \setminus \pi_0$ in π.

An important property of Δ_{BCK} over exotic aromatic forests is that it keeps the vertices of a liana on the same side of the tensor product, for example,



We use the coproduct Δ_{BCK} to characterize the exotic aromatic S-series of a composition of two integrators. Let Ψ_h^1 and Ψ_h^2 be two integrators that share the same timestep while having independent noise terms. It establishes that the composition of these two integrators, each with an exotic aromatic S-series, results in an integrator that also possesses an exotic aromatic S-series of a specific form.

Theorem 3.4.3 (Composition of integrators)**.** *Consider two independent integrators Ψ_h^1 and Ψ_h^2 with exotic aromatic S-series $S(a_1)$ and $S(a_2)$, then the composition of Ψ_h^1 and Ψ_h^2 has the following S-series*

$$\mathbb{E}[\phi((\Psi_h^2 \circ \Psi_h^1)(x))] = S(a_1 * a_2)[\phi](x).$$

with $a_1 * a_2 = m_{\mathbb{R}} \circ (a_1 \otimes a_2) \circ \Delta_{BCK}$ where Δ_{BCK} is the Butcher-Connes-Kreimer coproduct over exotic aromatic forests.

Proof. The integrators Ψ_h^1 and Ψ_h^2 have exotic aromatic S-series, that is,

$$\mathbb{E}[\phi(\Psi_h^1(X_0))] = S(a_1)[\phi]\,, \quad \text{and} \quad \mathbb{E}[\phi(\Psi_h^2(X_0))] = S(a_2)[\phi]\,.$$

We use the independence of their noise terms to have the following equality,

$$\mathbb{E}[\phi((\Psi_h^2 \circ \Psi_h^1)(X_0))] = S(a_1)\big[\mathbb{E}[\phi(\Psi_h^2(X_0))]\big] = S(a_1)\big[S(a_2)[\phi]\big]\,.$$

Composition of differential operators corresponds to the Grossman-Larson product of the corresponding exotic aromatic forests and the Grossman-Larson product is adjoint to the Butcher-Connes-Kreimer coproduct using 3.2.17. Therefore, we have,

$$\mathbb{E}[\phi((\Psi_h^2 \circ \Psi_h^1)(X_0))] = S(a_1 * a_2)[\phi],$$

and the composition law is proved. □

Let us now introduce the substitution law for exotic aromatic S-series. Let $S_f(a)$ denote the exotic S-series of a numerical integrator solving an SDE with drift f. The drift can be chosen to be an S-series $S_f(b_0)$, where $b_0 \in EAT^*$ is a functional that is non-zero only on exotic aromatic trees. The S-sereis $S_f(b_0)$ is also denoted by $B_f(b)$ for simplicity (see Definition 3.0.1). Theorem 3.4.5 asserts that applying a numerical integrator to an SDE with drift $S_f(b_0)$ is equivalent to applying a numerical integrator with an exotic S-series of a specific form to the SDE with drift f.

Definition 3.4.4. Let the CEM coaction $\Delta_{CEM} : \mathcal{EAF} \to \mathcal{CEF}_1 \otimes \mathcal{EAF}$ over exotic aromatic forests be defined as

$$\Delta_{CEM}(\pi) := \sum_{p \subset \pi} p \otimes \pi/_p,$$

where the sum is over all clumped exotic subforests $p \in \mathcal{CEF}_1$ that cover all black vertices and $\pi/_p$ is the exotic aromatic forest obtained by contracting the exotic aromatic trees of p into black vertices. If the forest $\pi \in \mathcal{EAF}$ doesn't have valid subforests $p \in \mathcal{CEF}_1$, then $\Delta_{CEM}(\pi) = \mathbf{1} \otimes \pi$. For details see the proof of Theorem 3.4.6.

Theorem 3.4.5 (Substitution law)**.** *Let $a \in \mathcal{EAF}^*$, $\ b \in \mathcal{EAT}^*$, then,*

$$S_{B_f(b)}(a) = S_f(b_c \star a), \quad \textit{with } b_c \star a = m_{\mathbb{R}} \circ (b_c \otimes a) \circ \Delta_{CEM},$$

where b_c is the character of $\mathcal{CEF}_1$ that extends b and Δ_{CEM} is defined in Definition 3.4.4.

The techniques of backward error analysis and the modified equation rely heavily on the substitution law and are introduced in the beginning of Section 3 (see Theorems 3.0.5 and 3.0.7). Let us prove that $\mathcal{CEF}$ is endowed with a Hopf algebra structure with coproduct Δ_{CEM} which gives us a simplified way of computing the $\mathcal{CEF}_1$-coaction Δ_{CEM} over $\mathcal{EAF}$. We use the map Φ^* studied in Section 3.3.1.

Theorem 3.4.6. *The space of clumped exotic forests $\mathcal{CEF}$ forms a Hopf algebra*

$$(\mathcal{CEF}, \mathbf{1}, \cdot, \mathbf{1}^*, \Delta_{CEM}, S_{CEM}),$$

where Δ_{CEM} is the coproduct extended from Prim($\mathcal{EAF}$) to $\mathcal{CEF}$ by respecting the concatenation product. Moreover, $\Phi^* : \mathcal{EAF} \to \mathcal{CEF}$ is a $\mathcal{CEF}_1$-comodule morphism where Φ^* is the adjoint of Φ, that is, $\Phi \circ \delta_\sigma = \delta_\sigma \circ \Phi^*$.

Proof. We recall that, following Theorem 3.3.7, $B_{CEM} := (\mathcal{CF}_D^{\otimes D}, \cdot, \Delta_{CEM})$ is a bialgebra which becomes a Hopf algebra $H_{CEM} := (\mathcal{CF}_D, \cdot, \Delta_{CEM})$ once we take the quotient of B_{CEM} by the ideal $\langle (\mathbf{1} - \bullet)\iota_\bullet \rangle + \mathcal{J}$ defined as

$$\mathcal{J} := \langle (\mathbf{1} - \textcircled{k})\iota_{\textcircled{k}}, \pi\iota_{\textcircled{k}} \ : \ \pi \notin \{\mathbf{1}, \textcircled{k}\}, k \in \mathbb{N} \rangle.$$

The obtained coproduct of H_{CEM} can now be described as

$$\Delta_{CEM}(\pi) = \sum_{p \subset \pi} p \otimes \pi/_p, \tag{3.4.4}$$

where the sum is over all clumped subforests $p \in \mathcal{CF}_D$ that cover all black vertices and $\pi/_p$ is the clumped forest obtained by contracting the aromatic trees of p

into black vertices. If the forest $\pi \in \mathcal{CF}_D$ doesn't have valid subforests $p \in \mathcal{CF}_D$, then $\Delta_{CEM}(\pi) = \mathbf{1} \otimes \pi$. We build the algebra $(\mathcal{CEF}, \cdot)$ by taking a subalgebra of $(\mathcal{CF}_D, \cdot)$. The coalgebra structure of $(\mathcal{CEF}, \Delta_{CEM})$ is obtained by taking a quotient over an appropriate coideal where we require the connected components of p to have a single root. The compatibility between the product and coproduct follows directly from the description of Δ_{CEM} in (3.4.4). We obtain the Hopf algebra

$$(\mathcal{CEF}, \mathbf{1}, \cdot, \mathbf{1}^*, \Delta_{CEM}, S_{CEM}),$$

with $\Delta_{CEM} : \mathcal{CEF} \to \mathcal{CEF} \otimes \mathcal{CEF}$ which corestricts to $\mathcal{CEF}_1 \otimes \mathcal{CEF}$ and Φ^* is a $\mathcal{CEF}_1$-comodule morphism. □

We can simplify the computation of the substitution law for exotic aromatic S-series using the map Φ and Theorem 3.4.6. Given a functional $a \in \mathcal{EAF}^*$, we use the discussion from Section 3.3.1 to define a functional $a_C \in \mathcal{CEF}$ over clumped exotic forests as

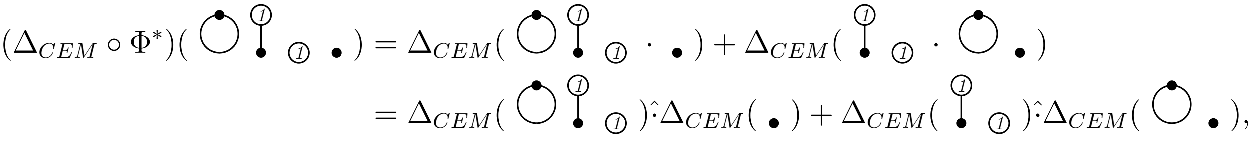
$$a_C(\pi) := \frac{1}{n^m} a(\Phi(\pi)),$$

where n is the number of rooted components and m is the number of aromas. We have the property $a = a_C \circ \Phi^*$ which we use to compute the substitution law as follows

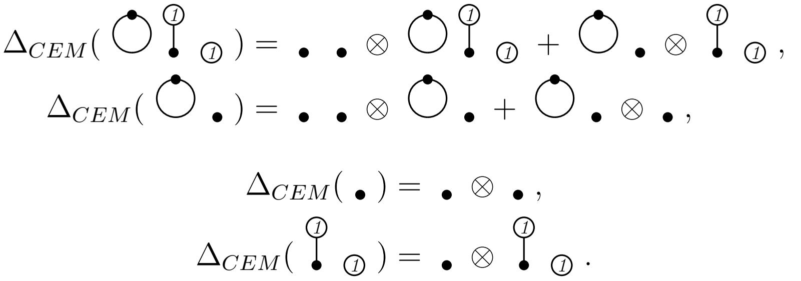
$$b_c \star a = b_c \star (a_C \circ \Phi^*) = (b_c \star a_C) \circ \Phi^*.$$

The substitution law $b_c \star a_C$ over $\mathcal{CEF}$ is easier to compute since Δ_{CEM} over $\mathcal{CEF}$ respects concatenation. Therefore, computing the values of Δ_{CEM} on $\mathcal{PEAF}$ is enough to obtain its values over all clumped exotic forests.

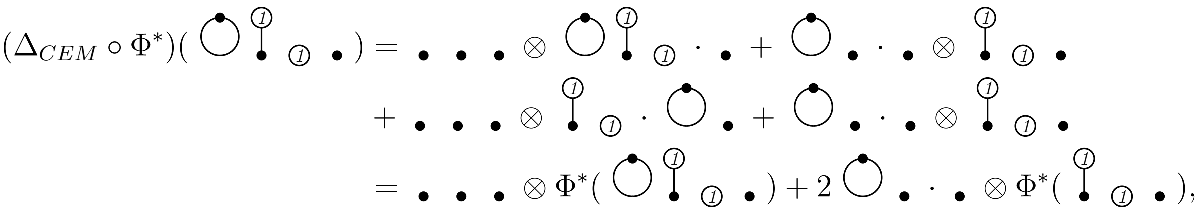
Example 3.4.7. *Let us compute the CEM coproduct over $\mathcal{EAF}$ using the comodule morphism Φ^*.*



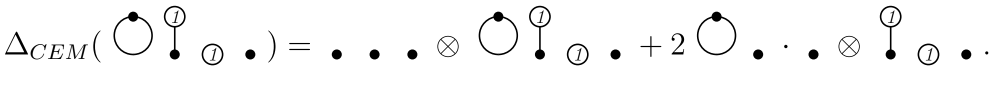
where $(\pi_{(1)} \otimes \pi_{(2)})\hat{\cdot}(\mu_{(1)} \otimes \mu_{(2)}) = (\pi_{(1)} \cdot \mu_{(1)}) \otimes (\pi_{(2)} \cdot \mu_{(2)})$ and



Therefore, we have



which agrees with a direct computation that gives



Chapter 4

Arboretum: symbolic package for automatic computation

The `Arboretum.hs` package offers a comprehensive suite of tools to streamline computations involving algebras of graphs. It was developed in collaboration with Jean-Luc Falcone from the Computer Science Department of University of Geneva and Gilles Vilmart. A publication detailing the contributions of this package [9] is currently in preparation. The design of the package is guided by three core principles:

1. Readability: The implementation of structures and operations closely follows their mathematical definitions, ensuring that the code is intuitive and transparent.
2. Extensibility: The package is built to be easily extendable, allowing new features and operations to be added without significant modifications to the existing codebase.
3. Testing: Special emphasis is placed on testing. The package includes mechanisms to easily validate the correctness of implementations, reflecting a commitment to reliability.

In developing this package, we have prioritized readability and user experience over raw performance. This trade-off is intentional, as the package is designed to serve both as a computational tool and a platform for exploring and experimenting with algebras of graphs. The simplicity of its implementation, combined with a focus on clarity, ensures that users—particularly those interested in algebraic structures—can easily work with `Arboretum.hs` while maintaining the flexibility to extend its functionality as needed.

One notable example is `BSeries.jl` [65], a Julia package that replaces `BSeries.py` [66], an earlier Python implementation by the same authors. It focuses on the efficient implementation of classical techniques for B-series and Runge-Kutta methods. Developed with different priorities in mind, it has its own specialization. A key distinction is that `BSeries.jl` is primarily designed for numerical analysts, emphasizing well-established structures from numerical analysis. Consequently, it is less flexible from an algebraic and combinatorial perspective, making it less suited for exploring extensions within the tree and forest framework. See [38] for details.

We also mention the master's thesis [73], which introduces a Python package [74] for automating various classical computations in the theory of B-series. Additionally, [59] presents recursive formulas for operations on planar forests, with the goal of developing a Haskell package for their computation.

4.1 Getting started

Users are encouraged to explore the examples provided in the manual for a practical understanding of the package's functionality. To begin using `Arboretum.hs`, install the Haskell Tool Stack (Stack) by following the instructions from

`https://haskellstack.org`,

clone the git repository

`https://gitlab.unige.ch/Eugen.Bronasco/arboretum.hs`

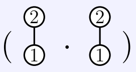
and run `stack repl` in the root folder. The package uses a custom extension of the LaTeX package `planarforest` to draw the trees and forests. The original `planarforest` package is available at

`https://hmarthinsen.github.io/planarforest/`

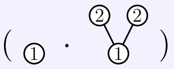
The `display` function is used to display vectors of trees and forests. It generates a PDF file in the root folder called `output.pdf` which contains the visual representation of the vector and attemps to open the file using Zathura document viewer available on Linux. The `display` function is used in the examples below to illustrate the results of the computations.

Usage Example 4.1.1. Below you can find some examples of the package usage where we define two planar forests `f1` and `f2`, graft `f1` onto `f2`, and substitute the vertices decorated by `1` in `f2` by trees of `f1`. The details can be found in Section 4.5.

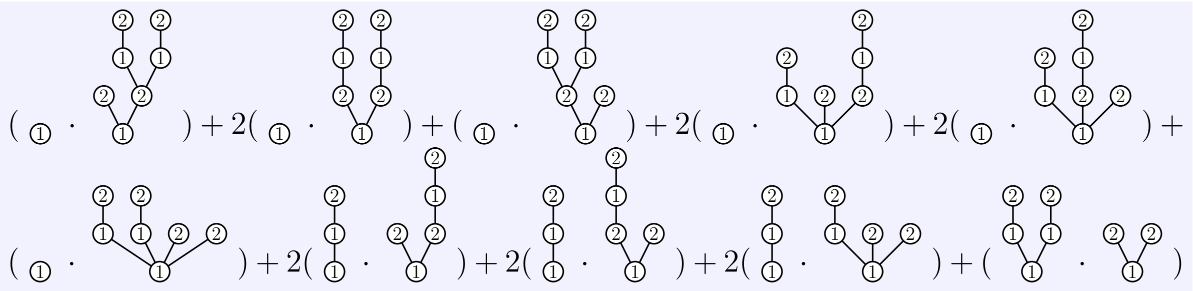
```
f1 = [
  PT 1 [PT 2 []],
  PT 1 [PT 2 []]
]
display $ vector f1
```



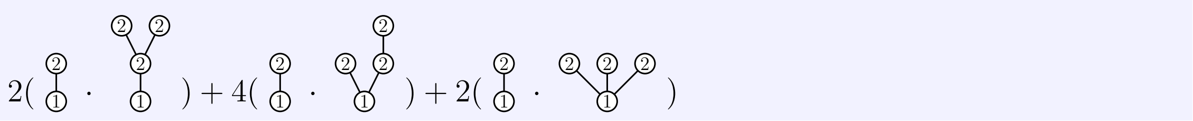
```
f2 = [
  PT 1 [],
  PT 1 [PT 2 [], PT 2 []]
]
display $ vector f2
```



```
display $ f1 `graft` f2
```



```
display $ substitute [PT 1 []] (map (:[]) f1) f2
```



4.2 Why Haskell?

Haskell is a strong choice for the Arboretum.hs package, as it aligns well with our goals of readability, extensibility, and testing. We are not the first to make this choice; an alternative discussion on the benefits of using Haskell can be found in [59].

Below, we explore its key advantages while also addressing potential drawbacks, particularly in comparison to languages such as Julia, Matlab, and Python.

4.2.1 Advantages of Haskell

1. The syntax is well-suited for writing code that mirrors mathematical definitions, particularly in domains like algebraic structures. This close alignment between code and mathematical notation enhances clarity, making the code easier to understand and reason about. Recursive structures, such as trees, are especially intuitive to implement in Haskell due to its functional paradigm.

2. Haskell enforces immutability, meaning data cannot be altered once created. This property simplifies reasoning about program behavior and ensures that functions remain free from side effects.

3. A robust, static type system enables the definition of custom types that reflect the algebraic structures in the package, allowing errors to be caught at compile time. This provides an additional layer of safety and encourages reusable, extendable code. In comparison, while languages like Python and Matlab offer flexibility, their dynamic typing can lead to runtime errors that are harder to catch.

4. Haskell's lazy evaluation model means that computations are deferred until their results are required. This is particularly advantageous for dealing with infinite structures, such as the B-series mentioned in the package's planned features. This allows for handling infinite sums efficiently without running into performance bottlenecks, a flexibility not as easily managed in languages like Julia or Matlab.

5. Haskell's emphasis on composability and modularity makes the codebase easy to extend. New structures and operations can be introduced without disrupting the existing framework, fostering scalability.

6. Haskell's support for property-based testing, through libraries like QuickCheck, allows us to define properties that algebraic structures should satisfy and automatically generate test cases to ensure correctness. This level of verification is harder to achieve in dynamically-typed languages like Python, where more reliance is placed on unit tests and manual validation.

4.2.2 Disadvantages and Remedies

1. While Haskell prioritizes readability and correctness, it may not always match the performance of Julia, Matlab, or Python, particularly in numerical computations. Performance bottlenecks can be mitigated by using Foreign Function Interface (FFI) which enables integration with high-performance code from other languages, allowing Haskell to leverage the strengths of Julia, Python, or C for specific tasks.

2. Haskell's functional programming model and emphasis on immutability can be challenging for developers accustomed to more imperative languages like Python, Matlab, or Julia. These languages are often easier to learn offering familiar syntax and workflows. However, the Arboretum.hs package is written without the use of advanced Haskell techniques which reduces the complexity and makes it more approachable for new users.

In summary, while Haskell may not offer the same performance or immediate accessibility as Julia, Matlab, or Python in some areas, it excels in readability, correctness, and modularity. With the right optimizations and careful design, Haskell is a robust and extendable tool for working with algebras of graphs.

4.3 Algebras of graphs

We present several examples of implementing algebras of graphs, showcasing and explaining the majority of the package's functionality. When constructing an algebra of graphs, two primary approaches can be adopted:

1. Defining a general graph and subsequently restricting the type of graphs we accept by imposing additional structure (as demonstrated in Section 4.4),

2. Building the specific graphs of interest directly from the ground up (as demonstrated in Section 4.5).

Each approach comes with its own set of advantages and drawbacks. The first approach is more flexible and can accommodate a broader variety of graphs, but it may introduce additional complexity and reduce efficiency. The second, more specialized approach, is often simpler and more efficient to implement, although it can require more upfront work to define the specific graphs.

In our examples, we focus on the grafting product and the associated algebra of forests and aromatic forests. Later, we present the tools and techniques used to implement substitution on classical and aromatic forests.

4.4 Grafting of graphs

In this section, we discuss the definition and implementation of graphs, along with the grafting operation. We provide an example of implementing an algebra of graphs, starting with a general graph and imposing only the minimal required structure to define the product of interest, which is the grafting product in our case.

Definition 4.4.1. A **graph** g is a set of vertices $V(g)$ together with a set of directed edges $E(g)$. An edge e is defined by its source $s(e) \in V(g)$ and target $t(e) \in V(g)$. A **rooted** graph has a marked vertex called a root. The set of rooted graphs is denoted by G_R.

We implement the Definition 4.4.1 below:

Implementation 4.4.2. A graph is any type `g` that is an instance of the `Graph` type class. The `Graph` type class requires the type of the edges `Edge g` to be an instance of the `GraphEdge` type class, which defines the source, target, and construction of edges. The `Graph` type class also requires the type of the vertices `Vertex g` to be the `EndPoint` of the edges. The `Graph` type class provides functions for constructing graphs, adding edges, and adding graphs together.

```
class GraphEdge e where
    -- | Ad edge of type `e` has endpoints of type `EndPoint e`.
    type EndPoint e

    source :: e -> EndPoint e
    target :: e -> EndPoint e

    -- | Constructs an edge from two endpoints an possibly
    --  additional data of type `a`.
    edge :: a -> EndPoint e -> EndPoint e -> e

class
    ( GraphEdge (Edge g)
    , Vertex g ~ EndPoint (Edge g)
    ) => Graph g where
    -- | The type of edges in the graph which must be
    -- an instance of `GraphEdge`.
    type Edge g

    -- | The type of vertices in the graph which must be
    -- the `EndPoint` of the edges.
    type Vertex g
    type Vertex g = EndPoint (Edge g)

    singleton :: Vertex g -> g
    edges :: g -> MultiSet (Edge g)
    vertices :: g -> MultiSet (Vertex g)
```

```haskell
    -- | Adds the vertices and edges of a graph to another graph.
    -- We note that the type of edges must coincide.
    addGraph :: (Graph g0, Edge g ~ Edge g0) => g0 -> g -> g

    addEdge :: Edge g -> g -> g

class (Graph g) => RootedGraph g where
    root :: g -> Vertex g
```

A straightforward implementation of a graph which uses integers as labels is included in the package and is presented in Implementation 4.4.3.

Implementation 4.4.3. A simple implementation of a graph where we create an instance of the `GraphEdge` type class for pairs of any type `a`. We also define the `IntegerGraph` type, which is an instance of the `Graph` type class and is fully defined by a multiset of vertices and edges. Finally, we define the `Rooted` type, which is an instance of the `RootedGraph` type class.

```haskell
instance GraphEdge (a, a) where
    type EndPoint (a, a) = a
    edge _ = (,)
    source (x, _) = x
    target (_, y) = y

data IntegerGraph
    = IG (MS.MultiSet Integer) (MS.MultiSet (Integer, Integer))

instance Graph IntegerGraph where
    type Edge IntegerGraph = (Integer, Integer)

    singleton v = IG (singleton v) empty
    edges (IG _ es) = es
    vertices (IG vs _) = vs
    addGraph g (IG vs es) =
        IG (vertices g `union` vs) (edges g `union` es)
    addEdge e (IG vs es) =
        IG vs (e `insert` es)

integerGraph :: [Integer] -> [(Integer, Integer)] -> IntegerGraph
integerGraph vs es = IG (MS.fromList vs) (MS.fromList es)

data Rooted g = R (Vertex g) g

instance (Graph g) => RootedGraph (Rooted g) where
    root (R r _) = r


rooted :: (Graph g, Eq (Vertex g)) => g -> Vertex g -> Rooted g
rooted g r =
    if r `vertexOf` g
        then R r g
        else error "Root vertex not in graph"
```

The function `integerGraph` is a helper function that simplifies the construction of graphs and the function `rooted` is used to set the root of a graph.

We define a graph with the set of vertices $V = \{1, 2, 3\}$ and the set of edges $E = \{(1, 1), (2, 1), (2, 3)\}$ in Example 4.4.4 and set the vertex 1 to be the root.

Usage Example 4.4.4. An example of a graph:

```
>>> g = integerGraph [1,2,3] [(1,1),(2,1),(2,3)]
>>> g
IntegerGraph(V=[1,2,3], E=[(1,1),(2,1),(2,3)])
>>> rg = rooted g 1
>>> rg
RootedIntegerGraph(V=[1,2,3], E=[(1,1),(2,1),(2,3)], R=1)
```

We note that since we don't assume any structure on the graph, we can't display it in a more visually appealing way.

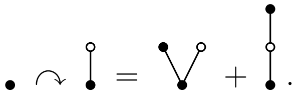
We are now ready to define and implement a generalization of the grafting product. Let $\mathcal{G}$ and $\mathcal{G}_R$ denote the vector spaces spanned by G and G_R, respectively.

Definition 4.4.5. The **grafting product** $\curvearrowright : \mathcal{G}_R \otimes \mathcal{G} \to \mathcal{G}$ is a bilinear map defined by taking the sum over all the ways to connect the root of the left operand to a vertex of the right operand. That is,

$$g_r \curvearrowright g = \sum_{v \in V(g)} g_r \curvearrowright_v g,$$

where the operation $\curvearrowright_v$ connects the root of g_r to the vertex v of g.

For example,



The implementation of the grafting product that closely resembles the Definition 4.4.5 is presented below.

Implementation 4.4.6. Implementation of the grafting product:

```
graftGraph
    :: ( Eq a2, Graded a2, Graph a2
       , RootedGraph a1, Edge a1 ~ Edge a2
       )
    => a1 -> a2 -> Vector Integer a2
graftGraph rg1 g2 =
    vectorFromNonDecList $
        map ((1 *^) . graftGraphTo rg1 g2) $
            toList $
                vertices g2

graftGraphTo
    :: ( RootedGraph a1
       , Graph a2, Edge a1 ~ Edge a2
       )
    => a1 -> a2 -> Vertex a2 -> a2
graftGraphTo rg1 g2 v = addGraph rg1 $ addEdge new_edge g2
  where
    new_edge = edge () (root rg1) v
```

The map `graftGraph` requires the left operand to be an instance of the `RootedGraph` type class and the right operand to be an instance of the `Graph` type class with coinciding edges. The function `graftGraphTo` connects the root of the left operand to a specified vertex of the right operand.

We use the `bilinear` function which extends a map from the basis set to the corresponding vector space to apply the grafting product to vectors of graphs. The maps `vector`, `vectorFromList`, and `vectorFromNonDecList` construct vectors and are discussed in great detail in the manual of the package.

Usage Example 4.4.7. We define `g1` to be a graph with vertices $\{1\}$ and no edges, `g2` to be a graph with vertices $\{2\}$ and no edges, `g3` to be a graph with vertices $\{3, 4\}$ and an edge $(4, 3)$, and `g4` to be a graph with vertices $\{5\}$ and an edge $(5, 5)$. We then compute the product of the vectors $g1 + 2 \cdot g2$ and $3 \cdot g3 + 4 \cdot g4$ using the grafting product.

```
>>> rg1 = rooted (integerGraph [1] []) 1
>>> rg2 = rooted (integerGraph [2] []) 2
>>> g1 = integerGraph [3,4] [(4,3)]
>>> g2 = integerGraph [5] [(5,5)]
>>> bilinear graftGraph
      (vector $ 1*^rg1 +: 2*^rg2 +: Zero)
      (vectorFromList [3*^g1,4*^g2])
(4 *^ IntegerGraph(V=[1,5], E=[(1,5),(5,5)])
      + 8 *^ IntegerGraph(V=[2,5], E=[(2,5),(5,5)]))_2
      + (3 *^ IntegerGraph(V=[1,3,4], E=[(1,3),(4,3)])
      + 3 *^ IntegerGraph(V=[1,3,4], E=[(1,4),(4,3)])
      + 6 *^ IntegerGraph(V=[2,3,4], E=[(2,3),(4,3)])
      + 6 *^ IntegerGraph(V=[2,3,4], E=[(2,4),(4,3)]))_3
```

We note that `rg1` and `rg2` are rooted graphs while `g1` and `g2` are non-rooted. See the manual of the package for more details on the use of `bilinear`, `vector`, and `vectorFromList`.

We recall that the technique presented in this subsection defines a general graph before imposing the structure necessary to define the grafting product. While this approach is flexible and can accommodate a wide variety of graphs, it may also introduce additional complexity and reduce computational efficiency.

4.5 Grafting and substituting of decorated forests

In this section, we take an alternative approach by constructing the graphs we are interested in from the ground up. This approach allows us to leverage their properties to implement efficient formulas for operations and ensures that the structure we aim to preserve is maintained through these operations—a guarantee provided by Haskell's strong and flexible type system. However, this approach requires a more detailed understanding of the graph's structure compared to the one discussed in Section 4.4.

We will explore the definition and implementation of decorated forests, along with grafting, Grossman-Larson, and substitution operations. Later, we extend the formalism to include aromatic forests, incorporating grafting and substitution operations as well.

4.5.1 Decorated forests

Decorated forests are graphs in which every connected component is a decorated tree. A tree is a connected graph with a unique vertex called *the root of the tree* and a unique path from every vertex to the root. A tree τ is decorated if, given a set D, there exists a map $\alpha : V(\tau) \to D$ which assigns a possibly non-unique decorator to each vertex of τ. All trees and forests considered in this section are decorated, so we will often omit mentioning this for simplicity. Below you can find an equivalent definition of decorated trees which is more suitable for implementation.

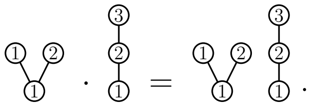
Definition 4.5.1. Given a set of decorations D, a **tree** decorated by D is a tuple (r, π) of a root $r \in D$ and a collection π of trees decorated by D called **children** of r.

Let the set of trees be denoted by T. We note that the collections of trees π are also called **forests** and their set is denoted by F.

Implementation 4.5.2. The `IsDecorated` type class is used to define the decoration of the tree, while the `IsTree` type class is used to define the root, children, and construction of the tree.

```haskell
class IsDecorated a where
    type Decoration a

class (IsDecorated t) => IsTree t where
    root :: t -> Decoration t
    children :: t -> [t]
    buildTree :: Decoration t -> [t] -> t
```

The vector space of forests $\mathcal{F}$ forms an algebra with concatenation product. For example,



An implementation of planar trees is presented in Implementation 4.5.3 with forests being represented as lists of planar trees.

Implementation 4.5.3. We define the `PlanarTree` data type which is an instance of the `IsDecorated` and `IsTree` type classes. The `PlanarTree` data type represents a tree with a root and a list of children.

```
data PlanarTree d = PT
    { planarRoot :: d
    , planarChildren :: [PlanarTree d]
    }

instance IsDecorated (PlanarTree d) where
    type Decoration (PlanarTree d) = d

instance IsTree (PlanarTree d) where
    root = planarRoot
    children = planarChildren

    buildTree = PT
```

An example of the use of planar forests can be found in Usage Example 4.1.1.

4.5.2 Planar and non-planar forests

A forest is *planar* if the order of its trees and the branches within the trees matters. In a *non-planar* forest, the order of the trees and their branches does not matter. Both classes of forests play important roles, but from an implementation perspective, planar forests are simpler to handle since they rely on list structures, whereas non-planar forests rely on multiset structures. Working with lists leads to cleaner code. Moreover, mathematically, non-planar trees can be viewed as a subspace or a quotient space of planar trees. To enable seamless transitions between planar and non-planar spaces, the `Planarable` type class is implemented.

Implementation 4.5.4. We implement the `Planarable` type class:

```
class Planarable t where
    type Planar t

    planar :: t -> Planar t
    nonplanar :: Planar t -> t
```

For example, we can use the `planar` map to obtain non-planar forests as is demonstrated in Usage Example 4.5.5.

Usage Example 4.5.5. There exist instances of the `Planarable` type class that have planar trees and planar forest as values of the `Planar` type. Therefore, the map `nonplanar` is defined on them. We note that `t1` and `t2` differ in the order of their children which makes them non-equal. We observe that for the non-planar versions of these trees, the order of the children does not matter.

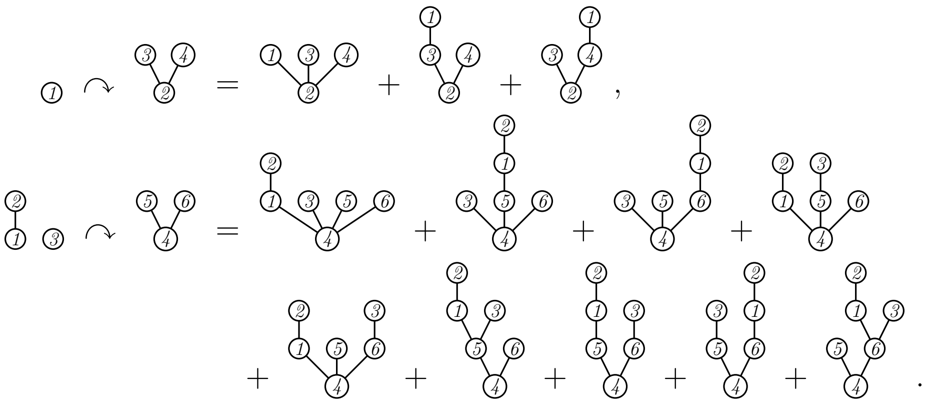
```
>>> t1 = PT 1 [PT 2 [], PT 3 []]
>>> t2 = PT 1 [PT 3 [], PT 2 []]
>>> t1 == t2
False
>>> t1' = nonplanar t1 :: Tree Integer
>>> t2' = nonplanar t2 :: Tree Integer
>>> t1' == t2'
True
```

The type annotation `::Tree Integer` specifies that the resulting trees are of `Tree Integer` type. This distinction is necessary because multiple instances of the `Planarable` class may share the same `Planar` type, which is `PlanarTree`. While the definition of the `Tree d` data type is omitted, it is structurally similar to its planar counterpart, with the key difference being that the list of children is replaced by a multiset of children.

4.5.3 Grafting

We use Sweedler notation to denote the deshuffle coproduct as $\Delta(\pi) = \sum_{(\pi)} \pi^{(1)} \otimes \pi^{(2)}$ and consider the operation of grafting on decorated forests. The forest π_1 is grafted onto the forest π_2 by taking the sum over all ways to attach the roots of the forest π_1 to the vertices of π_2.

Example 4.5.6. *Some examples of grafting are:*



Definition 4.5.7. **Grafting** of two empty forests gives an empty forest:

$$\mathbf{1} \curvearrowright \mathbf{1} := \mathbf{1},$$

forest on an empty forest is 0:

$$\pi \curvearrowright \mathbf{1} := 0, \quad \text{for } \pi \neq \mathbf{1},$$

forest on a tree is defined using Grossman-Larson product:

$$\pi \curvearrowright \tau := (r, \pi_1 \diamond \pi_2), \quad \text{for } \tau = (r, \pi_2) \in T,$$

forest on a non-empty forest is defined using deshuffle coproduct:

$$\pi_1 \curvearrowright (\tau \cdot \pi_2) := \sum_{(\pi_1)} (\pi_1^{(1)} \curvearrowright \tau) \cdot (\pi_1^{(2)} \curvearrowright \pi_2), \quad \text{for } \pi_1, \pi_2 \in F.$$

with $(r, -)$ being linear and $\diamond$ denoting the **Grossman-Larson** product defined as

$$\pi_1 \diamond \pi_2 := \sum_{(\pi_1)} \pi_1^{(1)} \cdot (\pi_1^{(2)} \curvearrowright \pi_2).$$

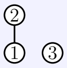
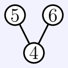
We note that Definition 4.5.7 implies $\mathbf{1} \curvearrowright \pi = \pi$. Implementation 4.5.8 provides the implementation of the grafting operation for planar forests.

Implementation 4.5.8. To facilitate future extension of the algebra to include aromatic forests, we define the `Graftable` type class for graphs that support grafting operations:

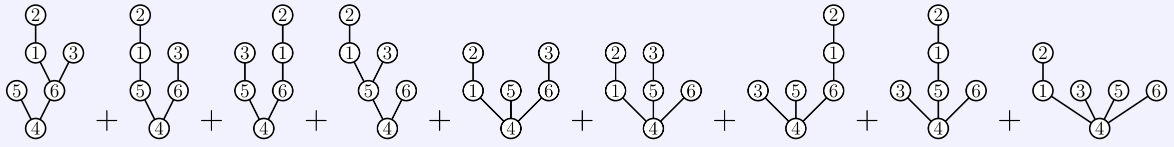
```
class (IsVector a) => Graftable a where
    graft :: a -> a -> Vector (VectorScalar a) (VectorBasis a)

instance
    ( IsTree t, IsVector t
    -- [...]
    )
    => Graftable [t]
    where
    graft [] [] = vector []
    graft _ [] = vector Zero
    graft [] f2 = vector f2
    graft f [t] =
        linear ((: []) . buildTree (root t)) $ gl f $ children t
    graft f1 (t : f2) =
        linear perCoproductTerm $ deshuffleCoproduct f1
      where
        perCoproductTerm (f11, f12) = graft f11 [t] * graft f12 f2

gl  :: ( IsTree t, IsVector t
       -- [...]
       )
    => [t] -> [t] -> Vector (VectorScalar t) [t]
gl f1 f2 = linear perCoproductTerm $ deshuffleCoproduct f1
  where
    perCoproductTerm (f11, f12) = vector f11 * graft f12 f2
```

We refer to the manual of the Arboretum package for the details related to the implementation of the vector spaces and the operations on them.

We demonstrate the use of the `graft` operation by computing an example found in Example 4.5.6.

Usage Example 4.5.9. We define the trees  and  and compute their grafting product.

```
>>> f1 = [PT 1 [PT 2 []], PT 3 []]
>>> f2 = [PT 4 [PT 5 [], PT 6 []]]
>>> display $ f1 `graft` f2
```



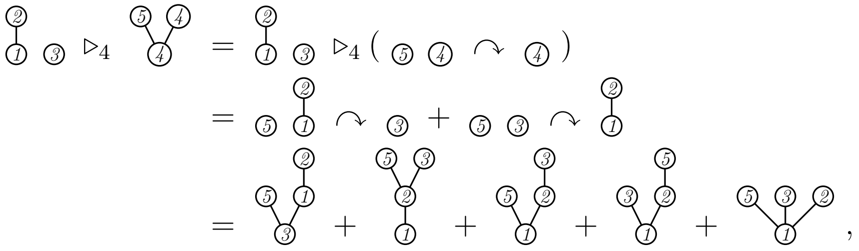
4.5.4 Substitution

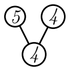
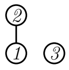
Decorated vertices together with grafting and concatenation generate the whole space of decorated forests. That is, for all $\tau \in T$, we have,

$$\tau = \pi \curvearrowright r, \quad \pi = \tau_1 \cdots \tau_n, \quad \text{where } \tau_i \in T. \tag{4.5.1}$$

We consider a substitution operation $\triangleright_d$ that substitutes the vertices decorated by d in (4.5.1) of the right operand by trees of the left operand in all possible ways.

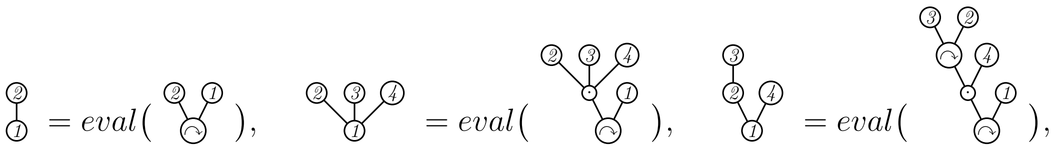
Example 4.5.10. *An example of substitution is:*



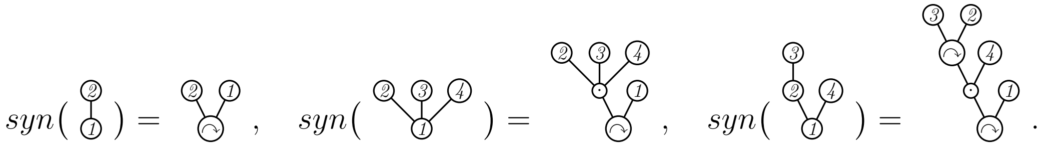
in which we substitute the vertices decorated by 4 *in*  *by the trees of*  *in all possible ways.*

A forest can be represented as a syntactic tree with leaves being vertices and internal nodes being the operations of grafting or concatenation. Let the set of syntactic trees be denoted by ST, then, there is a map $eval : ST \to F$ which evaluates the syntacitc tree and $syn : F \to ST$ that deconstructs a forest into a syntacitc tree.

Example 4.5.11. *Some examples of syntactic trees are:*



or, analogously,



By translating the substitution operation to syntactic trees, we can significantly simplify its implementation. In the case of syntactic trees, the vertices of a forest π become the leaves of the syntactic tree $syn(\pi)$, and the substitution $\triangleright_d$ reduces to simply replacing the leaves decorated by d. A detailed discussion of the syntactic tree implementation can be found in Section 4.7.

To implement the map `syn`, we need to define grafting and concatenation operations as decorations of internal vertices of a syntactic tree. To achieve this, we wrap these operations in `Operation a` objects.

Implementation 4.5.12. Wrapping of the grafting and concatenation operations:

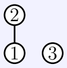
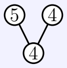
```
graftOp :: (IsVector a, Graftable a) => Operation a
graftOp = Op
    -- [...]
    $ \ops ->
        case ops of
            [x, y] -> graft x y
            _ -> error "graftOp: arity"

concatOp :: (IsVector a, Monoid a) => Operation a
concatOp = Op
    -- [...]
    $ vector . mconcat
```

We use `graftOp` and `concatOp` to define the `syn` function, which is implemented as part of the `HasSyntacticTree` type class. We note that the implementation of the `eval` function and the `HasSyntacticTree` type class is discussed in Section 4.7.

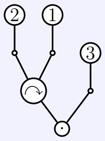
Implementation 4.5.13. Implementation of the instance of the `HasSyntacticTree` type class for forests:

```
instance
    ( IsVector t
    , IsTree t
    -- [...]
    ) => HasSyntacticTree [t]
    where
    syn [t] = case children t of
        [] -> Leaf [t]
        _ -> Node graftOp
                [syn (children t), Leaf [buildTree (root t) []]]
    syn ts = Node concatOp $ map (syn . (: [])) ts
```

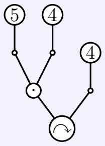
We present several examples of the usage of the map `syn` as well as `substitute` in Usage Example 4.5.14. Implementation of the `substitute` map is discussed in Section 4.7.

Usage Example 4.5.14. We define the trees  and  and compute their syntactic trees and substitution product.

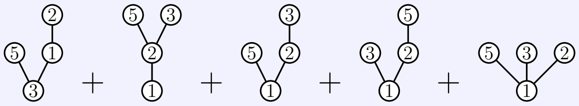
```
>>> f1 = [PT 1 [PT 2 []], PT 3 []]
>>> f2 = [PT 4 [PT 5 [], PT 4 []]]
>>> display $ vector $ syn f1
```



```
>>> display $ vector $ syn f2
```



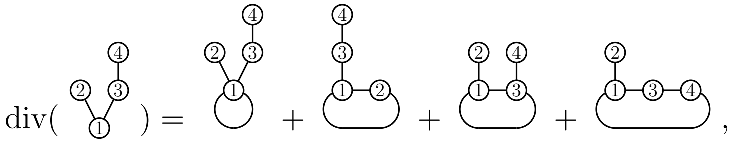
```
>>> display $ substitute [PT 4 []] (map (:[]) f1) f2
```



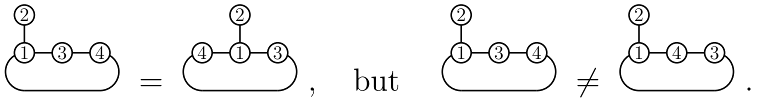
The result agrees with Example 4.5.10.

4.6 Decorated aromatic forests

Decorated aromatic forests arise in the field of geometric numerical integration, particularly when analyzing the divergence of vector fields represented by decorated rooted trees. Taking the divergence of such a vector field translates into taking the divergence of the decorated rooted trees, which involves connecting the root of the tree to each of its vertices in all possible configurations. For example,



where the cycles are oriented counterclockwise. These cycles are referred to as *aromas* and are defined as ordered forests that are invariant under cyclic permutations of their trees. This means that the aroma remains unchanged if the last tree in the cycle is moved to the first position, with all other trees shifting from position i to $i+1$. For example,



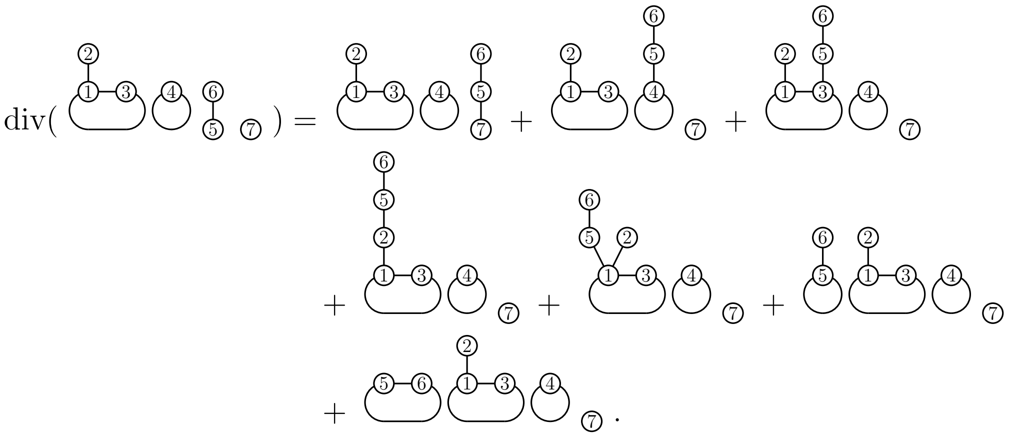
We define $\mathcal{A}$ as the vector space spanned by monomials of aromas (also referred to as *multi-aromas*), making it a free algebra of aromas. We then define aromatic forests as the tensor product $\mathcal{AF} := \mathcal{A} \otimes \mathcal{F}$. Below, we present the implementation of the planar version of aromatic forests, building on our discussion in Section 4.5.2. For a detailed discussion of decorated aromatic forests see Section 3.2.

Implementation 4.6.1. Planar implementation of the aromatic forests:

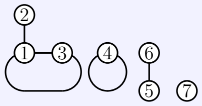
```
type PlanarAromatic t =
    ( [Cycle t]
    , [t]
    )
```

where `Cycle t` is a list of trees `[t]` invariant under cyclic permutations.

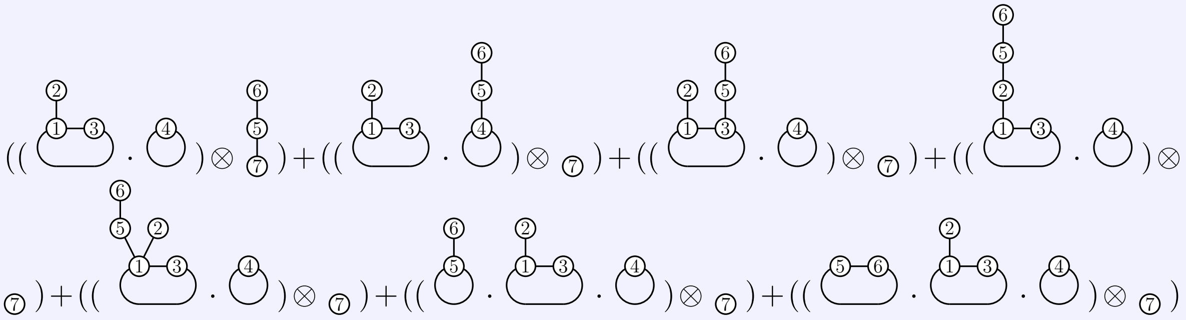
We implement an extended version of divergence, defined over planar aromatic forests, in which the leftmost root is connected to all vertices in all possible ways. For example,



This extension of divergence is studied in detail in [39]. To spare the reader from unnecessary technical details involved in handling aromas, we omit the implementation details of the divergence function.

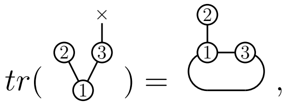
Usage Example 4.6.2. We define the planar aromatic forest  and compute its divergence.

```
>>> af = (
...        [ Cycle [PT 1 [PT 2 []], PT 3 []]
...        , Cycle [PT 4 []]
...        ]
...      , [ PT 5 [PT 6 []]
...        , PT 7 []
...        ]
...      )
>>> display $ divergence af
```



4.6.1 Grafting

We refer to [31] for a detailed study of the algebraic structure of the grafting algebra of aromatic trees. Aromas are obtained as the image of the trace function *tr*, which takes a rooted tree with a marked vertex and replaces the marked vertex with a connection to the root. For example,



where × denotes the marked vertex. Following this, we extend the grafting operation to aromatic forests as defined below.

Definition 4.6.3. The **grafting** of two aromatic forests (ω_1, π_1) and (ω_2, π_2) with $\omega_1, \omega_2 \in \mathcal{A}$ and $\pi_1, \pi_2 \in \mathcal{F}$ is defined as follows:

$$(\omega_1, \pi_1) \curvearrowright (\omega_2, \pi_2) := \sum_{(\pi_1)} (\omega_1 \cdot (\pi_1^{(1)} \curvearrowright \omega_2), \pi_1^{(2)} \curvearrowright \pi_2),$$

where the grafting of $\pi \in \mathcal{F}$ onto a multi-aroma $\omega \in \mathcal{A}$ with $\omega = \omega_1 \cdot \omega_2$ is defined as

$$\pi \curvearrowright \omega = \sum_{(\pi)} (\pi^{(1)} \curvearrowright \omega_1) \cdot (\pi^{(2)} \curvearrowright \omega_2),$$

with grafting of $\pi \in \mathcal{F}$ onto an aroma $\omega = tr(\eta)$ with $\eta \in \mathcal{F}$ defined as

$$\pi \curvearrowright tr(\eta) = tr(\pi \curvearrowright \eta).$$

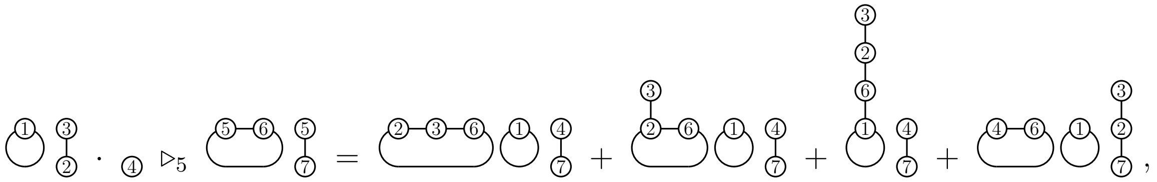
We implement the grafting operation for aromatic forests in Implementation 4.6.4 by following closely the definition in Definition 4.6.3.

Implementation 4.6.4. Implementation of grafting of aromatic forests:

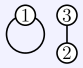
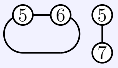
```
graftOnMultiAroma
    :: ( IsDecorated t, IsTree t, IsVector t
       -- [...]
       )
    => [t] -> [Cycle t] -> Vector (VectorScalar t) [Cycle t]
graftOnMultiAroma [] ma = vector (1 *^ ma)
graftOnMultiAroma _ [] = vector Zero
graftOnMultiAroma f [a] =
    linear ((1 *^) . (: []) . Cycle) $ (f `graft`) $ unCycle a
graftOnMultiAroma f (a : ma) =
    linear perCoproductTerm $ deshuffleCoproduct f
  where
    perCoproductTerm (x, y) =
        (x `graftOnMultiAroma` [a]) * (y `graftOnMultiAroma` ma)

instance
    ( IsDecorated t, IsTree t, IsVector t
    -- [...]
    )
    => Graftable (PlanarAromatic t)
    where
    graft (ma1, f1) (ma2, f2) =
        vector (ma1, [])
            * linear perCoproductTerm (deshuffleCoproduct f1)
      where
        perCoproductTerm (x, y) =
            linear (,[]) (x `graftOnMultiAroma` ma2)
                * linear ([],) (y `graft` f2)
```

4.6.2 Substitution

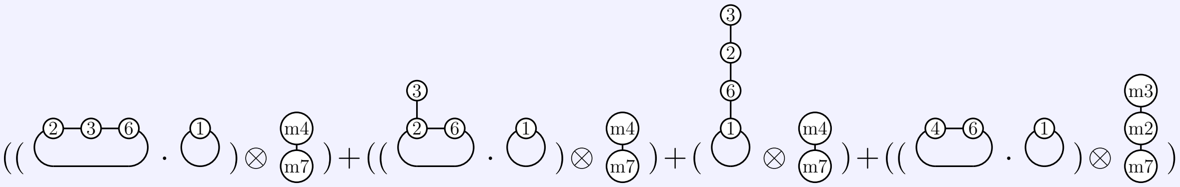
The substitution operation on aromatic forests is defined analogously to the substitution on decorated forests in Section 4.5.4. For example,



where the left operand is a clumped forest as introduced in Section 3.3.1. Following [31], we note that an aromatic forest can be decomposed into a syntactic tree using three operations: grafting, concatenation, and trace. Therefore, we implement the function `syn` using `graftOp` and `concatOp` introduced in Section 4.5.4, along with a new operation `traceOp`. We omit the technical details of the implementation of `syn` and `traceOp` for brevity.

Usage Example 4.6.5. We define the list of aromatic trees,  and , as well as the aromatic forest , and compute the corresponding substitution.

```
>>> afs = mark
...   [
...     ( [ Cycle [ PT 1 [] ] ]
...     , [ PT 2 [ PT 3 [] ] ]
...     )
...   , ( []
...     , [ PT 4 []]
...     )
...   ]
>>> af = mark
...   ( [ Cycle [ PT 5 [], PT 6 [] ] ]
...   , [ PT 7 [ PT 5 [] ] ]
...   )
>>> display $  substitute ([], [PT (Marked 5) []]) afs af
```



The result agrees with the example in Section 4.6.2. To avoid technical details, we omit the discussion of the `mark` map and `Marked` data type.

4.7 Syntactic trees

As mentioned in Section 4.5.4, representing objects as syntactic trees can simplify the implementation of operations on the objects of interest. In our case, we use syntactic trees to streamline the implementation of the substitution of (aromatic) forests; however, this approach can also be applied to implement morphisms of algebras, provided there is a systematic way to represent the objects of the algebra using a set of products and generators. In this section, we discuss the implementation of the syntactic tree data structure, the construction and evaluation of syntactic trees, and the operations performed on them.

We note that this approach is similar to that of the theory of operads, which uses trees to represent the composition of operations.

Definition 4.7.1. A **syntactic tree** is a tree whose leaves are decorated by elements of a set of generators and whose internal vertices are decorated by operations.

To allow operations to be used as decorations of internal vertices, we define the `Operation a` data type, which contains the name of the operation, its TeX representation, the arity of the operation (the number of arguments it takes), and the function that evaluates the operation on a list of elements of type `a`.

Implementation 4.7.2. We define the `Operation a` and `SyntacticTree a` data types:

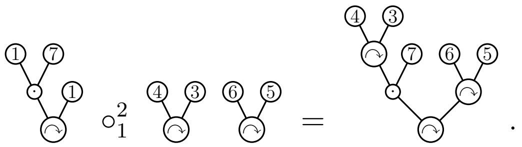
```
data Operation a = Op
    { name :: String
    , tex :: String
    , arity :: Int
    , func :: [a] -> Vector (VectorScalar a) (VectorBasis a)
    }

data SyntacticTree a
    = Node (Operation a) [SyntacticTree a]
    | Leaf a
```

All objects that can be represented using a syntactic tree must be instances of the `HasSyntacticTree` type class, which contains the function `syn`. The `syn` function constructs a syntactic tree from an object, while the `eval` function evaluates a syntactic tree to obtain the object it represents. We note that the `eval` function is already available and does not need to be defined separately for each algebra.

Implementation 4.7.3. The type class `HasSyntacticTree` wich contains `syn` and the function `eval` are implemented as follows:

```
class (IsVector a) => HasSyntacticTree a where
    syn :: a -> SyntacticTree a

eval
    :: ( IsVector a
       -- [...]
       )
    => SyntacticTree a -> Vector (VectorScalar a) a
eval (Leaf a) = vector a
eval (Node op as)
    = linear (func op) $ product $ map (linear (: []) . eval) as
```

Let the space of syntactic tree ST be endowed with a composition $\circ_d^n : ST \times ST^n \to ST$ that replaces the i-th leaf decorated by d of the first argument by the i-th syntactic tree of the second argument. For example,



We omit the implementation of the `compose` function for the sake of brevity. Its type signature is provided below. We note that the implementation is more general than the original definition. Instead of replacing only the leaves, it can also replace syntactic subtrees. Additionally, it returns `Nothing` if the number of subtrees to be replaced differs from the number of syntactic trees in the second argument.

Implementation 4.7.4. Type signature of the composition operation:

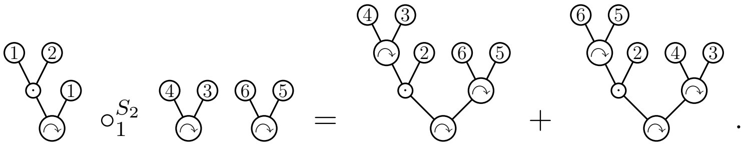
```
compose :: SyntacticTree a -> [SyntacticTree a] -> SyntacticTree a
           -> Maybe (SyntacticTree a)
```

where the first argument is the syntactic tree to be replaced, the second argument is the list of syntactic trees to replace with, and the third argument is the syntactic tree to replace in.

We go further and define symmetric composition $\circ_d^{S_n} : ST \times ST^n \to ST$ to be the composition $\circ_d^n$ which is summed over all permutations of the right operand. That is,

$$\gamma \circ_d^{S_n} (\gamma_1 \cdots \gamma_n) = \sum_{\sigma \in S_n} \gamma \circ_d^n (\gamma_{\sigma(1)} \cdots \gamma_{\sigma(n)}).$$

For example,



Implementation 4.7.5. Implementation of the symmetric composition:

```
symmetricCompose
    :: ( IsVector a
       -- [...]
       )
    => SyntacticTree a -> [SyntacticTree a] -> SyntacticTree a
       -> Vector (VectorScalar (SyntacticTree a)) (SyntacticTree a)
symmetricCompose x ops obj =
    mconcat
        $ map
            ( \perm_ops -> case compose x perm_ops obj of
                Just g -> vector (1 *^ g)
                Nothing -> vector Zero
            )
        $ permutations ops
```

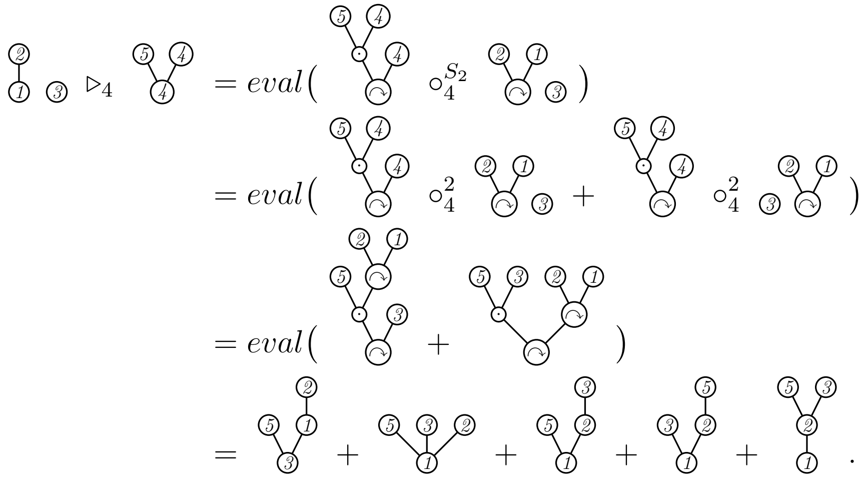
We note that once we represent the (aromatic) forests from Sections 4.5.4 and 4.6.2 as syntactic trees, the substitution operation corresponds to the symmetric composition.

Definition 4.7.6. Substitution operation $\triangleright_d$ is the operation on (aromatic) forests that corresponds to the symmetric composition on syntactic trees, that is,

$$(\tau_1 \cdots \tau_n) \triangleright_d \pi_2 = eval\big(syn(\pi_2) \circ_d^{S_n} (syn(\tau_1) \cdots syn(\tau_n))\big),$$

if π_2 has exactly n vertices decorated by d. Otherwise, $(\tau_1 \cdots \tau_n) \triangleright_d \pi_2 = 0$.

Example 4.7.7. *Let us compute the substitution seen in Example 4.5.10 step by step:*



Implementation 4.7.8. Implementation of the substitution operation:

```
substitute
    :: ( HasSyntacticTree a
       , IsVector a
       -- [...]
       )
    => a -> [a] -> a -> Vector (VectorScalar a) a
substitute x gens obj =
    linear eval $ symmetricCompose (syn x) (map syn gens) $ syn obj
```

Chapter 5

Conclusion and ongoing work

In this chapter, we summarize the main contributions of this thesis, give a brief outline of the ongoing projects, and provide an outlook on future work.

In this thesis, we introduced exotic S-series and used them to derive order conditions in the weak sense and for invariant measure sampling. Leveraging the framework of exotic S-series, we proposed a novel generalization of the Leimkuhler-Matthews scheme to address problems with position-dependent diffusion. We analyzed its stability and conducted numerical experiments to confirm its order. However, the method has not yet been applied to a real-world problem that would fully exploit its advantages, which remains a task for future work.

Additionally, we formulated a formal algorithm for generating order conditions for invariant measure sampling. By combining the algorithm's properties with the algebraic properties of exotic forests, we demonstrated that order conditions corresponding to exotic forests are automatically satisfied if the order conditions for trees are met, significantly reducing the number of conditions that need to be considered. The algorithm is currently defined for Langevin dynamics with additive noise. Extending it to the case of multiplicative noise remains an open problem for future research.

We also studied decorated aromatic forests, proving that they form both a Grossman-Larson Hopf algebroid, as defined in [6], and a pre-Hopf algebroid. While we provide an explicit definition of the left bialgebroid and the antipode, an explicit formulation of the right bialgebroid, as well as a deeper understanding of its role in numerical analysis, is still lacking. Another promising direction is to explore the application of the pre-Hopf algebroid structure in the context of solutions to the Yang-Baxter equation as well as the description of the Butcher-Connes-Kreimer structure adjoint to the Grossman-Larson Hopf algebroid and the generalization of the Butcher group.

To extend this framework, we introduced clumped forests, a necessary component in defining the substitution law, and explored their relationship with aromatic forests. We proposed the multi-pre-Lie family of insertion algebras, which enabled us to define the substitution law on clumped forests and extend it to aromatic forests. Furthermore, we analyzed the impact of moving to the context of exotic aromatic forests, describing the corresponding composition and substitution laws.

Recognizing the increasing complexity of the forest formalism in the stochastic context and with the inclusion of aromas, we developed a Haskell package to automate computations involving algebras of graphs. This package was designed with simplicity and extensibility in mind, accommodating the diverse variations of the forest formalism. As a result, it is applicable across various mathematical fields where algebras over tree- and forest-like graphs arise. The accompanying manual provides a detailed overview of the package's structure and its usage for implementing algebras. Currently, the package includes an implementation of decorated aromatic forests, along with grafting and insertion operations. However, the implementation of exotic aromatic forests, as well as composition and substitution laws, remains for future work. To facilitate the implementation of the substitution law, it is necessary to derive a recursive formula for its computation. Alternatively, an approach based on dualization could be used to derive coproducts from products automatically, which would significantly simplify their implementation, though at the cost of computational efficiency. Other potential developments include the construction of Butcher tableaus for Runge-Kutta methods and their analysis, the generation of order conditions for invariant measure sampling using exotic forests, and the implementation of algebras over planar forests using the recursive formulas introduced in [59].

Convergence analysis of Spectral Deferred Correction, *in preparation [10], in collaboration with Joscha Fregin, Daniel Ruprecht, and Gilles Vilmart.*

The Spectral Deferred Correction (SDC) method [29, 24, 25] is a time-stepping approach that iteratively improves the accuracy of a numerical solution. This is achieved through the use of low-order methods, known as preconditioners, which are computationally inexpensive. These preconditioners approximate the error in the numerical solution, enabling its correction and progressively enhancing the solution's accuracy.

The primary goal of this project is to represent SDC methods as Runge-Kutta methods and leverage Butcher series to analyze their convergence. Using the framework of rooted trees, we prove that each iteration of the method increases the order of accuracy by one, regardless of the choice of preconditioners, which do not even need to be consistent. Furthermore, we investigate the phenomenon of order jumps—where the order of the integrator increases by more than one in a single iteration—using the rooted tree framework. Since SDC methods are often employed as computationally efficient approximations of expensive collocation methods, we analyze these order jumps by examining the underlying structure and behavior of collocation methods.

Our analysis leads to the introduction of a novel preconditioner. This preconditioner can be computed in parallel and achieves an order two jump per iteration when the SDC method approximates a collocation method.

Planar exotic forests and integrators on manifolds, *in preparation [8], in collaboration with Adrien Busnot Laurent and Baptiste Huguet.*

We introduce a general framework for analyzing novel stochastic frozen flow methods of any order on any manifold in the weak sense, using a new algebraic formalism analogous to Butcher series. This approach builds on Lie group methods [57, 58] as well as Crouch-Grossman and commutator-free methods [27, 62] in the deterministic setting. The proposed methods offer improvements over existing approaches in terms of accuracy, versatility, and computational efficiency.

The calculation of order conditions is complex and necessitates appropriate algebraic tools. To address this, we propose a novel extension of exotic series and Lie series, referred to as exotic Lie series, for the systematic computation of order conditions for the new frozen flow methods at any order in the weak sense.

We extend the exotic forests formalism by incorporating planarity, introducing planar exotic forests. While the algebraic formalism serves primarily as a tool to simplify calculations in this context, it also establishes strong connections to several active research areas, including numerical methods on manifolds, differential geometry, stochastic processes, post-Lie and Hopf algebras, and rough paths. These connections open promising avenues for future research.

We explore the algebraic structure of planar exotic forests, focusing on their D-algebra, Hopf algebra, and post-Hopf algebra structures. Furthermore, we relate these structures to the composition of numerical methods and the derivation of order conditions in the weak sense.

Appendix

List of grafted and exotic trees up to size 3

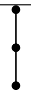
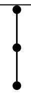
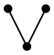
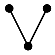
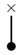
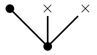
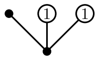
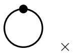
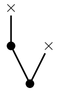
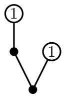
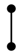
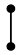
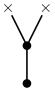
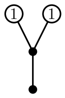
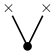
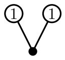
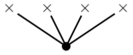
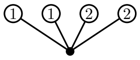
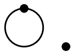
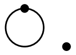
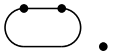
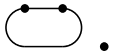
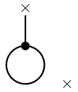
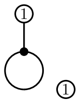
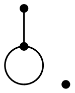
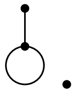
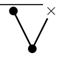
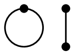
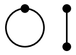
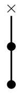
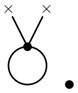
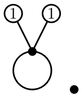
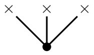
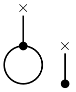
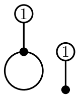
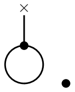
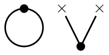
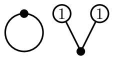
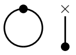
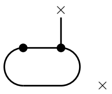
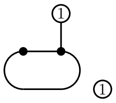
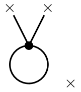
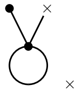
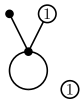
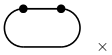
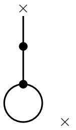
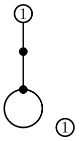
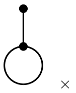
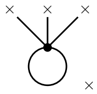
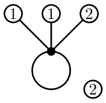
$\|\tau\|$	(τ,α_g)	$\sigma(\tau,\alpha_g)$	(τ,α_e)	$\sigma(\tau,\alpha_e)$	$\|\tau\|$	(τ,α_g)	$\sigma(\tau,\alpha_g)$	(τ,α_e)	$\sigma(\tau,\alpha_e)$
0.5		1			3		1		1
1		1		1			2		2
1.5		1					2		2
		1					1		1
2		1		1			2		2
		2		2			24		8
		1		1			2		2
		1		1			1		1
2.5		1					1		1
		1					2		2
		6					1		1
		1					2		2
		1					1		1
		2					1		1
		2					1		1
		1					6		2

Table .0.1: List of grafted and exotic trees up to size $|\tau| = 3$ with their symmetries σ as described in Section 1.4.

Order conditions for the invariant measure up to order 3

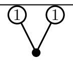
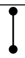
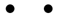
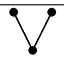
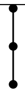
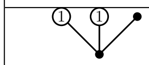
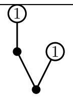
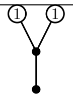
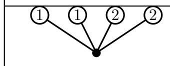
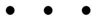
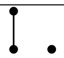
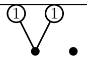
	π	$\omega(\pi)$
order 1		$\sum_{i=1}^{s} b_i - 1$
order 2		$\sum b_i d_i^2 - \frac{1}{2} + \sum b_i - 2\sum b_i d_i$
		$\sum b_i a_{ij} - \frac{1}{2} + \sum b_i - 2\sum b_i d_i$
$*$		$\sum b_i b_j + 1 - 2\sum b_i$
order 3		$-4\sum b_i d_i a_{ij} + \sum b_i + \sum b_i a_{ij} a_{ik} - 4\sum b_i d_i - \frac{1}{3} + 4\sum b_i d_i^2 + 2\sum b_i a_{ij}$
		$2\sum b_i d_i b_j + \frac{3}{2}\sum b_i - \sum b_i b_j - \sum b_i d_i b_j d_j + \sum b_i a_{ij} a_{jk} - 2\sum b_i d_i - \frac{1}{2} + \sum b_i a_{ij} - 2\sum b_i a_{ij} d_j$
		$-2\sum b_i d_i a_{ij} + \sum b_i + \sum b_i d_i^2 a_{ij} - 4\sum b_i d_i - \frac{1}{3} + 5\sum b_i d_i^2 - 2\sum b_i d_i^3 + \sum b_i a_{ij}$
		$\sum b_i d_i b_j - \sum b_i d_i a_{ij} + \sum b_i - \frac{1}{2}\sum b_i b_j - \frac{1}{2}\sum b_i d_i b_j d_j - 2\sum b_i d_i - \frac{1}{3} + \sum b_i d_i^2 + \sum b_i a_{ij} - \sum b_i a_{ij} d_j + \sum b_i d_i a_{ij} d_j$
		$2\sum b_i d_i b_j + \frac{3}{2}\sum b_i - \sum b_i b_j - \sum b_i d_i b_j d_j - 2\sum b_i d_i - \frac{1}{2} + \sum b_i a_{ij} - 2\sum b_i a_{ij} d_j + \sum b_i a_{ij} d_j^2$
		$\sum b_i - 4\sum b_i d_i + 6\sum b_i d_i^2 - \frac{1}{3} - 4\sum b_i d_i^3 + \sum b_i d_i^4$
$*$		$\sum b_i b_j b_k + 3\sum b_i - 3\sum b_i b_j - 1$
$*$		$\sum b_i a_{ij} b_j - 2\sum b_i d_i b_j - \frac{3}{2}\sum b_i + \sum b_i b_j + 2\sum b_i d_i - \sum b_i a_{ij} + \frac{1}{2}$
$*$		$\sum b_i d_i^2 b_j - 2\sum b_i d_i b_j - \frac{3}{2}\sum b_i + \sum b_i b_j + 2\sum b_i d_i - \sum b_i d_i^2 + \frac{1}{2}$

Table .0.2: List of the order conditions for stochastic Runge-Kutta methods up to order 3 generated by the Algorithm 1. The rows marked by $*$ correspond to the order conditions which are automatically satisfied using the Theorem 3.1.8.

List of Figures

Bibliography

[1] A. Abdulle, I. Almuslimani, and G. Vilmart. Optimal explicit stabilized integrator of weak order 1 for stiff and ergodic stochastic differential equations. *SIAM/ASA J. Uncertain. Quantif.*, 6(2):937–964, 2018. ISSN: 2166-2525.

[2] A. Abdulle, G. Vilmart, and K. C. Zygalakis. High order numerical approximation of the invariant measure of ergodic SDEs. *SIAM J. Numer. Anal.*, 52(4):1600–1622, 2014. ISSN: 0036-1429.

[3] A. Abdulle, G. Vilmart, and K. C. Zygalakis. Long time accuracy of Lie-Trotter splitting methods for Langevin dynamics. *SIAM J. Numer. Anal.*, 53(1):1–16, 2015. ISSN: 0036-1429.

[4] A. Abdulle, G. Vilmart, and K. C. Zygalakis. Weak second order explicit stabilized methods for stiff stochastic differential equations. *SIAM J. Sci. Comput.*, 35(4):A1792–A1814, 2013. ISSN: 1064-8275,1095-7197.

[5] G. Bogfjellmo. Algebraic structure of aromatic B-series. *J. Comput. Dyn.*, 6(2):199–222, 2019. ISSN: 2158-2491.

[6] G. Böhm. Hopf algebroids, 2008. arXiv: `0805.3806 [math.QA]`.

[7] E. Bronasco. Exotic B-series and S-series: algebraic structures and order conditions for invariant measure sampling. *To appear in Found. Comput. Math.*, 2025. (published electronically).

[8] E. Bronasco, A. Busnot Laurent, and B. Huguet. High order integration of stochastic dynamics on Riemannian manifolds with frozen flow methods, 2025. In preparation.

[9] E. Bronasco, J.-L. Falcone, and G. Vilmart. A symbolic package for the implementation of algebras of forest-like graphs, 2025. In preparation.

[10] E. Bronasco, J. Fregin, D. Ruprecht, and G. Vilmart. Convergence analysis of Spectral Deferred Correction methods using Butcher series, 2025. In preparation.

[11] E. Bronasco and A. Laurent. Hopf algebra structures for the backward error analysis of ergodic stochastic differential equations. *Submitted*, 2024.

[12] E. Bronasco, B. Leimkuhler, D. Phillips, and G. Vilmart. Efficient Langevin sampling with position-dependent diffusion. *Submitted*, 2024. arXiv: `2501.02943 [math.NA]`.

[13] Y. Bruned and D. Manchon. Algebraic deformation for (S)PDEs. *J. Math. Soc. Japan*, 75(2):485–526, 2023. ISSN: 0025-5645,1881-1167.

[14] K. Burrage and P. M. Burrage. Order conditions of stochastic Runge-Kutta methods by B-series. *SIAM J. Numer. Anal.*, 38(5):1626–1646, 2000. ISSN: 0036-1429.

[15] K. Burrage, P. M. Burrage, and T. Tian. Numerical methods for strong solutions of stochastic differential equations: an overview. In *Stochastic analysis with applications to mathematical finance*. Volume 460, number 2041, pages 373–402. Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 2004.

[16] J. C. Butcher. Coefficients for the study of Runge-Kutta integration processes. *J. Austral. Math. Soc.*, 3:185–201, 1963. ISSN: 0263-6115.

[17] J. C. Butcher. The effective order of Runge-Kutta methods. In *Conf. on Numerical Solution of Differential Equations (Dundee, 1969)*, pages 133–139. Springer, Berlin, 1969.

[18] 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. *Adv. in Appl. Math.*, 47(2):282–308, 2011. ISSN: 0196-8858.

[19] M. P. Calvo, A. Murua, and J. M. Sanz-Serna. Modified equations for ODEs. In *Chaotic numerics (Geelong, 1993)*. Volume 172, Contemp. Math. Pages 63–74. Amer. Math. Soc., Providence, RI, 1994.

[20] P. Cayley. On the Analytical Forms Called Trees. *Amer. J. Math.*, 4(1-4):266–268, 1881. ISSN: 0002-9327,1080-6377.

[21] F. Chapoton and M. Livernet. Pre-Lie algebras and the rooted trees operad. *Internat. Math. Res. Notices*, (8):395–408, 2001. ISSN: 1073-7928,1687-0247.

[22] P. Chartier, E. Hairer, and G. Vilmart. Numerical integrators based on modified differential equations. *Math. Comp.*, 76(260):1941–1953, 2007. ISSN: 0025-5718,1088-6842.

[23] P. Chartier and A. Murua. Preserving first integrals and volume forms of additively split systems. *IMA J. Numer. Anal.*, 27(2):381–405, 2007. ISSN: 0272-4979,1464-3642.

[24] A. Christlieb, B. Ong, and J.-M. Qiu. Comments on high-order integrators embedded within integral deferred correction methods. *Comm. App. Math. Comp. Sc.*, 4(1):27–56, 2009.

[25] A. Christlieb, B. W. Ong, and J.-M. Qiu. Integral deferred correction methods constructed with high order Runge-Kutta integrators. *Math. Comp.*, 79:761–783, 2010.

[26] A. Connes and D. Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Comm. Math. Phys.*, 199(1):203–242, 1998. ISSN: 0010-3616.

[27] P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *J. Nonlinear Sci.*, 3:1–33, 1993.

[28] A. Debussche and E. Faou. Weak backward error analysis for SDEs. *SIAM J. Numer. Anal.*, 50(3):1735–1752, 2012. ISSN: 0036-1429.

[29] A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numer. Math.*, 40(2):241–266, 2000.

[30] A. Einstein. Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen. *Annalen der Physik*, 322(8):549–560, 1905. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.19053220806`.

[31] G. Fløystad, D. Manchon, and H. Z. Munthe-Kaas. The universal pre-Lie-Rinehart algebras of aromatic trees. In *Geometric and harmonic analysis on homogeneous spaces and applications*. Volume 366, Springer Proc. Math. Stat. Pages 137–159. Springer, Cham, [2021] ©2021.

[32] L. Foissy. Algebraic structures on typed decorated rooted trees. *SIGMA Symmetry Integrability Geom. Methods Appl.*, 17:Paper No. 086, 28, 2021. ISSN: 1815-0659.

[33] E. Hairer and G. Wanner. On the Butcher group and general multi-value methods. *Computing (Arch. Elektron. Rechnen)*, 13(1):1–15, 1974.

[34] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006, pages xviii+644. ISBN: 3-540-30663-3; 978-3-540-30663-4. Structure-preserving algorithms for ordinary differential equations.

[35] D. J. Higham. Mean-square and asymptotic stability of the stochastic theta method. *SIAM J. Numer. Anal.*, 38(3):753–769, 2000. ISSN: 0036-1429,1095-7170.

[36] M. E. Hoffman. Combinatorics of rooted trees and Hopf algebras. *Trans. Amer. Math. Soc.*, 355(9):3795–3811, 2003. ISSN: 0002-9947,1088-6850.

[37] A. Iserles, G. R. W. Quispel, and P. S. P. Tse. B-series methods cannot be volume-preserving. *BIT Numer. Math.*, 47(2):351–378, 2007. ISSN: 0006-3835.

[38] D. I. Ketcheson and H. Ranocha. Computing with B-series. *ACM Trans. Math. Softw.*, 49(2), 2023.

[39] A. Laurent, R. I. McLachlan, H. Z. Munthe-Kaas, and O. Verdier. The aromatic bicomplex for the description of divergence-free aromatic forms and volume-preserving integrators. *Forum Math. Sigma*, 11:Paper No. e69, 2023. ISSN: 2050-5094.

[40] A. Laurent and H. Munthe-Kaas. The universal equivariance properties of exotic aromatic B-series. *To appear in Found. Comp. Math.*, 2024. ISSN: 1615-3383. (published electronically).

[41] A. Laurent and G. Vilmart. Exotic aromatic B-series for the study of long time integrators for a class of ergodic SDEs. *Math. Comp.*, 89(321):169–202, 2020. ISSN: 0025-5718.

[42] A. Laurent and G. Vilmart. Exotic aromatic B-series for the study of long time integrators for a class of ergodic SDEs. *Math. Comp.*, 89(321):169–202, 2020. ISSN: 0025-5718,1088-6842.

[43] A. Laurent and G. Vilmart. Order conditions for sampling the invariant measure of ergodic stochastic differential equations on manifolds. *Found. Comput. Math.*, 22(3):649–695, 2022. ISSN: 1615-3375.

[44] B. Leimkuhler, C. Matthews, and M. V. Tretyakov. On the long-time integration of stochastic gradient systems. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 470(2170):20140120, 16, 2014. ISSN: 1364-5021,1471-2946.

[45] B. Leimkuhler and C. Matthews. Rational construction of stochastic numerical methods for molecular sampling. *Appl. Math. Res. Express. AMRX*, 2013(1):34–56, 2013. ISSN: 1687-1200.

[46] T. Lelièvre, G. A. Pavliotis, G. Robin, R. Santet, and G. Stoltz. Optimizing the diffusion coefficient of overdamped Langevin dynamics. *Preprint*, 2024. arXiv:2404.12087 (76 pages).

[47] Y. Li, Y. Sheng, and R. Tang. Post-Hopf algebras, relative Rota-Baxter operators and solutions of the Yang-Baxter equation. *J. Noncomm. Geom.*, 2023.

[48] J.-L. Loday and M. Ronco. Combinatorial Hopf algebras. In *Quanta of maths*. Volume 11, Clay Math. Proc. Pages 347–383. Amer. Math. Soc., Providence, RI, 2010. ISBN: 978-0-8218-5203-3.

[49] J.-H. Lu. Hopf algebroids and quantum groupoids. *Internat. J. Math.*, 7(1):47–70, 1996. ISSN: 0129-167X,1793-6519.

[50] A. Lundervold and H. Munthe-Kaas. Backward error analysis and the substitution law for Lie group integrators. *Found. Comput. Math.*, 13(2):161–186, 2013. ISSN: 1615-3375,1615-3383.

[51] D. Manchon and A. Saïdi. Lois pré-Lie en interaction. *Comm. Algebra*, 39(10):3662–3680, 2011. ISSN: 0092-7872.

[52] J. C. Mattingly, A. M. Stuart, and D. J. Higham. Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise. *Stochastic Process. Appl.*, 101(2):185–232, 2002. ISSN: 0304-4149,1879-209X.

[53] G. N. Milstein and M. V. Tretyakov. *Stochastic numerics for Mathematical Physics*. Scientific Computing. Springer-Verlag, Berlin and New York, 2004.

[54] I. Moerdijk and J. Mrčun. On the universal enveloping algebra of a Lie algebroid. *Proc. Amer. Math. Soc.*, 138(9):3135–3145, 2010. ISSN: 0002-9939,1088-6826.

[55] G. J. Moro and F. Cardin. Saddle point avoidance due to inhomogeneous friction. *Chem. Phys.*, 235(1):189–200, September 1998. ISSN: 0301-0104.

[56] H. Z. Munthe-Kaas and W. M. Wright. On the Hopf algebraic structure of Lie group integrators. *Found. Comput. Math.*, 8(2):227–257, 2008. ISSN: 1615-3375.

[57] H. Munthe-Kaas. Lie-Butcher theory for Runge-Kutta methods. *BIT Num. Math.*, 35:572–587, 1995.

[58] H. Munthe-Kaas and A. Zanna. Numerical integration of differential equations on homogeneous manifolds. In *Foundations of Computational Mathematics: Selected Papers of a Conference Held at Rio de Janeiro, January 1997*, pages 305–315. Springer, 1997.

[59] H. Z. Munthe-Kaas and K. K. Føllesdal. Lie-Butcher series, geometry, algebra and computation. In *Discrete mechanics, geometric integration and Lie-Butcher series*. Volume 267, Springer Proc. Math. Stat. Pages 71–113. Springer, Cham, 2018.

[60] A. Murua. Formal series and numerical integrators, part I: Systems of ODEs and symplectic integrators. *Appl. Num. Math.*, 29:221–251, 1999.

[61] J.-M. Oudom and D. Guin. On the Lie enveloping algebra of a pre-Lie algebra. *J. K-Theory*, 2(1):147–167, 2008. ISSN: 1865-2433,1865-5394.

[62] B. Owren and A. Marthinsen. Runge-Kutta methods adapted to manifolds and based on rigid frames. *BIT*, 39(1):116–142, 1999. ISSN: 0006-3835.

[63] G. A. Pavliotis. *Stochastic processes and applications*, volume 60 of *Texts in Applied Mathematics*. Springer, New York, 2014, pages xiv+339. ISBN: 978-1-4939-1322-0; 978-1-4939-1323-7. Diffusion processes, the Fokker-Planck and Langevin equations.

[64] D. Phillips, B. Leimkuhler, and C. Matthews. Numerics with coordinate transforms for efficient Brownian dynamics simulations. *Mol. Phys.*, 0(0):e2347546, 2024. ISSN: 0026-8976.

[65] H. Ranocha and D. I. Ketcheson. BSeries.jl: Computing with B-series in Julia. `https://github.com/ranocha/BSeries.jl`, 2021.

[66] H. Ranocha and D. I. Ketcheson. Bseries.py. `https://github.com/ketch/BSeries`, 2021.

[67] A. Rößler. Runge-Kutta methods for Stratonovich stochastic differential equation systems with commutative noise. In *Proceedings of the 10th International Congress on Computational and Applied Mathematics (ICCAM-2002)*, volume 164/165, pages 613–627, 2004.

[68] A. Rößler. Second order Runge-Kutta methods for Itô stochastic differential equations. *SIAM J. Numer. Anal.*, 47(3):1713–1738, 2009. ISSN: 0036-1429,1095-7170.

[69] A. Saïdi. *Algèbres de Hopf d'arbres et structures pré-Lie*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2011.

[70] A. Saïdi. On a pre-Lie algebra defined by insertion of rooted trees. *Lett. Math. Phys.*, 92(2):181–196, 2010. ISSN: 0377-9017.

[71] Y. Saito and T. Mitsui. Mean-square stability of numerical schemes for stochastic differential systems. *Vietnam J. Math.*, 30:551–560, 2002. ISSN: 0866-7179.

[72] J. M. Sanz-Serna. Markov chain Monte Carlo and numerical differential equations. In *Current challenges in stability issues for numerical differential equations*. Volume 2082, Lecture Notes in Math. Pages 39–88. Springer, Cham, 2014.

[73] H. S. Sundklakk. *A library for computing with trees and B-Series*. Master's thesis, NTNU, 2015. Supervisor: B. Owren.

[74] H. S. Sundklakk. Pybs. `https://github.com/henriksu/pybs`, 2015.

[75] D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Anal. Appl.*, 8(4):483–509, 1990. ISSN: 0736-2994,1532-9356.

[76] X. Tang and A. Xiao. Efficient weak second-order stochastic Runge-Kutta methods for Itô stochastic differential equations. *BIT*, 57(1):241–260, 2017. ISSN: 0006-3835,1572-9125.

[77] A. Tocino. Mean-square stability of second-order Runge-Kutta methods for stochastic differential equations. *J. Comput. Appl. Math.*, 175(2):355–367, 2005. ISSN: 0377-0427,1879-1778.

[78] A. Tygliyan and S. Filippov. Elementary differentials, their graphs and codes. *Math. Models Comp. Sim.*, 2:156–161, 2010.

[79] G. Vilmart. Postprocessed Integrators for the High Order Integration of Ergodic SDEs. *SIAM J. Sci. Comput.*, 37(1):A201–A220, 2015.

Abstract

We develop the framework of exotic forests to study the algebraic structures underlying numerical integrators for sampling the invariant measure of Langevin dynamics. Leveraging these algebraic structures, we introduce a formal algorithm for generating order conditions for invariant measure sampling and prove that a large subset of these conditions is satisfied automatically. Building on these insights, we derive an efficient second-order integrator for sampling the invariant measure of Langevin dynamics with position-dependent diffusion.

We then consider a broader set of exotic forests that includes graph structures known as aromas and stolons, which play a crucial role in geometric numerical integration. We analyze their associated composition and substitution laws, enabling the study of integrator composition, post-processing, backward error analysis, and modified equation techniques within the Butcher series framework. Given the rapidly increasing complexity of the forest structures we consider, we develop a Haskell package to automate computations involving algebras over forests.

Résumé de la thèse

Nous développons les forêts exotiques d'arbres pour étudier les structures algébriques associées aux intégrateurs numériques utilisés dans l'échantillonnage de la mesure invariante de la dynamique de Langevin. Nous introduisons un algorithme formel, utilisant ces structures, pour générer les conditions d'ordre de ces intégrateurs numériques et prouvons qu'un grand sous-ensemble de ces conditions est automatiquement satisfait. Inspirés par ces développements, nous construisons un intégrateur d'ordre deux efficace pour l'échantillonnage de la mesure invariante de la dynamique de Langevin avec une diffusion dépendante de la position.

Ensuite, nous considérons un ensemble plus général de forêts exotiques qui inclut d'autres structures de graphes : les arômes et les stolons. Ces derniers jouent un rôle crucial dans le domaine de l'intégration numérique géométrique. Nous analysons les lois de composition et de substitution qui leur sont associées. Dans le cadre des séries de Butcher, ceci nous permet d'étudier la composition d'intégrateurs, le "post-traitement", l'analyse d'erreur rétrograde ainsi que les équations modifiées. Étant donnée la complexité croissante des structures considérées, nous développons un package en Haskell pour automatiser les calculs sur les algèbres définies sur l'espace des forêts.