



Article Set-Based B-Series

Julien Alexandre dit Sandretto 回

U2IS, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France; alexandre@ensta.fr

Abstract: B-series were defined to unify the formalism of solutions for ordinary differential equations defined by series. Runge–Kutta schemes can be seen as truncated B-series, similar to Taylor series. In the prolific domain of reachability analysis, i.e., the process of computing the set of reachable states for a system, many techniques have been proposed without obvious links. In the particular case of uncertain initial conditions and/or parameters in the definition of differential equations, set-based approaches are a natural and elegant method to compute reachable sets. In this paper, an extension to B-series is proposed to merge these techniques in a common formalism—named set-based B-series. We show that the main properties of B-series are preserved. A validated technique, based on Runge–Kutta methods, able to compute such series, is presented. Experiments are provided in order to illustrate the proposed approach.

Keywords: differential equations; Runge-Kutta series; Taylor series; set-based methods

MSC: 34G20; 34A06; 52A27; 65D30

1. Introduction and Related Works

Many scientific applications in physical fields, such as mechanics, robotics, chemistry and electronics, require solving differential equations. This particular type of equation appears, for example, when the location is required; however, only the velocity and/or the acceleration are available when modeling a system. In the general case, these differential equations cannot be formally integrated—that is to say, closed form solutions are not available, and a numerical integration scheme is used to approximate the state of the system. Let us consider the most classical continuous differential equations, the Ordinary Differential Equations (ODEs):

$$\dot{y}(t) = f(y(t)) . \tag{1}$$

One equation as above has a unique solution y(t) if certain hypotheses occur: f is Lipschitz, and the associated vector field is smooth. The most classical approach is to use a Runge–Kutta scheme—carefully chosen with respect to the problem, desired accuracy, etc.—to simulate the system behavior.

Historically, the first method for numerical solutions of differential equations was proposed by Euler in *Institutiones Calculi Integralis* [1]. His main idea is based on a simple principle: if a particle is located at y_0 at time t_0 and if its velocity at this time is known to be equal to v_0 , then, at time t_1 , the particle will be at approximately position $y_1 = y_0 + (t_1 - t_0)v_0$ under the condition that t_1 is sufficiently close to t_0 (that is, after a short time), and thus the velocity does not change "too much" over $[t_0, t_1]$. Based on this principle, around the year 1900, C. Runge and M. W. Kutta developed a family of iterative methods, now called Runge–Kutta methods. While many such methods have been proposed since then, a unified formalism and a deep analysis was first proposed by John Butcher in the 1960s [2].

The requirement of an unified and elegant formalism led to what we now call the Butcher series, or B-series (in respect to the seminal paper of John Butcher [3]). Indeed, from the works of Cayley [4] and Merson [5], it has been shown that the expressions appearing in



Citation: Alexandre dit Sandretto, J. Set-Based B-Series. *Mathematics* 2022, 10, 3165. https://doi.org/10.3390/ math10173165

Academic Editor: Xiangmin Jiao

Received: 19 July 2022 Accepted: 29 August 2022 Published: 2 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). the derivatives of the solution of Equation (1), $\ddot{y} = (f'f)(y)$, $\ddot{y} = (f''(f,f))(y) + (f'f'f)(y)$, are similar to rooted trees [6]. Therefore, considering the formal series, such as

$$B(a, hf, y) = a(\emptyset)y + ha(\bullet)f(y) + h^2a(\bullet)(f'f)(y) + \frac{h^3}{2}a(\bullet \bullet)(f''(f, f))(y) + \dots$$
(2)

is natural and leads to express the solution of Equation (1) under this form with the series coefficients $a(\emptyset) = a(\bullet) = 1$, $a(\bullet) = 1/2$, $a(\bullet) = 1/3$, etc. These series, with arbitrary coefficients, are named B-series [7]. B-series are then the more general formalism to express computable solutions of ODEs. It is then a natural choice to define our set-based formalism.

Models of physical systems defined with the help of ODEs are, in general, simplified and, thus, approximate the actual behavior of such systems. It is, then, common to consider the uncertainties on the parameters to improve the representativeness of the model. These uncertainties can be defined by probabilistic methods or enclosed in sets. If the first technique is the most common—with Markov Chains [8], Monte-Carlo approaches [9], Graph theory [10], etc.—the latter is more interesting in a verification procedure due to the guarantees provided (for safety, for example). Uncertainties, if bounded, can be enclosed in sets (for example, intervals), such that the models obtained are guaranteed to contain the actual ones [11]. Other sets have been considered, such as polytopes, zonotopes (particular polytopes) and ellipsoids.

In the special case of ODEs, when parameters are given under the form of intervals, the computation of solutions is called reachability, i.e., the goal is then to compute the sets that enclose the solution of an ODE for all time and all possible values of the parameters.

Set-membership methods, associated with interval analysis, lead one to producing validated numerical integrations of ODEs. Validated numerical integration is an appealing approach to produce rigorous bounds of the solution of ODEs. Most of the rigorous techniques defined thus far, since Moore's first algorithm based on Euler's method [12], are based on a Taylor series approach [13]; see, for example, [14–17] and the references therein. Nevertheless, it is unlikely that only one kind of methods is adapted to all various classes of ODEs. Alternative recent works [18–23] deal with the adaptation of Runge–Kutta methods for ODEs and Differential-Algebraic Equations (DAEs) [24].

Currently, important efforts are being made regarding the implementation of set-based methods for differential equation simulation. This leads to the releases of several tools. The most efficient, promising and innovative tools participate each year in the ARCH competition [25,26]: Ariadne, which adopts an approach with compositionality [27]; CORA, which proposes various set representations [28,29]; DynIbex using Runge–Kutta methods and constraint programming [23]; JuliaReach, which defines sets using Taylor models [30]; Kaa based on parallelotope bundles [31]; and KeYmaera X, a theorem prover able to consider differential Equations [32].

In this article, we propose a unification of these validated numerical integration methods using B-series. B-series were introduced by E. Hairer and G. Wanner in [7] after the seminal work of J. Butcher [2]. As said before, they follow the results from Cayley [4] and Merson [5] on the expressions arising when studying the derivatives of the solution of an ODE. Since then, B-series have been intensely studied (see, e.g., [6,33] or [34]); however, no papers refer to set-based B-series.

Outline.

We review the theory of trees, B-series and Runge–Kutta methods, in Section 2. In Section 3, we present the set-based framework proposed in this paper. In Section 4, we show how validate set-based B-series using Runge–Kutta schemes. A small experiment is presented in Section 5 to exhibit a result. In Section 6, we summarize the main contributions of the paper.

2. Theories behind B-Series

As said in the introduction, B-series are strongly connected to rooted trees, and thus they are recalled in this section, followed by B-series and a Runge–Kutta overview.

2.1. Rooted Trees

A rooted tree [35] is a graph with a distinguished node, called the root, in which every other node is connected to the root by a unique path. If the root of a tree τ is removed and the nodes connected to the original node by a single bond are taken as new roots, the tree τ breaks up into rooted trees τ_1, τ_2, \ldots . Reversing this process, a new tree $\tau = [\tau_1, \tau_2, \ldots]$ can be constructed by joining the roots of the trees to a new common root. The number of nodes in a tree is denoted by $|\tau|$ (order).

$$\mathcal{T} = \{ \varnothing \} \cup \{ [au_1 au_2 \dots au_m] : au_i \in \mathcal{T} \}$$
.

The number of equivalence classes of heap-orderings on a particular tree is denoted by $\alpha(\tau)$ and can be computed using the Butcher's formula:

$$\alpha(\tau) = \frac{|\tau|!}{\tau!|S_{\tau}|},\tag{3}$$

where S_{τ} denotes the symmetry group of τ , and the tree factorial is defined recursively by

$$[\tau_1,\ldots,\tau_n]!=|[\tau_1,\ldots,\tau_n]|\cdot\tau_1!\cdots\tau_n!, \qquad (4)$$

with the tree factorial of an isolated root defined to be 1

$$! = 1$$
. (5)

We denote, by $\sigma(\tau)$, the symmetry of the tree τ .

2.2. Initial Value Problems

An initial value problem (IVP) consists on the merge of an ODE and a value, while an ODE has a solution (a function of time) only when a formal expression y(t) such that $dy(t)/dt = \dot{y}(t) = f(y(t))$ exists, an IVP has always a solution (if some hypotheses on the flow are respected).

An IVP can be expressed as follows:

Definition 1 (An IVP with ODE [7]). *Starting from* $y(0) = y_0$, we want to compute y(h) w.r.t. *ODE* $\dot{y} = f(y)$; h > 0.

An exact solution is $y(h) = y_0 + \int_0^h f(y(s))ds$. As this expression makes the function y(t) (what we are looking for) appear, this solution can not be computed explicitly. This is why an approximation is often computed using numerical integration.

2.3. B-Series

Rooted trees allow one to build the formal series given in Equation (2), solution of the ODE $\dot{y} = f(y)$. The elementary differentials $F(\tau)(y)$ are expressed regarding trees such that

$$F(\bullet)(y) = f(y), \quad F(\tau)(y) = f^k(y)(F(\tau_1)(y), \dots, F(\tau_k)(y)),$$

with $\tau, \tau_1, \ldots, \tau_k \in \mathcal{T}, \mathcal{T} = \{\bullet, \bullet, \bullet, \bullet, \ldots\}$ the set of trees and $\tau = [\tau_1, \ldots, \tau_k]$ the result of *k* trees grafting.

A B-series [36] is a formal series of the form

$$B(a,hf,y) = a(\emptyset)y + \sum_{\tau \in \mathcal{T}} \frac{a(\tau)h^{|\tau|}}{\sigma(\tau)} F(\tau)(y) ,$$

with F(.)(y) the elementary differentials as described above.

Lemma 1 (B-series and IVP [36]). The exact solution of an IVP is given by a B-series

$$y(h) = B(E, hf, y_0), with E(\tau) = 1/\tau!$$
.

Lemma 2 (Taylor expansion). *The terms in a B-series correspond to the terms in the Taylor expansion of a map* [37].

Operations on B-series:

Two important operations can be done on B-series, the composition and the substitution.

- Composition law: For $b(\emptyset) = 1$, the composition $B(a, hf, B(b, hf, y)) = B(b \cdot a, hf, y)$ is also a B-series, where $(b \cdot a)(\tau) = \sum_{s \in S(\tau)} b(\tau \setminus s)a(s_{\tau})$ and $(b \cdot a)(\emptyset) = a(\emptyset)$.
- Substitution law: For $b(\emptyset) = 0$, the substitution $B(a, B(b, hf, \cdot), y) = B(b \star a, hf, y)$ is also a B-series, where $(b \star a)(\tau) = \sum_{p \in P(\tau)} b(\tau \setminus p)a(p_{\tau})$ and $(b \star a)(\emptyset) = a(\emptyset)$.

Definition of ordered subtrees S and partition P.

For the composition law, the notion of ordered subtrees (or admissible cuts) is required. An ordered subtree of a tree τ is a subset s of the set of all $|\tau|$ vertices connected and containing the root of τ . All the subtrees of τ are gathered in the set $S(\tau)$. In the previous paragraph, $\tau \setminus s$ is then a collection of rooted trees (a forest) obtained by removing the subtree s to the tree τ , and s_{τ} is the rooted tree obtained after the cuts.

For the substitution law, the notion of partitions is needed. A partition p of a tree τ is a subset of the edges of the tree. All the partitions are gathered in the set $P(\tau)$. In the previous paragraph, $\tau \setminus p$ is the forest remaining when edges of p are removed in τ , and p_{τ} is the tree built from all trees from $\tau \setminus p$ (also called the skeleton).

2.4. Runge-Kutta Methods

With the previous notations and definitions, a Runge–Kutta scheme can be defined as a B-series.

Lemma 3 (Runge–Kutta and B-series [36]). *A Runge–Kutta scheme approximates the solution of ODE* (1) *with a B-series:*

$$y(h) \approx y_1 = B(\phi(A, b), hf, y_0),$$

if the Runge–Kutta scheme is defined as follows

$$y_1 = y_0 + b^T h f(Y), \quad Y = ey_0 + A h f(Y),$$

with coefficients (A, b) given in a Butcher tableau [2] of the form

In terms of the form of the matrix A, consisting of the coefficients a_{ij} , a Runge–Kutta method can be:

- *Explicit*—for example, as in the classical Runge–Kutta method of order 4 given in Figure 1a. It means that the computation of the intermediate steps k_i only depends on the previous steps k_i for j < i.
- *Diagonally implicit*—for example, as in the diagonally implicit fourth-order method given in Figure 1b. In this case, the computation of an intermediate step *k_i* involves the value *k_i* itself; therefore, non-linear systems in *k_i* must be solved (with a Newton algorithm, for example). A method is *singly diagonally implicit* if the coefficients on the diagonal are all equal.
- *Fully implicit*—for example, the Runge–Kutta fourth-order method with a Lobatto quadrature formula given in Figure 1c. In this last case, the computation of intermediate steps involves the solution of a non-linear system of equations in all the values k_i for i = 1, 2, · · · , s.



Figure 1. Different kinds of Runge-Kutta methods [7].

The order of a Runge–Kutta method is *p* if and only if the *local truncation error*, in other words, the distance between the exact solution $y(t_{\ell}; y_{\ell-1})$ and the numerical solution y_{ℓ} is such that:

$$y(t_{\ell}; y_{\ell-1}) - y_{\ell} \in \mathcal{O}(h^{p+1})$$

In term of B-series, it means that a Runge–Kutta method is of order *p* if

$$E(\tau) = \phi(A, b)(\tau), \quad \forall \tau \in \mathcal{T}, |\tau| \le p,$$

which means that $B(E, hf, y_0) - B(\phi(A, b), hf, y_0) \in \mathcal{O}(h^{p+1})$.

The main idea behind Runge-Kutta methods.

As a B-series (or a Taylor expansion) is an infinite and noncomputable, sum of terms that gives the solution of an ODE. By cutting it at a given order, one can obtain a sufficient approximation. Runge–Kutta methods provide these approximations at a given order if the previous condition on coefficients holds without derivative computation.

3. B-Series and Sets

In this section, a formalism of set-based B-series is presented. First, different kinds of ODEs with initial conditions and parameters defined as sets are discussed. Indeed, considering uncertainties bounded in sets, an IVP with the ODE (1) can be:

- 1. a parametrized ODE $\dot{y} = f(y, p)$, with $y(0) = y_0$ and $p \in \mathcal{P} \subset \mathbb{R}^{|p|}$;
- 2. a non precise initial value problem $\dot{y} = f(y)$, with $y(0) \in \mathcal{Y}_0 \subset \mathbb{R}^{|y|}$; or
- 3. a non precise initial value problem with a parametrized ODE $\dot{y} = f(y, p)$, with $y(0) \in \mathcal{Y}_0$ and $p \in \mathcal{P}$.

Case 1 and 3 have to be discussed. Case 1 can be seen as the differential inclusion $\dot{y} \in f(y, \mathcal{P})$; however, the nature of the parameter is important as it could lead to errors in some contributions for reachability computations.

In terms of the literature, one paper studied the computation of rigorous reachable set in the particular case of differential inclusion [38]. The authors proposed some conditions on parameters and a novel algorithm based on [13].

3.1. Discussion on the Nature of Parameters

Two kind of parameters could appear in a model of physical systems: (i) a physical parameter, unknown but constant or (ii) a varying parameter, bounded but with an unknown derivative.

Physical Parameters

Considering a physical parameter in a model, such as a length, mass, etc, marred by uncertainties during the modeling process, the actual value is, of course, not perfectly known but is unique, constant (even if the mass can vary with respect to the fuel consumption in some applications, in these cases, the mass without fuel consideration is the parameter that interests us in this discussion) and bounded. This leads to the fact that the parameter derivative is equal to zero, and then the ODE can be extended as follows:

$$\begin{cases} \dot{y} = f(y, p) \\ \dot{p} = 0 \end{cases},$$

with $p(0) \in \mathcal{P}_0$ (and then $p(t) \in \mathcal{P}_0, \forall t \ge 0$). The problem is then similar to a non precise initial value problem with parameter added to states. If we split the parameter set \mathcal{P}_0 such that $\mathcal{P}_0 = \mathcal{P}_0^1 \cup \mathcal{P}_0^2$, then the reachable set corresponding to \mathcal{P}_0 is the set-union of the reachable sets associated with \mathcal{P}_0^1 and \mathcal{P}_0^2 due to the monotonicity of sets. It leads to the fact that algorithms based on bisection will be correct. Derivatives of p will not appear in the high order terms in the B-series; thus, the order of the B-series is respected.

3.2. Not Fixed Parameters

The other kind of parameter is time varying—for example, an input of a controlled system, a noise or some perturbations. These parameters (or variables) are unknown but bounded. Their derivatives are also unknown and different than zero. This leads to the following extended ODE:

$$\begin{cases} \dot{y} = f(y, p) \\ \dot{p} \in [-\infty, \infty] \end{cases}$$

with $p(t) \in \mathcal{P}_0, \forall t \ge 0$. This ODE cannot be integrated because it does not fulfill the requirements (non smooth and non Lipschitz). Moreover, if we split the parameter set \mathcal{P}_0 such that $\mathcal{P}_0 = \mathcal{P}_0^1 \cup \mathcal{P}_0^2$, then the reachable set corresponding to \mathcal{P}_0 is not the setunion of the reachable sets associated with \mathcal{P}_0^1 and \mathcal{P}_0^2 , because, at each instant, all the values for p are possible, i.e., taken in \mathcal{P}_0^1 or in \mathcal{P}_0^2 . Therefore, bisection-based approaches must be avoided in this case (some papers have proposed bisection-based algorithms—for example, for control synthesis, which leads to mistakes in reachable sets and then in safety verification).

Derivatives of *p* will appear in the high order terms in the B-series; thus, the order of the B-series is not respected. In this objective, parameters have to be considered as constant during an integration step (that can be small regarding a control frequency, for example).

To conclude this discussion, cases 1, 2 and 3 can be seen as differential inclusions, under the condition that parameters are considered as piecewise constant. This strong condition is finally due to a lack of knowledge in the model that we attempt to simulate. In order to define a more precise model, one can add parameters in the states and define these derivatives as parameters (at least one of its derivatives has to be piecewise constant).

3.3. Set-Based B-Series

We propose to define set-based B-series to express the solution of differential inclusions (with the assumption of piecewise constant parameters as discussed above), such as

$$\dot{y} \in \mathcal{F}(y), y(0) \in \mathcal{Y}_0$$
 (6)

From the B-series definition, the following can be proposed:

Proposition 1 (Set-based B-series for differential inclusions). *A set-based B-series* $B(E, h\mathcal{F}, \mathcal{Y}_0)$ (with $E(\tau) = 1/\tau!$) is the exact reachable set of the differential inclusion (6) at instant h.

Set-based B-series being defined as set of punctual B-series, we state what follows:

Proposition 2 (B-series define an element of set-based B-series). With the B-series $B(E, hf, y_0)$, with $f \in \mathcal{F}$ and $y_0 \in \mathcal{Y}_0$, the following inclusion holds due to the monotonic inclusion of sets: $B(E, hf, y_0) \in B(E, h\mathcal{F}, \mathcal{Y}_0)$.

Considering the pointwise union (or collecting), the following holds

$$\bigcup_{f\in\mathcal{F},y_0\in\mathcal{Y}_0}B(E,hf,y_0)\equiv B(E,h\mathcal{F},\mathcal{Y}_0).$$

This latter is equivalent to the principle behind the Monte-Carlo approach with uniform distributions for parameters and initial condition and an infinite number of simulations. The main advantage of set-based approaches is that one reachability computation is equivalent to an infinite number of pointwise computations.

Proposition 3 (Composition law is applicable to set-based B-series).

Proof. A set-based B-series describes the reachable set at instant *h* such that $\mathcal{Y}(h) = B(E, h\mathcal{F}, \mathcal{Y}_0)$, and then choosing $\mathcal{Y}_0 = \mathcal{Y}(h)$ leads to the composition law. \Box

Proposition 4 (Substitution law is applicable to set-based B-series).

Proof. A set-based B-series describes a set inclusion $\mathcal{Y}(h) = \{y(t) \in B(E, h\mathcal{F}, \mathcal{Y}_0), t = h\}$ that can be seen as a differential inclusion, which leads to the substitution law. \Box

4. Validated B-Series

In this section, a generalization of validated integration (reachability analysis or guaranteed simulation) is proposed using set-based B-series. In these methods, computation must be guaranteed by considering numerical errors from the integration scheme (i.e., the truncation of the B-series) and from the floating point arithmetic (this point is discussed in Section 4.5).

4.1. Validated Truncated B-Series

B-series are truncated in order to be computable (with Runge–Kutta methods or Taylor approximations). First of all, by considering the truncation error, one can validate a finite B-series at order p with the following formula:

$$y(h) = B(E, hf, y_0) = B(E_{0,\dots,p}, hf, y_0) + B(E_{p+1}, hf, \xi),$$
(7)

with ξ the unknown Lagrangian remainder point, $\xi = y(s), s \in [0, h]$ and the functions

$$E_{0,...,p} = 1/\tau! \text{ if } |\tau| \le p,$$
 (8)

$$E_{0,\dots,p} = 0 \text{ elsewhere.} \tag{9}$$

and

$$E_{p+1} = 1/\tau!$$
 if $|\tau| = p+1$, (10)

$$E_{p+1} = 0 \text{ elsewhere.} \tag{11}$$

This latter formula comes directly from the Lagrangian remainder of Taylor expansion [39]. The term $B(E_{p+1}, hf, \xi)$ is called the local truncation error (LTE). The term $E_{0,...,p}$ can be computed formally or with automatic differentiation for Taylor methods, or with Runge–Kutta methods (see [40] for details on Runge–Kutta coefficients for validated integration) while the term E_{p+1} can be computed formally, with automatic differentiation or rooted trees (more details in [23,41]).

Remark 1. The addition between a B-series and a set-based B-series is a Minkowski sum. This depends on the representation chosen for the sets (see Section 4.5), and this addition is never computed in practice as it is preferable to directly add two set-based B-series.

4.2. With ODEs

The main issue in the above formula is to find the Lagrangian term to compute the LTE. By the help of set-based B-series, an enclosure is computable. Indeed, the exact solution of an ODE, such as (1), can be enclosed in a validated B-series with the help of the following formula:

$$y(h) = B(E, hf, y_0) \in B(E_{0,\dots,p}, hf, y_0) + B(E_{p+1}, hf, \mathcal{Y}^*) .$$
(12)

The second term on the right part is a set-based B-series, evaluated on the set \mathcal{Y}^* , which is proven by construction (see Section 4.4) to contain the Lagrangian remainder point, i.e., $\xi \in \mathcal{Y}^*$.

Runge–Kutta methods:

Let us consider a Runge–Kutta scheme at order p and defined by the Butcher tableau (A, b), then a validated B-series can be constructed:

$$y(h) = B(E, hf, y_0) \in B(\phi(A, b), hf, y_0) + B(\psi(A, b), hf, \mathcal{Y}^*)$$

with $\mathcal{Y}^* \ni y(x), \forall x \in [0, h]$, and

$$\psi(A,b)(au) = 1/ au! - \phi(A,b)(au), \forall au \in \mathcal{T} ext{ if } | au| = p+1$$
 ,

or

$$\psi(A, b)(\tau) = 0$$
 if $|\tau| \neq p + 1$.

The expression $1/\tau! - \phi(A, b)(\tau)$ comes from the LTE definition, obtained with the difference of the two Lagrangian remainders at order p + 1 of two B-series (the one from the exact solution and the one from the numerical approximation), see [2] for more details. An important remark is that, if $|1/\tau! - \phi(A, b)(\tau)| < |1/\tau!|$, then the LTE of the Runge–Kutta scheme will be smaller than the Taylor's one, i.e., the approximation is then better with the Runge–Kutta scheme than with the order equivalent Taylor series.

Remark 2. With a zero order method, i.e., p = 0, we obtain the Euler method.

4.3. With Differential Inclusions

One can compute the reachable set of a differential inclusion, such as (6), with a validated truncated set-based B-series with the following formula:

$$\mathcal{Y}(h) = B(E, h\mathcal{F}, \mathcal{Y}_0) \subset B(E_{0,\dots,p}, h\mathcal{F}, \mathcal{Y}_0) + B(E_{p+1}, h\mathcal{F}, \mathcal{Y}^*).$$

Runge-Kutta methods can be used to compute these B-series, as described above.

4.4. Picard–Lindelöf

To validate a B-series, a Lagrangian remainder has to be computed in an unknown point, see Equation (7). In Equation (12), the proposed solution consists of enclosing this unknown point in a set \mathcal{Y}^* . In order to obtain this latter, the well-known Picard–Lindelöf theorem can be used. Furthermore, called the Cauchy–Lipschitz theorem, or existence and uniqueness theorem, it gives a set of conditions, under which, an initial value problem accepts a unique solution. Details on this theorem and proof can be found in several books, such as in [42].

A Picard–Lindelöf operator has been defined, already based on sets and B-series:

Definition 2 (Picard–Lindelöf operator with B-series [3]). *If there exists a set* $\mathcal{R} \subset \mathbb{R}^n$ *such that*

$$B(E,[0,h]f,\mathcal{R})\subset\mathcal{R}$$

then, due to Brouwer theorem (also proposed/studied by Picard, Poincaré and Hadamard) and Picard– Lindelöf theorem, the initial value problem has a unique solution in the set \mathcal{R} . As a consequence, this proves that $y(x) \in \mathcal{R}, \forall x \in [0, h]$ (existence and uniqueness simultaneously).

Using this operator in a simple inflate and contract algorithm, one can obtain the set \mathcal{R} , such that $\mathcal{R} \supset \mathcal{Y}^*$, and can then replace \mathcal{Y}^* by \mathcal{R} to compute the LTE in Equation (12). The infinite B-series $B(E, [0, h]f, \mathcal{R})$ can be replaced by a validated truncated B-series such that

$$B(E_{0,\dots,\nu},[0,h]f,y_0) + B(E_{\nu+1},[0,h]f,\mathcal{R}) \subset \mathcal{R}.$$
(13)

Picard–Lindelöf operator for Runge–Kutta methods:

Let us consider a Runge–Kutta scheme at order p and defined by the Butcher tableau (A, b), then a validated B-series can be constructed:

$$y(h) = B(E, hf, y_0) \in B(\phi(A, b), hf, y_0) + B(\psi(A, b), hf, \mathcal{R}).$$

In addition, the latter can be itself used as a Picard–Lindelöf operator:

Definition 3 (Picard–Lindelöf operator with validated Runge–Kutta). *If there exists a set* $\mathcal{R} \subset \mathbb{R}^n$ *such that*

$$B(\phi(A,b),[0,h]f,y_0)+B(\psi(A,b),[0,h]f,\mathcal{R})\subset\mathcal{R}$$
,

then, the initial value problem has a unique solution in the set \mathcal{R} .

With differential inclusions:

Due to the monotonic inclusion, the previous results are directly applied to set-based B-series. Therefore, one can compute the reachable set of a differential inclusion, such as (6) with a validated truncated set-based B-series with the following formula:

$$\mathcal{Y}(h) = B(E, h\mathcal{F}, \mathcal{Y}_0) \subset B(E_{0,\dots,p}, h\mathcal{F}, \mathcal{Y}_0) + B(E_{p+1}, h\mathcal{F}, \mathcal{R})$$

Furthermore, the same result is obtained with validated truncated set-based B-series using Runge–Kutta (by replacing E_{\cdot} by $\phi(.)$ and $\psi(.)$).

Remark 3. With a zero order method, i.e., p = 0, we obtain the validated Euler method, also called the rectangle rule, as proposed by Moore [12].

4.5. Remarks on Set Representation

The abstraction used to represent sets in a computer must have several properties in order to implement the presented techniques. First, as said in the section on validated B-series, the chosen abstraction must handle numerical errors due to floating numbers. Second, it has to come with an arithmetic, such as a Minkowski sum, as well as all the guaranteed arithmetic operators, even the nonlinear ones (for non linear ODES, e.g., with the cosine or square root). Finally, set operators, at least the inclusion test, are needed for the Picard–Lindelöf operator.

Some abstractions are well-known and mainly used for reachability, such as intervals [12], zonotopes [43] or Taylor models [44]. Other approaches, less famous but efficient in certain cases/applications, have been tested, such as ellipsoids [45], parallelotopes [46], polytopes [47], etc.

For example, the tool DynIbex (Website: https://perso.ensta-paris.fr/~chapoutot/ dynibex/) proposes validated Runge–Kutta methods using numerically guaranteed intervals and zonotopes for set abstraction.

A trick to compute with polytopes:

As said above, polytopes can be used in set-based B-series, even if no associated arithmetic is available. Indeed, a polytope can be described by the intersection of a finite number of zonotopes [47]. Let us define an initial condition \mathcal{Y}_0 enclosed by a polytope \mathcal{P}_0 . If $\mathcal{P}_0 = \mathcal{Z}^0 \cap \mathcal{Z}^1 \cap \cdots \cap \mathcal{Z}^z$, then the following holds:

$$\mathcal{Y}(h) = B(E, h\mathcal{F}, \mathcal{Y}_0) \subset B(E, h\mathcal{F}, \mathcal{P}_0) = B(E, h\mathcal{F}, \mathcal{Z}^0) \cap B(E, h\mathcal{F}, \mathcal{Z}^1) \cap \cdots \cap B(E, h\mathcal{F}, \mathcal{Z}^z).$$

4.6. A Basic Algorithm to Compute Reachability

An algorithm can be built gathering previous definitions, able to compute reachability of differential inclusions. An overview of this algorithm, based on the well-known Lohner two-step method [13], is given in the following. Many steps are not described; for more details on the algorithm and its implementation, refer to [23].

- 0. Starting with an initial condition \mathcal{Y}_0 , a differential inclusion \mathcal{F} , a stepsize h, a guess \mathcal{R} .
- 1. Initialize reachable set $\mathcal{Y} = \mathcal{Y}_0$.
- 2. While $B(E_{0,\dots,p},[0,h]\mathcal{F},\mathcal{Y}) + B(E_{p+1},[0,h]\mathcal{F},\mathcal{R}) \not\subset \mathcal{R}$, inflate \mathcal{R} .
- 3. Let $\mathcal{Y}^* = \mathcal{R}$.
- 4. Compute reachable set $\mathcal{Y} = B(E_{0,\dots,p}, h\mathcal{F}, \mathcal{Y}) + B(E_{p+1}, h\mathcal{F}, \mathcal{Y}^*).$
- 5. Go back to step 2.

This algorithm can be strongly improved [23]—for example, with some tricks for the guess \mathcal{R} , for the stepsize controller, with additive contraction on \mathcal{R} , the techniques used to compute the B-series terms, set operations, etc.

5. Experimentation

As experiments, in order to show the kind of results that can be computed with the proposed approach, we propose to study the well-known Lotka–Volterra problem. This is defined by the following equation:

$$\begin{cases} \dot{y}_1 = 2y_1(1-y_2) \\ \dot{y}_2 = -y_2(1-y_1) \end{cases}$$
(14)

The initial condition is taken in the following zonotope:

$$Z_0 = \begin{bmatrix} 1.1007 \\ 3.0422 \end{bmatrix} \oplus \begin{bmatrix} -0.0032 & -0.0008 \\ 0.0016 & 0.0099 \end{bmatrix} \mathcal{B}^2$$

with \mathcal{B} the ball in 2D. Simulation with this zonotope as initial conditions is performed with Kutta third order (which is a good compromise between time and precision) validated

method with threshold of 10^{-6} on the LTE (which constrains the stepsize *h*) and until six time units. Reachable sets are plotted in Figure 2.

One may want to validate the proposed techniques by the help of classical integration schemes and a Monte-Carlo approach. However, this does not prove anything as classical integration schemes compute an approximation of a real function that may or may not be included in a validated reachable tube. Nevertheless, one can verify the major property of set-based B-series, which is the following:

$$\mathcal{Y}_0 \supset \mathcal{Y}_0^0 \cup \mathcal{Y}_0^1 \implies B(E, h\mathcal{F}, \mathcal{Y}_0) \supset B(E, h\mathcal{F}, \mathcal{Y}_0^0) + B(E, h\mathcal{F}, \mathcal{Y}_0^1)$$

Combined with a Monte-Carlo approach (10 discretizations per dimension, i.e., 100 initial conditions), several validated simulations are performed, and the inclusion in the previously computed tube is verified (see Figure 3).



Figure 2. On top: reachable tube obtained with zonotopes—for example, (14) (y_1 w.r.t. y_2). Bottom left and right: zoom on a small part computed with intervals or zonotopes (respectively).



Figure 3. Reachable tube obtained for example (14) (y_1 w.r.t. y_2) and Z_0 as the initial condition in blue. One hundred reachable tubes with the Monte-Carlo approach in green.

6. Discussion and Conclusions

To conclude this paper, we proposed a formalism based on B-series and sets to compute the reachability of initial value problems with ordinary differential equations or differential inclusions (under certain assumptions on the parameters). This unified formalism of setbased B-series gathers the known methods for reachability: techniques (Taylor series or Runge–Kutta) and set abstractions (intervals, zonotopes, Taylor models, etc.). A simple algorithm can be deduced from the associated definitions.

As future work, a strong theoretical study is needed to transfer all the previous results on B-series (50 years of studies) to set-based B-series. In particular, some theorems could lead to unexpected results, such as conditions on Runge–Kutta coefficients that can help to create schemes with lower LTE complexity by reducing the order of rooted trees.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

- 1. Euler, L. *Institutiones Calculi Integralis;* Academia Imperialis Scientiarum, Petropolitanae Impressa, 1792. Available online: https://scholarlycommons.pacific.edu/euler-works/342/ (accessed on 18 July 2022).
- 2. Butcher, J.C. Coefficients for the Study of Runge–Kutta Integration Processes. J. Aust. Math. Soc. 1963, 3, 185–201. [CrossRef]
- 3. Butcher, J.C. An algebraic theory of integration methods. *Math. Comput.* **1972**, *26*, 79–106. [CrossRef]
- 4. Cayley, A. XXVIII. On the theory of the analytical forms called trees. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1857**, *13*, 172–176. [CrossRef]
- Merson, R. An operational method for the study of integration processes. In Proceedings of the Conference on Data Processing and Automatic Computing Machines, Salisbury, South Australia, 9–13 December 1957; Weapons Research Establishment: Salisbury, South Australia, 1957; Volume 1, pp. 110–125.
- 6. Chartier, P.; Hairer, E.; Vilmart, G. Algebraic Structures of B-series. Found. Comput. Math. 2010, 10, 407–420. [CrossRef]
- 7. Hairer, E.; Wanner, G. On the Butcher group and general multi-value methods. *Computing* 1974, 13, 1–15. [CrossRef]

- Markov, A.A. Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga. *Izv.-Fiz.-Mat. Obs. Pri Kazan. Univ.* 1906, 15, 135–156.
- 9. Metropolis, N.; Ulam, S. The Monte Carlo Method. J. Am. Stat. Assoc. 1949, 44, 335–341. [CrossRef]
- 10. Erdös, P. Graph Theory and Probability. Can. J. Math. 1959, 11, 34–38. [CrossRef]
- 11. Jaulin, L.; Kieffer, M.; Didrit, O.; Walter, E. Applied Interval Analysis; Springer: Berlin/Heidelberg, Germany, 2001.
- 12. Moore, R.E. Interval Analysis; Series in Automatic Computation; Prentice Hall: Hoboken, NJ, USA, 1966.
- 13. Lohner, R.J. Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In *Proceedings of the Institute of Mathematics and Its Applications Conference Series;* Oxford University Press: Oxford, UK, 1992; Volume 39, p. 425.
- 14. Nedialkov, N.S.; Jackson, K.R.; Corliss, G.F. Validated solutions of initial value problems for ordinary differential equations. *Appl. Math. Comput.* **1999**, *105*, 21–68. [CrossRef]
- 15. Makino, K.; Berz, M. COSY INFINITY Version 9. Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip. 2006, 558, 346–350. [CrossRef]
- Lin, Y.; Stadtherr, M.A. Validated solutions of initial value problems for parametric ODEs. *Appl. Numer. Math.* 2007, 57, 1145–1162. [CrossRef]
- 17. Dzetkulič, T. Rigorous integration of non-linear ordinary differential equations in Chebyshev basis. *Numer. Algorithms* **2015**, 69, 183–205. [CrossRef]
- Gajda, K.; Marciniak, A.; Szyszka, B. Three- and Four-Stage Implicit Interval Methods of Runge-Kutta Type. Comput. Methods Sci. Technol. 2000, 6, 41–59. [CrossRef]
- 19. Marciniak, A.; Szyszka, B. On Representations of Coefficients in Implicit Interval Methods of Runge–Kutta Type. *Comput. Methods Sci. Technol.* **2004**, *10*, 57–71. [CrossRef]
- 20. Marciniak, A. Implicit Interval Methods for Solving the Initial Value Problem. Numer. Algorithms 2004, 37, 241–251. [CrossRef]
- 21. Bouissou, O.; Martel, M. GRKLib: A Guaranteed Runge Kutta Library. In Proceedings of the Scientific Computing, Computer Arithmetic and Validated Numerics, Duisburg, Germany, 26–29 September 2006.
- Bouissou, O.; Chapoutot, A.; Djoudi, A. Enclosing Temporal Evolution of Dynamical Systems Using Numerical Methods. In Proceedings of the NASA Formal Methods, Moffett Field, CA, USA, 14–16 May 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 108–123.
- 23. Alexandre dit Sandretto, J.; Chapoutot, A. Validated Explicit and Implicit Runge-Kutta Methods. Reliab. Comput. 2016, 22, 79–103.
- 24. Alexandre dit Sandretto, J.; Chapoutot, A. Validated simulation of differential algebraic equations with Runge–Kutta methods. *Reliab. Comput.* **2016**, *22*, hal-01243044.
- Geretti, L.; Sandretto, J.A.D.; Althoff, M.; Benet, L.; Chapoutot, A.; Chen, X.; Collins, P.; Forets, M.; Freire, D.; Immler, F.; et al. ARCH-COMP20 Category Report: Continuous and Hybrid Systems with Nonlinear Dynamics. In Proceedings of the ARCH20—7th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH20), Berlin, Germany, 12 July 2020; Frehse, G., Althoff, M., Eds.; Volume 74, pp. 49–75. [CrossRef]
- Geretti, L.; Sandretto, J.A.D.; Althoff, M.; Benet, L.; Chapoutot, A.; Collins, P.; Duggirala, P.S.; Forets, M.; Kim, E.; Linares, U.; et al. ARCH-COMP21 Category Report: Continuous and Hybrid Systems with Nonlinear Dynamics. In Proceedings of the eighth International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH21), Brussels, Belgium, 9 July 2021; Frehse, G., Althoff, M., Eds.; Volume 80, pp. 32–54. [CrossRef]
- Bresolin, D.; Collins, P.; Geretti, L.; Segala, R.; Villa, T.; Gonzalez, S.Ž. A Computable and Compositional Semantics for Hybrid Automata. In Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control (HSCC '20), Virtually, 22–24 April 2020; Association for Computing Machinery: New York, NY, USA, 2020. [CrossRef]
- Althoff, M. An Introduction to CORA 2015. In Proceedings of the Workshop on Applied Verification for Continuous and Hybrid Systems, Seattle, WA, USA, 13 April 2015; pp. 120–151.
- 29. Althoff, M.; Grebenyuk, D. Implementation of Interval Arithmetic in CORA 2016. In Proceedings of the third International Workshop on Applied Verification for Continuous and Hybrid Systems, Vienna, Austria, 11 April 2016; pp. 91–105.
- Bogomolov, S.; Forets, M.; Frehse, G.; Potomkin, K.; Schilling, C. JuliaReach: A Toolbox for Set-Based Reachability. In Proceedings of the HSCC, Montreal, QC, Canada, 16–18 April 2019. [CrossRef]
- Kim, E.; Duggirala, P.S. Kaa: A Python Implementation of Reachable Set Computation Using Bernstein Polynomials. *EPiC Ser. Comput.* 2020, 74, 184–196.
- 32. Platzer, A. A Complete Uniform Substitution Calculus for Differential Dynamic Logic. J. Autom. Reason. 2017, 59, 219–265. [CrossRef]
- 33. Munthe-Kaas, H. Lie-Butcher Theory for Runge-Kutta Methods. BIT Numer. Math. 1995, 35, 572-587. [CrossRef]
- Hairer, E.; Lubich, C.; Wanner, G. Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations; Computational Mathematics; Springer: Berlin/Heidelberg, Germany, 2006.
- 35. Butcher, J.C. Numerical Methods for Ordinary Differential Equations; Wiley: Hoboken, NJ, USA, 2003.
- Butcher, J.; Sanz-Serna, J. The number of conditions for a Runge–Kutta method to have effective order p. *Appl. Numer. Math.* 1996, 22, 103–111. [CrossRef]
- Mclachlan, R.I.; Modin, K.; Munthe-Kaas, H.; Verdier, O. B-Series Methods Are Exactly the Affine Equivariant Methods. *Numer. Math.* 2016, 133, 599–622. [CrossRef]

- 38. Kapela, T.; Zgliczyński, P. A Lohner-type algorithm for control systems and ordinary differential inclusions. *Discret. Contin. Dyn. Syst.-B* **2009**, *11*, 365–385. [CrossRef]
- 39. Hairer, E.; Nørsett, S.P.; Wanner, G. Solving Ordinary Differential Equations I: Nonstiff Problems, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009.
- 40. Alexandre dit Sandretto, J. Runge–Kutta theory and constraint programming. Reliab. Comput. 2017, 25, 178–201.
- 41. Bartha, F.; Munthe-Kaas, H.Z. Computing of B-series by automatic differentiation. *Discret. Contin. Dyn. Syst.* **2014**, *34*, 903–914. [CrossRef]
- 42. Coddington, A.; Levinson, N. *Theory of Ordinary Differential Equations*; International Series in Pure and Applied Mathematics; McGraw-Hill: New York, NY, USA, 1955.
- Girard, A. Reachability of uncertain linear systems using zonotopes. In Proceedings of the International Workshop on Hybrid Systems: Computation and Control, Zurich, Switzerland, 9–15 March 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 291–305.
- 44. Chen, X. Reachability Analysis of Non-Linear Hybrid Systems Using Taylor Models. Ph.D. Thesis, Fachgruppe Informatik, RWTH Aachen University, Aachen, Germany, 2015.
- Kurzhanski, A.B.; Varaiya, P. Ellipsoidal techniques for reachability analysis. In Proceedings of the International Workshop on Hybrid Systems: Computation and Control, Pittsburgh, PA, USA, 23–25 March 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 202–214.
- Dreossi, T.; Dang, T.; Piazza, C. Parallelotope bundles for polynomial reachability. In Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, Vienna, Austria, 12–14 April 2016; pp. 297–306.
- Alexandre dit Sandretto, J.; Wan, J. Reachability analysis of nonlinear odes using polytopic based validated runge-kutta. In Proceedings of the International Conference on Reachability Problems, Marseille, France, 24–26 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–14.