Trees, Bialgebras and Intrinsic Numerical Algorithms

PETER CROUCH
ROBERT GROSSMAN
RICHARD LARSON

Laboratory for Advanced Computing
Technical Report LAC90-R23

Department of Mathematics, Statistics,
and Computer Science
University of Illinois at Chicago (M/C 249)
P. O. Box 4348
Chicago, IL 60680

May, 1990

Trees, Bialgebras and Intrinsic Numerical Algorithms

Peter Crouch[*], Robert Grossman[†] and Richard Larson[‡]

May, 1990

Abstract

This report describes preliminary work about intrinsic numerical integrators evolving on groups. Fix a finite dimensional Lie group G, let g denote its Lie algebra, and let $Y_1, \ldots, Y_N$ denote a basis of g. We give a class of numerical algorithms to approximate solutions to differential equations evolving on G of the form:

$$\dot{x}(t) = F(x(t)), \qquad x(0) = p \in G,$$

where

$$F = \sum_{\mu=1}^{N} a^\mu Y_\mu, \quad a^\mu \in C^\infty(G).$$

The algorithm depends upon constants c_i and c_{ij}, for $i = 1, \ldots, k$ and $j < i$. The algorithm has the property that if the algorithm starts on the group, then it remains on the group. It also has the property that if G is the abelian group $\mathbf{R}^N$, then the algorithm becomes the classical Runge-Kutta algorithm. We use the Cayley algebra generated by labeled, ordered trees to generate the equations that the coefficients c_i and c_{ij} must satisfy in order for the algorithm to yield an rth order numerical integrator and to analyze the resulting algorithms.

[*]This research was supported by the NSF and EPRI.

[†]This research is supported in part by NASA grant NAG2-513 and NSF Grant DMS-8904740.

[‡]This research is supported by NSF Grant DMS-8904740.

1 Introduction

Fix a finite dimensional Lie group G, let g denote its Lie algebra, and let $Y_1, \ldots, Y_N$ denote a basis of g. We give a class of numerical algorithms to approximate solutions to differential equations evolving on G of the form:

$$\dot{x}(t) = F(x(t)), \qquad x(0) = p \in G,$$

where

$$F = \sum_{\mu=1}^{N} a^{\mu} Y_{\mu}, \quad a^{\mu} \in C^{\infty}(G).$$

The algorithm depends upon constants c_i and c_{ij}, for $i = 1, \ldots, k$ and $j < i$. The algorithm has the property that if the algorithm starts on the group, then it remains on the group. It also has the property that if G is the abelian group $\mathbf{R}^N$, then the algorithm becomes the classical Runge-Kutta algorithm. Our analysis requires the Cayley algebra generated by labeled, ordered trees, introduced in [10], [11] and [6]. We use the Cayley algebra of trees to generate the equations that the coefficients c_i and c_{ij} must satisfy in order for the algorithm to yield an rth order numerical integrator and to analyze the resulting algorithms.

This is a preliminary report. A final report containing complete proofs, examples, and a further analysis of the algorithms is in preparation.

2 Families of trees

The relation between trees and Taylor's theorem goes back as least as far as Cayley [3] and [4]. Important use of this relation has been made by Butcher in his work on high order Runge-Kutta algorithms [1] and [2]. In this section and the next, we follow the treatment in [10] and [11].

By a tree we mean a rooted finite tree. If $\{F_1, \ldots, F_M\}$ is a set of symbols, we will say a tree is *labeled with* $\{F_1, \ldots, F_M\}$ if every node of the tree other than the root has an element of $\{F_1, \ldots, F_M\}$ assigned to it. We denote the set of all trees labeled with $\{F_1, \ldots, F_M\}$ by $\mathcal{LT}(F_1, \ldots, F_M)$. Let $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ denote the vector space over k with basis $\mathcal{LT}(F_1, \ldots, F_M)$. We show that this vector space is a graded connected algebra.

We define the multiplication in $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ as follows. Since the set of labeled trees form a basis for $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$, it is sufficient to describe the product of two labeled trees. Suppose t_1 and t_2 are two labeled trees. Let $s_1, \ldots, s_r$ be the children of the root of t_1. If t_2 has $n+1$

nodes (counting the root), there are $(n+1)^r$ ways to attach the r subtrees of t_1 which have $s_1, \ldots, s_r$ as roots to the labeled tree t_2 by making each s_i the child of some node of t_2, keeping the original labels. The product $t_1 t_2$ is defined to be the sum of these $(n+1)^r$ labeled trees. It can be shown that this product is associative, and that the tree consisting only of the root is a multiplicative identity; see [5].

We can define a grading on $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ by letting $k\{\mathcal{LT}_n(F_1, \ldots, F_M)\}$ be the subspace of $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ spanned by the trees with $n+1$ nodes. The following theorem is proved in [9].

Theorem 2.1 *$k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ is a graded connected algebra.*

If $\{F_1, \ldots, F_M\}$ is a set of symbols, then the free associative algebra $k\langle F_1, \ldots, F_M \rangle$ is a graded connected algebra, and there is an algebra homomorphism

$$\phi : k\langle F_1, \ldots, F_M \rangle \rightarrow k\{\mathcal{LT}(F_1, \ldots, F_M)\}.$$

The map ϕ sends F_i to the labeled tree with two nodes: the root, and a child of the root labeled with F_i; it is then extended to all of $k\langle F_1, \ldots, F_M \rangle$ by using the fact that it is an algebra homomorphism.

We say that a rooted finite tree is *ordered* in case there is a partial ordering on the nodes such that the children of each node are non-decreasing with respect to the ordering. We say such a tree is labeled with $\{F_1, \ldots, F_M\}$ in case every element, except the root, has an element of $\{F_1, \ldots, F_M\}$ assigned to it. Let $k\{\mathcal{LOT}(F_1, \ldots, F_M)\}$ denote the vector space over k whose basis consists of labeled ordered trees. It turns out that $k\{\mathcal{LOT}(F_1, \ldots, F_M)\}$ is also a graded connected algebra using the same multiplication defined above. See [9] for a proof of the following theorem.

We say that a rooted finite tree is heap-ordered in case there is a total ordering on all nodes in the tree such that each node procedes all of its children in the ordering. We define $k\{\mathcal{LHOT}(F_1, \ldots, F_M)\}$ as above to be the vector space over k whose basis consists of heap-ordered trees labeled with $\{F_1, \ldots, F_M\}$. In [9] we show that $k\{\mathcal{LHOT}(F_1, \ldots, F_M)\}$ is also a graded connected algebra [9] and satisfies:

Theorem 2.2 *The map*

$$\phi : k\langle F_1, \ldots, F_M \rangle \rightarrow k\{\mathcal{LHOT}(F_1, \ldots, F_M)\}$$

is injective.

Fix N derivations $Y_1, \ldots, Y_N$ of R and consider M other derivations of R of the form

$$F_i = \sum_{\mu=1}^{N} a_i^\mu Y_\mu, \quad a_i^\mu \in R, \quad i = 1, \ldots, M. \tag{1}$$

Let End(R) denote the endormorphisms of the ring R. Using the data (1), we now define a map

$$\psi : k\{\mathcal{LT}(F_1, \ldots, F_M)\} \to \mathrm{End}(R)$$

in the following steps.

Step 1. Given a labeled tree $t \in \mathcal{LT}_m(F_1, \ldots, F_M)$, assign the root the number 0 and assign the remaining nodes the numbers $1, \ldots, m$. From now on we identify the node with the number assigned to it. Let $j \in$ nodes t , and suppose that $l, \ldots, l'$ are the children of j and that j is labeled with F_{γ_j}. Fix $\mu_l, \ldots, \mu_{l'}$ with

$$1 \leq \mu_l, \ldots, \mu_{l'} \leq N$$

and define

$$\begin{aligned} R(j; \mu_l, \ldots, \mu_{l'}) &= Y_{\mu_l} \cdots Y_{\mu_{l'}} a_{\gamma_j}^{\mu_j} \\ &\quad \text{if } j \text{ is not the root} \\ &= Y_{\mu_l} \cdots Y_{\mu_{l'}} \\ &\quad \text{if } j \text{ is the root .} \end{aligned}$$

We abbreviate this to $R(j)$. Observe that $R(j) \in R$ for $j > 0$.

Step 2. Define

$$\psi(t) = \sum_{\mu_1, \ldots, \mu_m = 1}^{N} R(m) \cdots R(1) R(0).$$

Step 3. Extend ψ to all $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ by k-linearity.

Remark 2.1 In exactly the same way, we define a map

$$\psi : k\{\mathcal{LT}(F_1, \ldots, F_M)\} \to \mathrm{End}(R),$$

by ignoring the ordering of the nodes.

Remark 2.2 Let H denote one of the algebras of labeled trees above, possibly with additional structure such as an ordering or heap ordering. It is easy to check that the ψ map makes R into a left H-module.

Let χ denote the map

$$k<F_1, \ldots, F_M> \rightarrow \mathrm{End}(R)$$

defined by using the substitution (1) and simplifying to obtain an endormorphim of R.

Lemma 2.1 *(i) The map ψ is an algebra homomorphism*
(ii) and $\chi = \psi \circ \phi$.

PROOF: The proof of (i) is a straightforward verification and is contained in [8]. Since χ and $\psi \circ \phi$ agree on the generating set $E_1, \ldots, E_M$, part (ii) follows from part (i).

In the later sections, we will also require two other products defined on families of trees. Given $t_1, t_2 \in \mathcal{LT}(F_1, \ldots, F_M)$, define the *meld product* $t_2 \odot t_1$ to be the labeled tree obtained by identifying the roots of the two trees. The meld product is then extended to all of $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ by linearity. Given a derivation $F \in \mathrm{Der}(R)$, let t_2 be the tree $\phi(F)$ and let $t_1 \in \mathcal{LT}(F_1, \ldots, F_M)$. Recall t_2 is a tree consisting of a root and a node laveled F. We define the *composition product* $t_2 \circ t_1$ to be the tree formed by attaching the subtrees whose roots are the children of the root of t_1 to the node labeled F of the tree t_2.

3 Trees and Taylor Series

Fix a Lie group G of dimension N, with Lie algebra g, and let R denote a ring of infinitely differentiable functions on G. We let $\exp : g \longrightarrow G$ denote the exponential map.

Fix a basis of the Lie algebra g consisting of left invariant vector fields $Y_1, \ldots, Y_N$. We will need a map

$$\sharp : R^N \longrightarrow R \otimes g, \quad (a_1, \ldots, a_N) \mapsto \sum_{\mu=1}^{N} a_\mu Y_\mu$$

and its inverse, which we denote $\flat$. We usually write these maps as superscripts, as in $(a_1, \ldots, a_N)^\sharp$.

We are interested in derivations F of the form

$$F = \sum_{\mu=1}^{N} a^\mu Y_\mu, \quad a^\mu \in R, \quad \mu = 1, \ldots, N$$

and the corresponding differential equation

$$\dot{x}(t) = F(x(t)), \quad x(0) = p \in G. \qquad (2)$$

Let $\exp(tF)(x)$ denote the resulting of flowing for time t along the trajectory of (2) through the initial point $p \in G$. We require two lemmas concerned with Taylor series expansion of a solution of (2). These lemmas will use the maps ϕ and ψ defined in the previous section.

If α is a tree, define the *exponential* and *Meld-exponential* of a tree by the formal power series

$$\exp(t\alpha) = 1 + t\alpha + \frac{t^2}{2!}\alpha^2 + \frac{t^3}{3!}\alpha^3 + \cdots$$

$$\mathrm{Mexp}(t\alpha) = 1 + t\alpha + \frac{t^2}{2!}\alpha \odot \alpha + \frac{t^3}{3!}\alpha \odot \alpha \odot \alpha + \cdots.$$

Lemma 3.1 *Assume $f \in R$ and $F \in \mathrm{Der}(R)$. Then*

1. $$(F^k f)(x) = \frac{d^k}{dt^k} f(\exp(tF)x) \mid_{t=0} .$$

2. *If f is analytic near x, then for sufficiently small t,*

   $$f(\exp(tF)x) = \sum_{k=0}^{\infty} f(x; F^k)\frac{t^k}{k!},$$

   where $f(x; F^k)$ is defined to be $(F^k f)(x)$.

3. *If f is analytic near x, then for sufficiently small t,*

   $$f(\exp(tF)x) = \psi(\exp(t\phi(F)))f|_x,$$

   where $\alpha = \phi(F)$.

PROOF. Assertions (1) and (2) can be found in [12]. Since ϕ is an algebra homomorphism, $\phi(F^k) = \alpha^k$. Assertion (3) then follows immediately from Assertion (2). ∎

Lemma 3.2 *Assume $f \in R$ and $F \in \mathrm{Der}(R)$ is left-invariant. Let $\alpha = \phi(F)$. Then*

1.
$$f(\exp(tF)x) = f(x) + tDf(x) \cdot F(x) + \frac{t^2}{2!} D^2 f(x)(F(x), F(x)) + \cdots.$$

2.
$$f(\exp(tF)x) = \psi(\mathrm{Mexp}(t\alpha)) \cdot f|_x.$$

3. *If $G \in \mathrm{Der}(R)$,*
$$\sharp(\flat(G)(\exp(tF)x)) = \psi(\beta \circ \mathrm{Mexp}(t\alpha)),$$
where $\beta = \phi(G)$.

PROOF. Assertion (1) is simply Taylor's theorem. Assertion (2) follows from Assertion (1) and the definition of the ψ map, since left-invariant vector fields have "constant coefficients" with respect to the basis Y_μ. Assertion (3) follows from Assertion (2) and the definition of the ψ, flat and sharp maps. ∎

4 The algorithm

We are interested in numerical algorithms of the Runge-Kutta type to approximate solutions of

$$\dot{x}(t) = F(x(t)), \qquad x(0) = p \in G,$$

where $F \in \mathrm{Der}(R)$. The algorithm depends upon constants c_i and c_{ij}, for $i = 1, \ldots, k$ and $j < i$. For fixed constants, define the following elements of the Lie algebra g

$$\begin{aligned}
\bar{F}_1 &= \sum_{\mu=1}^{N} a^\mu(\nu_0) Y_\mu \in g \\
\bar{F}_2 &= \sum_{\mu=1}^{N} a^\mu(\exp(hc_{21}\bar{F}_1) \cdot \nu_0) Y_\mu \in g \\
\bar{F}_3 &= \sum_{\mu=1}^{N} a^\mu(\exp(hc_{32}\bar{F}_2) \cdot \exp(hc_{31}\bar{F}_1) \cdot \nu_0) Y_\mu \in g \\
&\vdots
\end{aligned}$$

These arise by "freezing the coefficients" of F at various points along the flow of F.

Algorthm 1. Version 1. Let $x_0 = p$ and put

$$x_{n+1} = \exp hc_k \bar{F_k} \cdots \exp hc_1 \bar{F_1} x_n,$$

for $n \geq 0$.

Version 2. Let $x_0 = p$ and put

$$x_{n+1} = \exp\left(hc_k \bar{F_k} + \cdots + \exp hc_1 \bar{F_1}\right) x_n,$$

for $n \geq 0$.

5 Necessary conditions

We prepare with two lemmas.

Lemma 5.1 *Let $f \in R$ and*

$$X_i = \phi(\bar{F_i}) \in k\{\mathcal{LT}(F_1, \ldots, F_M)\}[[h]].$$

Then

$$\begin{aligned}
\bar{F_1}(f) &= \bar{\psi}(\phi(\bar{F}))(f) \\
\bar{F_2}(f) &= \bar{\psi}(\phi(\bar{F}) \circ \mathrm{Mexp}(hc_{21} X_1))(f) \\
\bar{F_3}(f) &= \bar{\psi}(\phi(\bar{F}) \circ \mathrm{Mexp}(hc_{31} X_1 \odot \mathrm{Mexp}(hc_{32} X_2)(f) \\
&\vdots
\end{aligned}$$

Here $\bar{\psi}$ is essentially the ψ map followed by "freezing the coefficients" at ν_0. More precisely,

$$\bar{\psi} : k\{\mathcal{LT}(\bar{F_1}, \ldots, \bar{F_M})\} \rightarrow \mathrm{End}(R).$$

We do this in several steps.

Step 1. Given a labeled tree $t \in \mathcal{LT}_m(\bar{F_1}, \ldots, \bar{F_M})$, assign the root the number 0 and assign the remaining nodes the numbers $1, \ldots, m$. From now on we identify the node with the number assigned to it. Let $j \in$ nodes t , and suppose that $l, \ldots, l'$ are the children of j and that j is labeled with F_{γ_j}. Fix $\mu_l, \ldots, \mu_{l'}$ with

$$1 \leq \mu_l, \ldots, \mu_{l'} \leq N$$

and define

$$
\begin{aligned}
R(j;\mu_l,\ldots,\mu_{l'}) &= Y_{\mu_l}\cdots Y_{\mu_{l'}} a_{\gamma_j}^{\mu_j}(\nu_0) \\
&\qquad \text{if } j \text{ is not the root} \\
&= Y_{\mu_l}\cdots Y_{\mu_{l'}} \\
&\qquad \text{if } j \text{ is the root }.
\end{aligned}
$$

We abbreviate this to $R(j)$.

Step 2. Define

$$\bar{\psi}(t) = \sum_{\mu_1,\ldots,\mu_m=1}^{N} R(m)\cdots R(1)R(0).$$

Step 3. Extend ψ to all $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ by k-linearity.

It is useful to have an intrinsic characterization of the elements $X_i \in k\{\mathcal{LT}(F_1, \ldots, F_M)\}[[h]]$. Order the labels $F_1, \ldots, F_M$ according to their subscripts: $F_1 < \cdots < F_M$. Let $k\{\mathcal{LOHOT}(F_1, \ldots, F_M)\}$ denote those elements of $k\{\mathcal{LT}(F_1, \ldots, F_M)\}$ satisfying

1. The nodes are heap ordered with respect to the labels $F_1, \ldots, F_M$; in other words, the label of a child of a node is (strictly) smaller than the label of the node itself.

2. The children of a node are ordered with respect to the labels $F_1, \ldots, F_M$; in other words, the labels of the children of a node are nondecreasing.

Using ordered, heap ordered trees it is easy to keep track of the constants c_i and c_{ij} that arise in Taylor series computations. To do this we define a map analogous to the ψ map.

Define

$$\rho : k\{\mathcal{LOHOT}(F_1, \ldots, F_M)\} \to \mathrm{End}(R)$$

as follows

Step 1. Given a labeled tree $t \in \mathcal{LOHOT}(F_1, \ldots, F_M)$, with $m+1$ nodes, assign the root the number 0 and assign the remaining nodes the numbers $1, \ldots, m$. From now on we identify the node with the number assigned to it. Fix a node j of t and let $l, \ldots, l'$ denote its children. Let F_{γ_j} denote the label of node j. Let p_i denote the number of children of j labeled with the label F_i, for $i = 1, \ldots, M$. Let $|p|$ denote $p_1 + \cdots + p_M$. Fix $\mu_l, \ldots, \mu_{l'}$ with

$$1 \leq \mu_l, \ldots, \mu_{l'} \leq N$$

and define

$$\begin{aligned} R(j; \mu_l, \ldots, \mu_{l'}) &= \frac{h^{|p|} c_{jl} \cdots c_{jl'}}{p_1! \cdots p_M!} Y_{\mu_l} \cdots Y_{\mu_{l'}} a_{\gamma_j}^{\mu_j}(\nu_0) \\ &\quad \text{if } j \text{ is not the root} \\ &= Y_{\mu_l} \cdots Y_{\mu_{l'}} \\ &\quad \text{if } j \text{ is the root .} \end{aligned}$$

We abbreviate this to $R(j)$.

Step 2. Define

$$\rho(t) = \sum_{\mu_1, \ldots, \mu_m = 1}^{N} R(m) \cdots R(1) R(0).$$

Step 3. Extend ρ to all $k\{\mathcal{COHOT}(F_1, \ldots, F_M)\}$ by k-linearity.

Lemma 5.2 *Let $X_i = \phi(\bar{F}_i)$ and $f \in R$. Then*

$$X_i(f) = \sum \rho(t)(f),$$

where the sum is over all trees $t \in \mathcal{COHOT}(F_1, \ldots, F_M)$ satisfying (i) t consists of $i+1$ or fewer nodes; (ii) the root of the tree has a single child labeled F_i.

It is now straigtforward to derive the following necessary condition for a kth order Runge-Kutta algorithm on a group.

Theorem 5.1 *A necessary condition for a Runge-Kutta method of order k on a group is that for each rooted, ordered tree t consisting of $k+1$ or fewer nodes*

$$\sum \rho(t) = \frac{1}{(\#(nodes\ (\ t)) - 1)!},$$

where the sum is over all $t \in \mathcal{COHOT}(F_1, \ldots, F_M)$ having the same topology as t.

References

[1] J. C. Butcher, "An order bound for Runge-Kutta methods," *SIAM J. Numerical Analysis,* **12** (1975), pp. 304–315.

[2] J. C. Butcher, *The Numerical Analysis of Ordinary Differential Equations,* John Wiley, 1986.

[3] A. Cayley, "On the theory of the analytical forms called trees," in *Collected Mathematical Papers of Arthur Cayley,* Cambridge Univ. Press, Cambridge, 1890, Vol. 3, pp. 242–246.

[4] A. Cayley, "On the analytical forms called trees. Second part," in *Collected Mathematical Papers of Arthur Cayley,* Cambridge Univ. Press, Cambridge, 1891, Vol. 4, pp. 112–115.

[5] R. Grossman, "The evaluation of expresssions involving higher order derivations," *Journal of Mathematical Systems, Estimation, and Control,* Vol. 1, 1990.

[6] R. Grossman and R. Larson, "The symbolic computation of derivations using labeled trees," submitted for publication.

[7] R. Grossman and R. Larson, "Hopf-algebraic structure of combinatorial objects and differential operators," *Israeli J. Math.,* to appear.

[8] R. Grossman and R. Larson, "Solving nonlinear equations from higher order derivations in linear stages," *Advances in Mathematics,* Vol. 82 (1990), pp. 180–202.

[9] R. Grossman and R. Larson, "Hopf algebraic structures of families of trees," *J. Algebra,* Vol. 26 (1989), pp. 184–210.

[10] R. Grossman and R. Larson, "Labeled trees and the algebra of differential operators," *Algorithms and Graphs,* B. Richter, editor, American Mathematical Society, Providence, 1989, pp. 81–87.

[11] R. Grossman and R. Larson, "Labeled trees and the efficient computation of derivations," in *Proceedings of 1989 International Symposium on Symbolic and Algebraic Computation,* ACM, 1989, pp. 74-80.

[12] V. S. Varadarajan, "Lie Groups, Lie Algebras, and their Representations," Prentice-Hall, Inc., 1974.w