The Explicit Computation of Integration Algorithms and First Integrals for Ordinary Differential Equations With Polynomial Coefficients Using Trees

P. E. Crouch*
Arizona State University

R. L. Grossman†
University of Illinois at Chicago

Abstract

This note is concerned with the explicit symbolic computation of expressions involving differential operators and their actions on functions. The derivation of specialized numerical algorithms, the explicit symbolic computation of integrals of motion, and the explicit computation of normal forms for nonlinear systems all require such computations. More precisely, if $R = k[x_1, \ldots, x_N]$, where $k = \mathbf{R}$ or $\mathbf{C}$, F denotes a differential operator with coefficients from R, and $g \in R$, we describe data structures and algorithms for efficiently computing $F \cdot g$. The basic idea is to impose a multiplicative structure on the vector space with basis the set of finite rooted trees and whose nodes are labeled with the coefficients of the differential operators. Cancellations of two trees with $r+1$ nodes translates into cancellation of $O(N^r)$ expressions involving the coefficient functions and their derivatives.

1 Introduction

This note is concerned with the explicit symbolic computation of expressions involving differential operators and their actions on functions. The derivation of specialized numerical algorithms [5], the explicit symbolic computation of integrals of motion [17], and the explicit computation of normal forms for nonlinear systems [13] all require such computations. The following problem is interesting, since in the cases just mentioned cancellations take place.

Problem. Let $R = k[x_1, \ldots, x_N]$, where $k = \mathbf{R}$ or $\mathbf{C}$, let F denote a differential operator with coefficients from R, and let $g \in R$. Compute $F \cdot g$ using as few operations as possible.

In this note, we specialize the algorithms in [11], [12] and [9] to differential operators F with polynomial coefficients and to polynomial rings R and apply the calculus for illustrative purposes to a version of Duffing's equation. By restricting to the special case of polynomial data, it is possible to refine the calculus described in our earlier work.

The basic idea is to introduce a data structure involving trees to code the effect of higher order versions of Leibntz's rule. Essential properties of this data structure are described in Theorem 3.1. These data structures naturally lead to algorithms which are more efficient than naive ones, since the cancellation of two trees with $r+1$ nodes is equivalent to the cancellation of $O(N^r)$ terms from R. These algorithms are described in Theorem 3.2 and the subsequent remarks.

This problem is commonly attacked by viewing the higher order derivations which arise as elements of a tensor space and reducing the computations to questions about linear algebra. This approach is illustrated in [6] and [17]. The computations of higher order derivations requires a lot of bookkeeping. Combinatorial approaches have also been used to keep track of this [14], [15], [16]. In contrast, our viewpoint is algebraic and exploits the natural algebraic structure that trees enjoy.

Section 2 contains a motivating example. Section 3 describes our data structures and algorithms and Section 4 applies them to our original example.

*This research is supported in part by the NSF grant INT 891-4643 and DMS 910-1964.

†This research is supported in part by NASA grant NAG2-513, NSF grant DMS 910-1089 and the Laboratory for Advanced Computing.

ISSAC '92-7/92/CA, USA

2 A simple example

Consider a nonlinear differential equation with polynomial coefficents of the form

$$\dot{x}(t) = F(x(t)), \quad x(0) = x^0 \in \mathbf{R}^N. \qquad (1)$$

To be explicit, write F as

$$F = \sum_{\mu_1=1}^{N} b^{\mu_1} D_{\mu_1} + \sum_{\mu_1,\mu_2=1}^{N} b^{\mu_1}_{\mu_2} x^{\mu_2} D_{\mu_1} + \sum_{\mu_1,\mu_2,\mu_3=1}^{N} b^{\mu_1}_{\mu_2,\mu_3} x^{\mu_3} x^{\mu_2} D_{\mu_1} + \cdots, \qquad (2)$$

where $b^{\mu_1}_{\mu_2,\ldots,\mu_k} \in \mathbf{R}$ and at most a finite number are nonzero. The explicit symbolic computation of series approximating the solutions of Equation 1 in terms of the coefficients b^{μ_1}, $b^{\mu_1}_{\mu_2}$, $b^{\mu_1}_{\mu_2,\mu_3}$, …, etc. is straightforward, but expensive. Typically, cancellation takes place. An interesting problem is to introduce a data structure, and an algorithm which takes advantage of it, to exploit cancellations to reduce the cost of computing series which approximate the solutions of the equation.

To be more precise, consider the following version of Duffing's equation:

Example 2.1

$$\begin{aligned} \dot{x}_1(t) &= x_2 \\ \dot{x}_2(t) &= -\eta_1 x_1 - \eta_2 x_1^3. \end{aligned} \qquad (3)$$

This is a simplified model describing the buckling of a beam, with a cubic nonlinearity. Here x_1 denotes the deflection of the beam from equilibrium, x_2 is the velocity of the beam at this point, and η_1 and η_2 are constants. The following function is conserved

$$h(x_1, x_2) = \frac{\eta_1}{2} x_1^2 + \frac{1}{2} x_2^2 + \frac{\eta_2}{4} x_1^4$$

in the sense that the map $t \mapsto h(x_1(t), x_2(t))$ is constant along any solution $t \mapsto (x_1(t), x_2(t))$ of the system.

It is useful for symbolic computations to view this system algebraically. To do this, introduce the differential operator

$$F = x_2 \frac{\partial}{\partial x_1} + -(\eta_1 x_1 + \eta_2 x_1^3) \frac{\partial}{\partial x_2},$$

which can be viewed as acting as a derivation on the polynomial algebra, $\mathbf{R}[x_1, x_2]$, for example. From this view point, to check that the energy h is conserved to order $r+1$, it is sufficient to check that

$$F \cdot h|_{x(0)} = 0, \ldots, F^r \cdot h|_{x(0)} = 0. \qquad (4)$$

We return now to the general case of the nonlinear system 1. Consider next the derivation of a multistep numerical algorithm to integrate the system. Fix a step size δ and write $x^n = (x_1^n, \ldots, x_N^n)$ for the approximation to the trajectory $x(t)$ of the system at time $t = n\delta$. By choosing constants α_0, …, α_r suitably the update

$$\begin{aligned} x^n &= \alpha_0 x^{n-1} + \delta\alpha_1 F(x^{n-1}) + \delta\alpha_2 F(x^{n-2}) \\ &\quad + \cdots + \delta\alpha_r F(x^{n-r}), \quad \text{for } n \geq 1 \end{aligned}$$

approximates the discrete time Taylor flow

$$x^n = x^{n-1} + \mathcal{T}F(x^{n-1}),$$

where

$$\begin{aligned} \mathcal{T}F &= \delta F + \frac{\delta^2}{2!} DF \cdot F \\ &\quad + \frac{\delta^3}{3!}(D^2F(F,F) + DF \cdot DF \cdot F) + \cdots \end{aligned}$$

to order r, and hence the flow of the orginal system to the same order. Multi-step methods are easy to derive in a variety of ways. One of the simplest, but not the most efficient, is to use the method of undetermined coefficients. Define the vector field G by

$$G = \alpha_0 G_0 + \delta\alpha_1 G_1 + \cdots + \delta\alpha_r G_r,$$

where

$$\begin{aligned} G_0 &= -\delta F + \frac{\delta^2}{2!} DF \cdot F \\ &\quad - \frac{\delta^3}{3!}(D^2F(F,F) + DF \cdot DF \cdot F) + \cdots \\ G_1 &= F - \delta DF \cdot F \\ &\quad + \frac{\delta^2}{2!}(D^2F(F,F) + DF \cdot DF \cdot F) + \cdots \\ G_2 &= F - 2\delta DF \cdot F \\ &\quad + \frac{(2\delta)^2}{2!}(D^2F(F,F) + DF \cdot DF \cdot F) + \cdots \\ &\;\;\vdots \\ G_r &= F - r\delta DF \cdot F \\ &\quad + \frac{(r\delta)^2}{2!}(D^2F(F,F) + DF \cdot DF \cdot F) + \cdots \end{aligned}$$

Then the algorithm is order r in case the constants are chosen so that

$$\mathcal{T}F - G = O(\delta^{r+1}). \qquad (5)$$

We return to these computations in the last section.

3 The algebra of Cayley trees

In this section, we define an algebra structure on spaces of trees and describe algorithms to simplify expressions involving differential operators. The relation between trees and differential operators goes back at least as far as Cayley [3] and [4]. The work most closely related to the view point taken here is Butcher's use of trees to analyze Runge-Kutta algorithms [1] and [2].

Let k denote the real or complex numbers. By a *tree* we will mean a finite rooted tree. Let $\mathcal{T}$ be the set of finite rooted trees, and let $k\{\mathcal{T}\}$ be the k-vector space which has $\mathcal{T}$ as a basis.

We now define an algebra structure on $k\{\mathcal{T}\}$. Suppose that σ_1, $\sigma_2 \in \mathcal{T}$ are trees. Let $\tau_1, \ldots, \tau_r$ be the children of the root of σ_1. If σ_2 has $n+1$ nodes (counting the root), there are $(n+1)^r$ ways to attach the r subtrees of σ_1 which have $\tau_1, \ldots, \tau_r$ as roots to the tree σ_2 by making each τ_i the child of some node of σ_2. The product $\sigma_1\sigma_2$ is defined to be the sum of these $(n+1)^r$ trees. It can be shown that this product is associative, and that the trivial tree consisting only of the root is a right and left unit for this product. The algebra $k\{\mathcal{T}\}$ is graded: $k\{\mathcal{T}\}_n$ has as basis all trees with $n+1$ nodes.

It can also be shown that if each node of the tree (except for the root) is labeled, then the same product turns the vector space $k\{\mathcal{LT}\}$, whose basis consists of labeled trees, into a graded algebra. The multiplication is illustrated in Figure 4. For details, see [10]. We summarize this discussion with the following theorem.

Theorem 3.1 *(i) The vector space $k\{\mathcal{T}\}$ with basis the set of finite rooted trees is a graded algebra. (ii) The vector space $k\{\mathcal{LT}\}$ with basis the set of finite rooted trees, all of whose nodes (except for the root) are labeled, is a graded algebra.*

Setup. Let R denote the polynomial ring $k[x_1, \ldots, x_N]$. Assume that we are given derivations F, G, ... of R, as in Equation 2, whose coefficients are respectively $b^{\mu_1}_{\mu_2,\ldots,\mu_j}$, $c^{\mu_1}_{\mu_2,\ldots,\mu_j}$, $\ldots \in k$. Let $B = k\{\mathcal{LT}\}$ denote the Cayley algebra of trees labeled with the symbols b, c, ... and the symbol x. Let A denote the free associative algebra over k generated by F, G, ... Then since A is free, it is easy to check that the map defined in Figure 1 on the generators F, G, ... extends to an algebra homorphism $\phi : A \longrightarrow B$.

We now define an action of the algebra of Cayley trees

$$B = k\{\mathcal{LT}\}$$

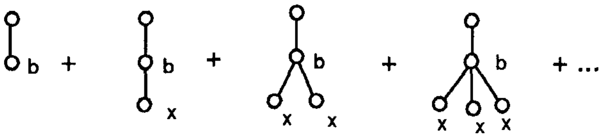

Figure 1: The image under ϕ of the generator F of A.

on the ring R which captures the action of trees as higher derivations. This is illustrated in Figure 2. The action is defined using the map

$$\psi : k\{\mathcal{LT}\} \rightarrow \mathrm{End}_k\, R,$$

as follows:

1. Given a labeled, ordered tree σ with $m+1$ nodes, assign the root the number 0 and assign the remaining nodes the numbers $1, \ldots, m$. Identify the node with the number assigned to it. To node j associate a dummy summation index μ_j. Denote $(\mu_1, \ldots, \mu_m)$ by μ.
2. Let j be a node of σ, and let $l, \ldots, l'$ be the children of j, labeled with a symbol b, c, ... representing a derivation of R. Define $R(j;\mu)$ as follows:
   (a) If the node j is labeled with x, set $R(j,\mu) = D_{\mu_l} \cdots D_{\mu_{l'}} x^{\mu_j}$.
   (b) If the node j is labeled with b, say, set
   $$R(j,\mu) = D_{\mu_l} \cdots D_{\mu_{l'}} b^{\mu_j}_{\mu_l \cdots \mu_{l'}}.$$
3. Define
$$\psi(\sigma) = \sum_{\mu_1,\ldots,\mu_m=1}^{N} R(m;\mu) \cdots R(1;\mu) R(0;\mu).$$
4. Extend ψ to all of $k\{\mathcal{LT}\}$ by linearity.

The following theorem follows directly from this construction.

Theorem 3.2 *Let R denote the polynomial algebra $k[x_1, \ldots, x_N]$. Let B denote the algebra of Cayley trees $k\{\mathcal{LT}\}$. Then R is a B-module with respect to the action defined by ψ and satisfies*

$$(\sigma\tau)(f) = \sigma(f)\tau(f), \quad \sigma, \tau \in B, \quad f \in R.$$

Moreover, if σ, $\tau \in k\{\mathcal{T}\}_r$, then the cancellation $\sigma - \tau = 0$ implies that $2N^r$ terms in $\mathrm{End}_k\, R$ cancel.

Remark 3.1 Note that $\psi(\sigma)$ is 0, if the tree σ contains a node labeled x which has two or more children labeled with b, c, ... representing one of the derivations. Note also that the number of children of the root of σ determine the degree of the differential operator defined by $\psi(\sigma)$.

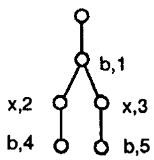

Figure 2:
The tree is sent by ψ to the following first order differential operator

$$\begin{aligned} & \sum_{\mu_1=1,\dots,\mu_5=1}^{N} b^{\mu_5} b^{\mu_4} D_{\mu_4}(x^{\mu_2}) D_{\mu_5}(x^{\mu_3}) b^{\mu_1}_{\mu_2,\mu_3} D_{\mu_1} \\ = & \sum_{\mu_1=1,\mu_2=1,\mu_3=1}^{N} b^{\mu_3} b^{\mu_2} b^{\mu_1}_{\mu_2,\mu_3} D_{\mu_1} \end{aligned}$$

Here D_μ is equal to $\frac{\partial}{\partial x_\mu}$ and a node is labeled with a symbol together with an integer indicating the subscript of its summation index.

Remark 3.2 Since A is generated by symbols F, G, ... representing derivations of R, there is natural map

$$\chi : A \longrightarrow \mathrm{End}_k\, R$$

defined by sending the formal expression to the higher order derivation of R it represents. The maps are defined so that the following diagram commutes:

$$\begin{array}{ccc} A & \rightarrow & B \\ & \searrow & \downarrow \\ & & \mathrm{End}_k\, R \end{array} \qquad (6)$$

Remark 3.3 Simple examples involving the simplification of expressions in A, such as arised in the first section, show that it is often more efficient to compute using the algebra B than directly in the algebra A.

4 Some Applications

In this section, we describe two applications of the Cayley algebra of trees to the symbolic computation of expressions arising in the analysis of Duffing's Equation 3 described in Section 2.

Example 4.1 We begin by describing how the algebra of Cayley trees can be used to compute conserved functions. Figure 3 illustrates the Cayley trees associated with $\phi(F)$. Figure 4 illustrates some of the trees that arise when computing $F \cdot F$. Denote the sum of trees in Figure 4 by σ. Then $\psi(\sigma) \cdot f = 0$, for f of the form

$$f = f_0 + \sum_{\mu_1=1}^{N} f_{\mu_1} x^{\mu_1} + \sum_{\mu_1=1,\mu_2=1}^{N} f_{\mu_1,\mu_2} x^{\mu_1} x^{\mu_2}$$

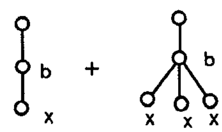

Figure 3:
The image $\phi(F)$ of the Duffing operator in the algebra of Cayley trees. Note that $b^1_1 = -\eta_1$, $b^1_2 = 1$, $b^2_{111} = -\eta_2$, and all the other coefficients are 0.

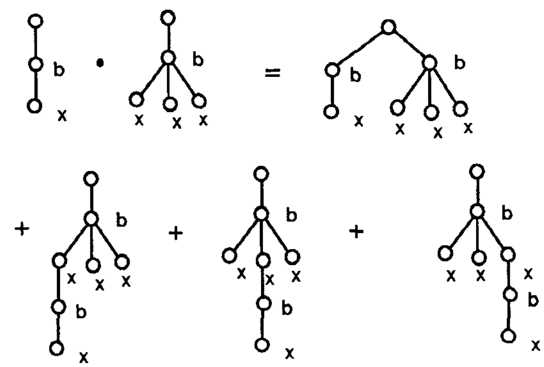

Figure 4:
Some of the trees in the Cayley algebra that arise when computing $F \cdot F$.

yields

$$\begin{aligned} & \sum_{\mu_1,\dots,\mu_6=1}^{N} x^{\mu_2} b^{\mu_1} b^{\mu_3} x^{\mu_4} x^{\mu_5} x^{\mu_6} f_{\mu_1,\mu_3} \\ + \; 3 & \sum_{\mu_1,\dots,\mu_5=1}^{N} x^{\mu_3} x^{\mu_4} x^{\mu_5} b^{\mu_2}_{\mu_3} b^{\mu_1}_{\mu_2,\mu_4,\mu_5} f_{\mu_1} = 0. \end{aligned}$$

Here $N = 2$. The equations for the other trees arising in $F \cdot F$ are handled in the same way. Note that this calculus can be used in this way to compute whether a given function f is conserved to a specified order and to compute the equations specifiying the relations among the coefficients for a function f with undetermined coefficients.

Example 4.2 In this example, we show how the Cayley calculus of trees can be used to derive specialized integration algorithms. First view the differential operator F specifying the dynamics of the Duffing system as the sum of two operators $F_0 + F_1$, where

$$\begin{aligned} F_0 &= x_2 \frac{\partial}{\partial x_1} - \eta_1 x_1 \frac{\partial}{\partial x_2} \\ F_1 &= a(x) \frac{\partial}{\partial x_2}, \quad a(x) = -\eta_2 x_1^3. \end{aligned}$$

Note that the system $\dot{x} = F_0(x(t))$ is a linear system and can be explicitly integrated. "Freezing" the coefficient $a(x)$ of F_1, yields a system F which is explicitly integrable. Let

$$\mathcal{F}(F)(y) = F_0 + a(y)F_1$$

denote this sytem and let $t \mapsto x(t) = \exp(\mathcal{F}(F)(y)) \cdot x^0$ denote the corresponding solution through the initial point x^0. The computation of $x(t)$ does not require an integration, but simply the evaluation of a sine and cosine. Consider now a multi-step integration algorithm of the form

$$\begin{aligned} x^{k+1} &= \exp(\delta\alpha_0\mathcal{F}(F)(x^k))\exp(\delta\alpha_1\mathcal{F}(F)(x^{k-1})) \\ &\quad \cdot \exp(\delta\alpha_2\mathcal{F}(F)(x^{k-2}))\exp(\delta\alpha_3\mathcal{F}(F)(x^k))x^k. \end{aligned}$$

To determine the constants $\alpha_i \in \mathbf{R}$, we can simply use the Cayley algebra of trees to compute

$$(\mathcal{T}F - G) \cdot (x^\mu), \quad \mu = 1, \ldots, N,$$

where

$$\begin{aligned} G &= \exp(\delta\alpha_0\mathcal{F}(F)(x^k))\exp(\delta\alpha_1\mathcal{F}(F)(x^{k-1})) \\ &\quad \cdot \exp(\delta\alpha_2\mathcal{F}(F)(x^{k-2}))\exp(\delta\alpha_3\mathcal{F}(F)(x^k)) \end{aligned}$$

and we view x^μ as a coordinate *function*. This gives equations for the unknown α_i. It turns out that

$$\alpha_0 = \frac{17}{72}, \quad \alpha_1 = \frac{-4}{3}, \quad \alpha_2 = \frac{5}{12}, \quad \alpha_3 = \frac{121}{72}.$$

See [7] and [8] for a further description of these types of algorithms.

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] A. Chorin, T. J. R. Hughes, J. E. Marsden, and M. McCracken, "Product Formulas and Numerical Algorithms," *Comm. Pure and Appl. Math.*, Vol. 31, pp. 205–256, 1978.

[6] G. Chen, J. Della Dora, L. Stolovitch, "Nilpotent normal form via Carleman linearization," *Proceedings of 1991 International Symposium on Symbolic and Algebraic Computation*, ACM, 1991, pp. 281–288.

[7] P. E. Crouch and R. L. Grossman, "Numerical integration of ordinary differential equations on manifolds," submitted to *Journal of Nonlinear Science.*

[8] P. E. Crouch, R. L. Grossman, and R. G. Larson, "Trees, bialgebras, and intrinsic numerical algorithms," *Laboratory for Advanced Computing Technical Report* Number LAC90-R23, University of Illinois at Chicago, May, 1990.

[9] P. Crouch, R. Grossman, and R. G. Larson, "Computations involving differential operators and their actions on functions," *Proceedings of 1991 International Symposium on Symbolic and Algebraic Computation,* ACM, 1991, pp. 301–307.

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

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

[12] R. Grossman and R. Larson, "The symbolic computation of derivations using labeled trees," *Journal of Symbolic Computation,* to appear.

[13] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields,* Springer, 1984, pp. 138–145.

[14] P. Leroux and X. G. Viennot, "Combinatorial resolution of systems of differential equations, I: Ordinary differential equations," in *Combinatoire Énumérative, UQAM 1985, Proceedings, Lecture Notes in Mathematics,* Volume 1234, Springer-Verlag, 1986, pp. 210–245.

[15] P. Leroux and X. G. Viennot, "Combinatorial systmes of differential equations, IV: Separation of variables," in *Proceedings of First Japan Conference on Graph Theory and Applications, June, 1986, Discrete Mathematics,* Volume 72, pp. 237–250, 1988.

[16] P. Leroux and X. G. Viennot, "A combinatorial approach to nonlinear functional expansions," *27th Conference on Decision and Control,* Austin, Texas, pp. 1314–1319, 1988.

[17] F. Schwarz, "Existence theorems for polynomial first integrals," in *Proceedings of 1991 International Symposium on Symbolic and Algebraic Computation,* ACM, 1991, pp. 256–264.