



B-series and B-series Coefficients¹

J. C. Butcher²

Department of Mathematics,
University of Auckland,
Auckland, New Zealand

Received February 8, 2010; accepted in revised form February 22, 2010.

Abstract: B-series, together with the algebraic system which underpins them, are essential tools in the study of properties of numerical methods for evolutionary problems. This paper surveys the properties of these constructs and relates the theory to applications in numerical analysis.

© 2010 European Society of Computational Methods in Sciences and Engineering

Keywords: B-series, Runge–Kutta methods, order conditions, effective order, singly-implicit methods.

Mathematics Subject Classification: 65L05, 65L06

1 Introduction

B-series are formal series of the form

$$B(a, y) = a(\emptyset)y + \sum_{t \in T} \frac{a(t)h^{r(t)}}{\sigma(t)} F(t)(y), \quad (1)$$

where $F(t)(y)$ is the elementary differential associated with the tree t and the differential equation $y'(x) = f(y(x))$ and evaluated at the point y . They were introduced by E. Hairer and G. Wanner [8] in 1974 and are related to the algebraic structures introduced by the author [1]. If the elementary differentials do not actually exist, the coefficients $a(t)$ still have a significance and they still characterise the specific Runge–Kutta method from which they were defined. This paper is a survey of the algebraic system behind B-series and the relationship between this algebraic system and numerical methods.

The use of B-series is now well established in numerical analysis and we will include a discussion of some applications including a new way of looking at the order of singly-implicit methods.

In Section 2 a survey will be given of elementary differentials which form the building blocks of B-series expansions.

¹Published electronically October 15, 2010

²Corresponding author. E-mail: butcher@math.auckland.ac.nz

2 Trees and elementary differentials

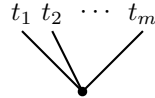
The starting point is the set of rooted-trees:

$$T = \left\{ \bullet, \begin{array}{c} \bullet \\ | \\ \bullet \end{array}, \begin{array}{c} \bullet & \bullet \\ \diagdown & / \\ \bullet \end{array}, \begin{array}{c} \bullet & \bullet & \bullet \\ \diagdown & | & / \\ \bullet \end{array}, \begin{array}{c} \bullet & \bullet & \bullet & \bullet \\ \diagdown & | & / & | \\ \bullet \end{array}, \begin{array}{c} \bullet & \bullet & \bullet & \bullet & \bullet \\ \diagdown & | & / & | & | \\ \bullet \end{array}, \dots \right\}$$

We can generate members of T recursively by starting with

$$\tau = \bullet$$

and using the operation $[\]$ defined by the following diagram for $[t_1 t_2 \cdots t_m]$



Thus, the tree $t = [t_1 t_2 \cdots t_m]$ is formed by introducing a vertex, which serves as the root of t , and attaching the original roots of t_1, t_2, \dots, t_m to the root of t . If there are repetitions amongst the t_i , it is sometimes convenient to write

$$t = [t_1^{k_1} t_2^{k_2} \cdots t_n^{k_n}].$$

Examples of this notation are shown in Table 1. It is possible to construct functions on T using this recursion. The order, defined as the number of vertices and written $r(t)$, satisfies the recursion

$$r(t) = \begin{cases} 1, & t = \tau, \\ 1 + r(t_1) + r(t_2) + \cdots + r(t_m), & t = [t_1 t_2 \cdots t_m]. \end{cases}$$

The symmetry, denoted by $\sigma(t)$ is defined from the recursion

$$\sigma(t) = \begin{cases} 1, & t = \tau, \\ \prod_{i=1}^n k_i! \sigma(t_i)^{k_i}, & t = [t_1^{k_1} t_2^{k_2} \cdots t_n^{k_n}]. \end{cases}$$

The density, $\gamma(t)$, is defined by

$$\gamma(t) = \begin{cases} 1, & t = \tau, \\ r(t) \gamma(t_1) \gamma(t_2) \cdots \gamma(t_m), & t = [t_1 t_2 \cdots t_m]. \end{cases}$$

Given an autonomous initial value problem

$$y'(x) = f(y(x)), \quad y(x_0) = y_0,$$

certain expressions known as “elementary differentials” arise in the study of the Taylor series for this problem. To construct these quantities, we need to introduce values of f, f', f'' evaluated at $y(x)$. Denote these by









$$\mathbf{f}, \quad \mathbf{f}', \quad \mathbf{f}'', \quad \dots$$

The elementary differential $F(t)(y_0)$ is defined making use of $\mathbf{f}, \mathbf{f}', \mathbf{f}''$, etc with $y(x)$ replaced by y_0 .

$$F(t)(y_0) = \begin{cases} \mathbf{f}, & t = \tau, \\ \mathbf{f}^{(m)}(F(t_1)(y_0), F(t_2)(y_0), \dots, F(t_m)(y_0)), & t = [t_1 t_2 \cdots t_m]. \end{cases}$$

The various functions that have been introduced are included in Figure 1.

Table 1: Examples of tree functions

t	$r(t)$	$\sigma(t)$	$\gamma(t)$	$F(t)(y_0)$
	τ	1	1	\mathbf{f}
	$[\tau]$	2	1	$\mathbf{f}'\mathbf{f}$
	$[\tau^2]$	3	2	$\mathbf{f}''(\mathbf{f}, \mathbf{f})$
	$[[\tau]]$	3	1	$\mathbf{f}'\mathbf{f}'\mathbf{f}$
	$[\tau^3]$	4	6	$\mathbf{f}'''(\mathbf{f}, \mathbf{f}, \mathbf{f})$
	$[\tau[\tau]]$	4	1	$\mathbf{f}''(\mathbf{f}, \mathbf{f}'\mathbf{f})$
	$[[\tau^2]]$	4	2	$\mathbf{f}'\mathbf{f}''(\mathbf{f}, \mathbf{f})$
	$[[[\tau]]]$	4	1	$\mathbf{f}'\mathbf{f}'\mathbf{f}'\mathbf{f}$

3 B-series and elementary weights

Recall the formula for B-series (1). The special case in which

$$a(\emptyset) = 1, \quad a(t) = \frac{1}{\gamma(t)}, \quad t \in T,$$

is of particular interest because it is the Taylor expansion for the exact solution to an initial value problem.

In the B-series for the result computed by a Runge–Kutta method, $a(t)$ is equal to the “elementary weight” constructed from the coefficients of the specific method. If the elementary differentials do not actually exist, the coefficients $a(t)$ still have a significance and they still characterise the specific Runge–Kutta method from which they were defined.

Given a Runge–Kutta method defined by the tableau

$$\frac{c}{b^T} \left| \begin{array}{c} A \\ b^T \end{array} \right.,$$

the corresponding elementary weight $\Phi(t)$ for the method and the elementary weight for stage number i , corresponding to a tree t is given by the recursion

$$\begin{aligned} \Phi_i(\tau) &= c_i, & i = 1, 2, \dots, s, \\ \Phi(\tau) &= \sum_{j=1}^s b_j, \\ \Phi_i([t_1 t_2 \dots t_k]) &= \sum_{j=1}^s a_{ij} \Phi_j(t_1) \Phi_j(t_2) \dots \Phi_j(t_k), & i = 1, 2, \dots, s, \\ \Phi([t_1 t_2 \dots t_k]) &= \sum_{j=1}^s b_j \Phi_j(t_1) \Phi_j(t_2) \dots \Phi_j(t_k). \end{aligned}$$

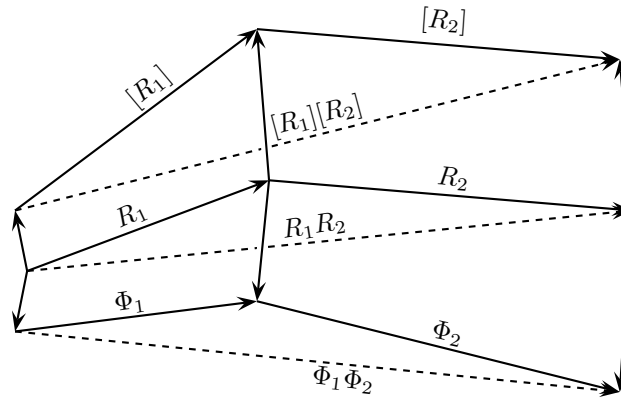


Figure 1: Homomorphisms related to compositions of Runge–Kutta methods

The formal Taylor expansion for the solution computed by a Runge–Kutta method, over a single step of size h , starting from an initial value $y(x_0) = y_0$, is found from

$$B(a, y) = a(\emptyset)y + \sum_{t \in T} \frac{a(t)h^{r(t)}}{\sigma(t)} F(t)(y),$$

by substituting $a(\emptyset) = 1$, $a(t) = \Phi(t)$ and $y = y_0$.

However, the coefficients in the formal expansion still have a significance even when the series does not converge or the terms of the series are not defined.

4 The group of rooted trees mappings

There are several ways in which two Runge–Kutta methods could be regarded as equivalent.

It might be possible, for example, to delete irrelevant or duplicated stages from a method to yield a simpler “reduced” method. If two methods both reduce to the same method, although possibly with their stages re-ordered, they are equivalent in this obvious sense.

This implies, and is also implied by, the fact that, for any Lipschitz continuous problem, they yield identical numerical results, for sufficiently small stepsizes. It is possible to show, by a recursive argument, that the elementary weight $\Phi(t)$, for any specific tree t , has the same value for each of two equivalent methods. This yields a third way of looking at equivalence because the mapping from trees to elementary weights characterises the Runge–Kutta method to within its equivalence class.

By using equivalence classes, the semigroup of Runge–Kutta tableaux, under the composition operation, can be used to form a group. For a Runge–Kutta method R , denote the equivalence class which contains it by $[R]$. It can be shown that the mapping $R \mapsto [R]$ is a homomorphism so that $[R_1][R_2] = [R_1R_2]$.

Furthermore, this group is homomorphic to the set of all mappings from trees to real numbers, with the group operation to be described below. The relationship between these various structures is shown schematically in Figure 1. Let G denote the set of mappings from T to \mathbb{R} . Let a, b denote two members of G . Our aim now is to describe the group operation which gives the value of $(ab)(t)$ for every $t \in T$. For a given tree t let $P(t)$ denote the set of prunings of t . A pruning of t is represented by a subtree \tilde{t} , which shares the same root as t . Let the set of trees formed from the parts of t left over, when \tilde{t} is removed from t , be denoted by $t \setminus \tilde{t}$.

Table 2: Example of tree prunings

Pruning	\tilde{t}	$t \setminus \tilde{t}$

The value of $\widehat{a}(t \setminus \tilde{t})$ is defined as the product of $a(u)$ over each of the subtrees u in $t \setminus \tilde{t}$. This is illustrated in Table 2, using a particular tree t of order 5. The target tree t also appears in the table, from the trivial pruning for which $\tilde{t} = t$.

To find the formula for $(ab)(t)$, the value of $a(t)$ is added to the sum, over all prunings, of $\widehat{a}(t \setminus \tilde{t}) b(\tilde{t})$. An alternative formula for $(ab)(t)$ was given in [1] and in a simplified form in [4] and [5].

For $a \in G$ and $t \in T$, write $\widehat{t}(a) = a(t)$. Also write $1(a) = 1$. Let \widehat{T} denote the span of $\{1, \widehat{\tau}, \widehat{[\tau]}, \dots\}$. If $t = [t_1 t_2 \cdots t_m]$, write

$$\widehat{t} = [\widehat{t}_1 \widehat{t}_2 \cdots \widehat{t}_m]$$

and generalise this appropriately when the \widehat{t}_i ($i = 1, 2, \dots, m$) are replaced by members of \widehat{T} .

Introduce a function $\Lambda(a, t) \in \widehat{T}$ such that

$$(ab)(t) = \Lambda(a, t)(b).$$

Then Λ satisfies the following recursion:

$$\Lambda(a, t) = \begin{cases} a(\tau) + \widehat{\tau}, & t = \tau, \\ a(t) + [\Lambda(a, t_1)\Lambda(a, t_2) \cdots \Lambda(a, t_m)], & t = [t_1 t_2 \cdots t_m]. \end{cases}$$

The product $(a, b) \mapsto ab$ defined in this way is known to be associative and to satisfy the group axioms. It can be extended to a two-operation algebra by allowing $b(\emptyset)$ to have a value other than 1 and admitting vector space operations. It is known to be closely related to the Hopf algebra of Connes and Kreimer [6].

5 Applications in numerical analysis

The standard application is the order conditions for Runge–Kutta methods.

We obtain these conditions by equating the B-series coefficients for a single step of the method, with the limiting method in which the set of stage indices is replaced by the interval $[0, 1]$, the matrix A is replaced by the integration operator

$$Y(x) \mapsto \int_0^x Y(x)dx, \quad (2)$$

and b^\top is replaced by the functional

$$Y(x) \mapsto \int_0^1 Y(x)dx. \quad (3)$$

The B-series coefficients are then given by

$$\begin{aligned} \tilde{\Phi}_x(\tau) &= x, \\ \tilde{\Phi}(\tau) &= 1, \\ \tilde{\Phi}_x([t_1 t_2 \dots t_k]) &= \int_0^x \prod_{j=1}^k \tilde{\Phi}_x(t_j) dx, \\ \tilde{\Phi}([t_1 t_2 \dots t_k]) &= \int_0^1 \prod_{j=1}^k \tilde{\Phi}_x(t_j) dx. \end{aligned}$$

A simple recursion argument shows that $\tilde{\Phi}_x = x^{r(t)}/\gamma(t)$, $\tilde{\Phi} = 1/\gamma(t)$ and the order p conditions $\Phi(t) = 1/\gamma(t)$ (for $r(t) \leq p$) immediately follow.

As a second application we consider the conditions for *effective* order p , which can be expressed using the group product. Let α be the group element for a specific Runge–Kutta method, whose order conditions are sought, and let β be the group element for a method used to perturb the solution. Also let E be the group element for the exact solution. That is, $E(t) = \gamma(t)^{-1}$.

Then the Runge–Kutta method corresponding to α has effective order p if and only if β exists such that $\beta\alpha = E\beta$, up to all trees up to order p . This criterion can be generalised to the condition for a general linear method to have order p .

A final application is to the order conditions for a singly-implicit Runge–Kutta method. Write the tableau for this method in one of the forms

$$\frac{c + \gamma \mathbf{1}}{\quad} \left| \begin{array}{c} A + \gamma I \\ b^\top \end{array} \right. \quad (4)$$

$$\frac{c + \gamma \mathbf{1}}{\quad} \left| \begin{array}{c} A + \gamma I \\ b^\top + \gamma e_s^\top \end{array} \right. \quad (5)$$

The formulation (5) would be used for methods designed so that $b^\top = e_s^\top A$, with the aim of making the stability function vanish at infinity, in contrast to (4), for which there is no such requirement.

For each of these methods, the elementary weights are calculated from the tableau

$$\frac{c}{\quad} \left| \begin{array}{c} A \\ b^\top \end{array} \right.$$

and we will write the order conditions in the form

$$\Phi(t) = \tilde{\Phi}(t, \gamma).$$

Table 3: Modified right-hand sides for singly-implicit order conditions

t	$\tilde{\Phi}_x(t, \gamma)$	$\tilde{\Phi}(t, \gamma)$
•	$x - \gamma$	1
••	$\frac{1}{2}x^2 - 2\gamma x + \gamma^2$	$\frac{1}{2} - \gamma$
•••	$\frac{1}{3}x^3 - 2\gamma x^2 + 3\gamma^2 x - \frac{4}{3}\gamma^3$	$\frac{1}{3} - \gamma + \gamma^2$
••••	$\frac{1}{6}x^3 - \frac{3}{2}\gamma x^2 + 3\gamma^2 x - \gamma^3$	$\frac{1}{6} - \gamma + \gamma^2$
•••••	$\frac{1}{4}x^4 - 2\gamma x^3 + \frac{9}{2}\gamma^2 x^2 - 4\gamma^3 x + \frac{5}{4}\gamma^4$	$\frac{1}{4} - \gamma + \frac{3}{2}\gamma^2 - \gamma^3$
••••••	$\frac{1}{8}x^4 - \frac{4}{3}\gamma x^3 + 4\gamma^2 x^2 - 4\gamma^3 x + \gamma^4$	$\frac{1}{8} - \frac{5}{6}\gamma + \frac{3}{2}\gamma^2 - \gamma^3$
•••••••	$\frac{1}{12}x^4 - \gamma x^3 + \frac{7}{2}\gamma^2 x^2 - \frac{13}{3}\gamma^3 x + \frac{4}{3}\gamma^4$	$\frac{1}{12} - \frac{2}{3}\gamma + \frac{3}{2}\gamma^2 - \frac{4}{3}\gamma^3$
••••••••	$\frac{1}{24}x^4 - \frac{2}{3}\gamma x^3 + 3\gamma^2 x^2 - 4\gamma^3 x + \gamma^4$	$\frac{1}{24} - \frac{1}{2}\gamma + \frac{3}{2}\gamma^2 - \gamma^3$

The values of $\tilde{\Phi}(t, \gamma)$ are based on modifying the linear operator which represents A from (2) to

$$Y(x) \mapsto \int_0^x Y(x)dx - \gamma Y(x).$$

For method (4), (3), which represents b^\top , is retained unchanged. However, for (5), this is replaced by $Y(x) \mapsto \int_0^1 Y(x)dx - \gamma Y(1)$.

We can now define $\tilde{\Phi}(t, \gamma)$ in the case of (4) recursively as follows:

$$\begin{aligned} \tilde{\Phi}_x(\tau, \gamma) &= x - \gamma, \\ \tilde{\Phi}(\tau, \gamma) &= 1, \\ \tilde{\Phi}_x([t_1 t_2 \dots t_k], \gamma) &= \int_0^x \prod_{j=1}^k \tilde{\Phi}_x(t_j, \gamma) dx - \gamma \prod_{j=1}^k \tilde{\Phi}_x(t_j, \gamma), \\ \tilde{\Phi}([t_1 t_2 \dots t_k], \gamma) &= \int_0^1 \prod_{j=1}^k \tilde{\Phi}_x(t_j, \gamma) dx. \end{aligned}$$

In the case of method (5), we see that $\tilde{\Phi}(t, \gamma) = \tilde{\Phi}_1(t, \gamma)$ The values of $\tilde{\Phi}_x(t, \gamma)$ and $\tilde{\Phi}(t, \gamma)$ up to order 4 are shown in Table 3.

6 Some remarkable Runge–Kutta methods

The first method of order six seems to have been constructed by A. Huřa [9], who decided to seek a method with 8 stages because he had 31 order conditions to satisfy and 7 stages would give only 28 free parameters compared with 36 parameters available with 8 stages.

His method made extensive use of what are now called “simplifying assumptions”, but even so, his derivation required intricate manipulations with some formulae filling two and a half pages.

If Huřa had known about order conditions for non-scalar differential equations he would have known that there should have been 37 conditions instead of just 31. Therefore he would probably have assumed that 9 stages would have been necessary.

Surprisingly, his method satisfying the 31 conditions for scalar problems also satisfies the 6 additional conditions for systems of equations [3] Thus he had magically satisfied 37 conditions in 36 unknowns. Even more surprisingly, 6th order methods exist with only 7 stages! This means satisfying the 37 conditions in only 28 unknowns.

According to the theory for scalar differential equations, there are 16 conditions for order 5 but, for vector based differential equations, there are 17 conditions. The reason is that the two elementary differentials

$$\mathbf{f}''(\mathbf{f}, \mathbf{f}'\mathbf{f}) \quad \text{and} \quad \mathbf{f}'\mathbf{f}''(\mathbf{f}, \mathbf{f}'\mathbf{f})$$

correspond to identical terms

$$(ff_{yy} + f_{xy})f_y(ff_y + f_x)$$

in the scalar theory.

The consequence is a confusion of two order conditions $\sum b_i c_i a_{ij} a_{jk} c_k = \frac{1}{30}$ and $\sum b_i a_{ij} c_j a_{jk} c_k = \frac{1}{40}$ which are replaced by the single condition

$$\sum b_i c_i a_{ij} a_{jk} c_k + \sum b_i a_{ij} c_j a_{jk} c_k = \frac{1}{30} + \frac{1}{40}. \quad (6)$$

In a short 1995 paper [2] it is shown how to construct a method which satisfies all the equations for order 5 in the scalar theory but *not* for the vector theory.

A new method satisfying the same requirements is given by the tableau

0						
$\frac{1}{10}$	$\frac{1}{10}$					
$\frac{1}{5}$	$\frac{11}{260}$	$\frac{41}{260}$				
$\frac{3}{10}$	$-\frac{3}{170}$	$\frac{63}{340}$	$\frac{9}{68}$			
$\frac{3}{4}$	$\frac{13035}{544}$	$-\frac{75447}{1088}$	$\frac{9009}{136}$	$-\frac{1287}{64}$		
1	$-\frac{165709}{918}$	$\frac{733747}{1326}$	$-\frac{638689}{1122}$	$\frac{32053}{162}$	$-\frac{5320}{11583}$	
	$\frac{5}{54}$	0	0	$\frac{250}{567}$	$\frac{32}{81}$	$\frac{1}{14}$

For this method, all order 5 conditions are satisfied except those appearing in (6), which are replaced by

$$\begin{aligned} \sum b_i c_i a_{ij} a_{jk} c_k &= \frac{1}{30} - \frac{127}{20800}, \\ \sum b_i a_{ij} c_j a_{jk} c_k &= \frac{1}{40} + \frac{127}{20800}, \end{aligned}$$

so that (6) is still satisfied.

A suitable single differential equation to test the order of convergence of this method, together with a closely related autonomous system, are

$$\begin{aligned} \frac{dy}{dx} &= \frac{y-x}{y+x}, & y(0) &= 1, & (7) \\ \frac{d}{dt} \begin{bmatrix} y \\ x \end{bmatrix} &= \frac{1}{\sqrt{x^2+y^2}} \begin{bmatrix} y-x \\ y+x \end{bmatrix}, & \begin{bmatrix} y(1) \\ x(1) \end{bmatrix} &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}. & (8) \end{aligned}$$

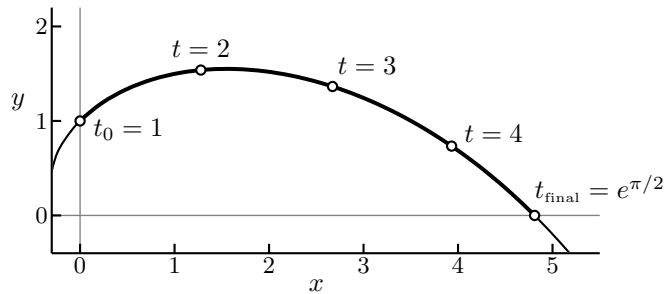


Figure 2: Solutions to (7) and (8)

Table 4: Numerical verification of order behaviour

	$n = 20$	$n = 40$	$n = 80$	$n = 160$	$n = 320$
$E_n^{(0)}/E_{2n}^{(0)}$	33.5068	33.0459	32.8983	32.5659	32.4374
$E_n^{(1)}/E_{2n}^{(1)}$	13.4076	14.8227	15.4446	15.7309	15.8740
$E_n^{(2)}/E_{2n}^{(2)}$	14.8548	15.5186	15.7833	15.8971	15.9678

The solution of (7), written in parametric coordinates, is

$$x = t \sin(\ln(t)), \quad (9)$$

$$y = t \cos(\ln(t)), \quad (10)$$

and this is exactly the solution to (8).

In numerical tests, we will approximate the solution when $t = x = \exp(\pi/2)$, $y = 0$. The solution on the interval $t \in [0, \exp(\pi/2)]$ is shown in Figure 2.

In two experiments, the same method was used to solve each of the problems and to find the absolute values of the computational errors in $y(\exp(\pi/2))$, found by solving (1) for a range of values of n , the number of equally spaced steps.

Denote this quantity by $E_n^{(0)}$. Similarly, using the formulation (2), quantities $E_n^{(1)}$ and $E_n^{(2)}$ are computed as the absolute values of the errors for the two components $y(\exp(\pi/2))$ and $x(\exp(\pi/2))$ respectively.

Our purpose is to assess the rate of convergence as n increases and it is appropriate to calculate $E_n^{(i)}/E_{2n}^{(i)}$ for $i = 0, 1, 2$. If the orders are as claimed in this paper, the values of these ratios for $i = 0$ should become close to 32 as n increases and for $i = 1$ and $i = 2$, they should approximate 16.

The actual values are given in Table 4 for values of n from 20 up to 320, in powers of 2, and the claim concerning orders is supported by these results.

Finally, a discussion of interesting Runge–Kutta methods, would not be complete without a mention of the 17 stage tenth order method constructed in 1978 by E. Hairer [7]. A recurring theme in the construction of high order methods is the use of simplifying assumptions. This method makes use of such assumptions adapted and combined in an ingenious way to form the ideal structure on which to build this method. It is unknown how many stages are required for

order eight and it may well be the case that the Hairer method is optimal in this respect but no proof is known and there seems to be little prospect of finding a similar method with fewer stages.

Acknowledgements

I owe a debt of nearly forty years to Ernst Hairer and Gerhard Wanner for their many important contributions to numerical analysis and especially for their formulation of the theory of B-series. I am also especially grateful to Tina Ming-Hua Chan for her collaborations with me and for her own original work, and in particular her introduction in [5] of the function Λ , which I have used in Section 4. The present paper, especially, owes a great deal to Tina's advice and assistance.

References

- [1] J. C. Butcher, An algebraic theory of integration methods, *Math. Comput.* **26** (1972), 79–106.
- [2] J. C. Butcher, On fifth order Runge–Kutta methods, *BIT* **35** (1995), 202–209.
- [3] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, Second Edition, J. Wiley, Chichester, 2008.
- [4] J. C. Butcher and T. M. H. Chan, A new approach to the algebraic structures for integration methods, *BIT* **42** (2002), 477–489.
- [5] T. M. H. Chan, *Algebraic Structures for the Analysis of Numerical Methods*, Ph.D Thesis, The University of Auckland, 1998.
- [6] A. Connes and D. Kreimer, Lessons from quantum field theory: Hopf algebras and spacetime geometries, *Lett. Math. Phys.* **48** (1999), 85–96.
- [7] E. Hairer, A Runge–Kutta method of order 10, *J. Inst. Math. Appl.* **21** (1978), 47–59.
- [8] E. Hairer and G. Wanner, On the Butcher group and general multi-value methods, *Computing* **13** (1974), 1–15.
- [9] A. Huťa, Une amélioration de la méthode de Runge–Kutta–Nyström pour la résolution numérique des équations différentielles du premier ordre, *Acta Fac. Nat. Univ. Comenian. Math.* **1** (1956), 201–224.