MATHEMATICS OF COMPUTATION
Volume 66, Number 219, July 1997, Pages 1089–1099
S 0025-5718(97)00873-9

# COMPOSITION CONSTANTS FOR RAISING THE ORDERS OF UNCONVENTIONAL SCHEMES FOR ORDINARY DIFFERENTIAL EQUATIONS

WILLIAM KAHAN AND REN-CANG LI

Abstract. Many models of physical and chemical processes give rise to ordinary differential equations with special structural properties that go unexploited by general-purpose software designed to solve numerically a wide range of differential equations. If those properties are to be exploited fully for the sake of better numerical stability, accuracy and/or speed, the differential equations may have to be solved by unconventional methods. This short paper is to publish composition constants obtained by the authors to increase efficiency of a family of mostly unconventional methods, called *reflexive*.

## 1. Introduction

Modeling many problems in physics, chemistry, and engineering gives rise to systems of ordinary differential equations. Typically these systems take the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}), \quad \text{with} \quad \mathbf{y}(0) = \mathbf{y}_0. \tag{1}$$

The initial vector $\mathbf{y}_0$ and the vector-valued function $\mathbf{f}(\,\cdot\,)$ are given, and the function is assumed as smooth as necessary. An interval $0 \le t \le T$ is usually specified for the scalar variable $t$, often identified with *Time*. The problem (1) is known as an *Initial Value Problem* (IVP), and as an *Autonomous Initial Value Problem* (AIVP) if $\mathbf{f}(t, \mathbf{y}) \equiv \mathbf{f}(\mathbf{y})$. Any given IVP (1) can be rewritten in a way that suppresses all explicit references to $t$; in other words, any IVP is equivalent to an AIVP. In what follows, we will consider AIVP

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}), \quad \text{with} \quad \mathbf{y}(0) = \mathbf{y}_0 \tag{2}$$

only, unless otherwise stated, in order to simplify the formulas that will arise.

In relatively few instances can analytical solutions be found for (2), and therefore the only option for most IVPs is a numerical solution. A typical program to solve the initial value problem is expected to generate a sequence of approximations

Received by the editor June 10, 1996.

1991 *Mathematics Subject Classification.* Primary 34A50, 65L05.

*Key words and phrases.* Ordinary differential equations, reflexive methods, composition schemes, palindromic schemes.

The first author was supported in part by the Office of Naval Research contract N00014-90-J-1372 and National Science Foundation contract ASC-9005933.

The second author was supported in part by a Householder Fellowship in Scientific Computing at Oak Ridge National Laboratory, supported by the Applied Mathematical Sciences Research Program, Office of Energy Research, United States Department of Energy contract DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation.

$\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N$ to $\mathbf{y}(t)$ at *Sample-Times* $t_0 = 0 < t_1 < t_2 < \cdots < t_N = T$. Numerical methods are classified into two major categories—*One-Step Methods* and *Multi-Step Methods* according to how they use past information. A method is classified as a *one-step method* if the computation of $\mathbf{y}_{n+1}$ involves only the approximation $\mathbf{y}_n$ to $\mathbf{y}(t_n)$, but not approximations at previous sample times; it is a *multi-step method* otherwise.

Many *conventional methods* (linear multi-step, Runge-Kutta methods) [1, 2, 7] are in use. To achieve generality, they have evolved into complicated programs thousands of lines long, and have become highly refined and relatively efficient solvers of a wide range of differential equations. Yet, because of their generality, conventional methods may do worse than what we called unconventional methods which exploit what may be known *a priori* about the initial value problem. In applications, differential systems often have some special structures and properties. Such structures and properties, if known and incorporated, may improve the efficiency of a numerical method greatly. Normally, constructing low order numerical formulas that preserve the structure is often much easier than going directly for higher order formulas that preserve the structure. These *ad hoc* formulas are often better than *conventional formulas* in some respect, but may be inaccurate because of their low orders of convergence. *Composition Schemes* are then particularly helpful to obtain higher order methods while retaining the properties of simple lower order updating formulas.

In this short paper, we will present composite constants that may help to increase the efficiency of certain numerical methods, called *reflexive*, for solving IVPs. While keeping this paper as short as possible, we try to give enough details for someone who'd like to try out our schemes on their particular applications. A more complete theory and history behind the schemes will be published in forthcoming papers.

## 2. Updating formulas and convergence

In principle, any one-step method for solving the initial value problem (2) yields an *updating formula* $\mathbf{Q}(\theta, \mathbf{g})$ which advances $\mathbf{g} \approx \mathbf{y}(\tau)$ to $\mathbf{Q}(\theta, \mathbf{g}) \approx \mathbf{y}(\tau + \theta)$.

Any updating formula appropriate to problem (2) is intended to be iterated $N$ times thus:

$$\mathbf{y}(T) \approx \mathbf{y}_N = \mathbf{Q}(\theta_{N-1}, \mathbf{Q}(\theta_{N-2}, \mathbf{Q}(\theta_{N-3}, ..., \mathbf{Q}(\theta_1, \mathbf{Q}(\theta_0, \mathbf{y}_0)) \cdots ))).$$

For this numerical solution to make sense, it is natural to ask that this $N$-fold composition of the updating formula yields a value converging to $\mathbf{y}(T)$ as $\max_n \theta_n \to 0$. It turns out that convergence depends on the *local error*

$$\mathbf{Q}(\theta, \mathbf{g}) - \Phi(\theta, \mathbf{g})$$

where $\Phi(\theta, \mathbf{y})$ is the solution operator defined by $\Phi(\theta, \mathbf{g}) \stackrel{\text{def}}{=} \mathbf{y}(\tau+\theta)$ for the problem

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}), \quad \text{with} \quad \mathbf{y}(\tau) = \mathbf{g}.$$

The updating formula $\mathbf{Q}(\theta, \mathbf{g})$ is called *consistent* if the local error is at most $o(\theta)$. It turns out that convergence is guaranteed if the updating formula $\mathbf{Q}(\theta, \mathbf{g})$ is consistent. A one-step method with updating formula $\mathbf{Q}(\theta, \mathbf{g})$ is of order $p$ if the local error satisfies

$$\mathbf{Q}(\theta, \mathbf{g}) - \Phi(\theta, \mathbf{g}) = O(\theta^{p+1}). \tag{3}$$

This means that the Taylor series of the numerical updating formula in powers of $\theta$ matches that of the true solution $\Phi(\theta, \mathbf{g})$ up to the term in $\theta^p$ for all $\mathbf{g}$. It is provable [1] that under (3) the *global error* behaves like

$$\mathbf{y}(T) - \mathbf{y}_N = O(\max_n \theta_n^p).$$

An updating formula $\mathbf{Q}(\theta, \mathbf{g})$ is *Reflexive* if

$$\mathbf{Q}(-\theta, \mathbf{Q}(\theta, \mathbf{g})) = \mathbf{g}.$$

(It has been called *Symmetric, Reversible*, and *Self-Adjoint* too but, as argued by Kahan [6], these terms are already overworked, so we prefer the word *reflexive*.) One example is *the Implicit Mid-point Rule:* $\mathbf{y}_{n+1} = \mathbf{y}_n + \theta_n \mathbf{f}\left(\frac{\mathbf{y}_n + \mathbf{y}_{n+1}}{2}\right)$. A consistent and reflexive formula has *at least second order convergence* [1, 2, 4, 8] and has other properties which allow efficient constructions of higher order approximations. One such construction composes $\mathbf{Q}(\theta_j, \cdot)$ with specially correlated step-sizes $\theta_j$; details will be given in the coming section.

In principle, a reflexive scheme can be obtained out of any conventional one-step numerical scheme by composing it with its *Reflection*. Various other unconventional ways [6, 8] to design reflexive schemes will be published in forthcoming papers.

## 3. Palindromic composition devised to increase a formula's order

Assume now $\mathbf{g} \approx \mathbf{y}(\tau)$. By composing the existing updating formula $\mathbf{Q}(\cdot, \cdot)$ to obtain higher order methods we mean, for example, that with appropriately chosen integer $m$ and scalar $\delta_j$'s

$$\mathbf{Q}(\delta_m \theta, \mathbf{Q}(\delta_{m-1}\theta, \mathbf{Q}(\cdots, \mathbf{Q}(\delta_1 \theta, \mathbf{g}) \cdots))) \tag{4}$$

approximates $\mathbf{y}(\tau + \theta)$ (much) more accurately than $\mathbf{Q}(\theta, \mathbf{g})$ does provided $\theta$ is small enough. We call (4) an *m-Stage Scheme.* Consistency implies that $\sum_{j=1}^{m} \delta_j = 1$. Because some of the $\delta_j$'s may be negative, the approximation (4) may be called a *Back-and-Forth* numerical scheme. Particularly interesting are the

$$\textit{Palindromic Compositions:} \quad \delta_i = \delta_{m-i+1} \quad \text{for } i = 1, 2, \cdots, m$$

when $\mathbf{Q}$ is reflexive. (This term was coined by Kahan in his lecture notes [6, 1993].) They preserve reflexiveness, and then lead to far simpler determining equations than do non-palindromic compositions. In what follows, we will be considering *Palindromic Compositions* (4) only.

An immediate question is "how shall we find these magic numbers $\delta_j$?" It turns out there are determining equations–so-called *order conditions*–that these $\delta_j$ must satisfy for (4) to be a certain (even) order approximations. Surprisingly the determining equations in this general context are equivalent to those that would be otherwise derived from special cases like for *separable Hamiltonian systems* [13], the implicit mid-point rule $\mathbf{Q}$ [10], and decompositions of exponential operators [12]. An explanation of such equivalence resides in *Lie Algebra Tools* [8]. In Yoshida [13], order conditions for orders up to 8 are given; while Suzuki [12] attempted to give order conditions for orders[1] up to 12, but his order 10 conditions are incorrect and so would his order 12 conditions. Since

[1] A palindromic scheme is always of even order of convergence. Nevertheless, Suzuki still assigned an odd order to a palindromic scheme. A scheme to which he assigned order $2k - 1$ would actually have order $2k$.

this paper is meant to be short, we will not go any further in this matter. The reader is referred to Li [8] and forthcoming papers for different ways of derivations. MAPLE codes that generate order conditions and more is available from NETLIB; see `http://www.netlib.org/ode/composition.txt`. The following table lists the numbers of determining equations for the approximations (4) to have a certain (even) order. (See also McLachlan [9].) By counting the numbers of equations and

| Order $2p$ | 2 | 4 | 6 | 8 | 10 | 12 |
|---|---|---|---|---|---|---|
| The #'s of det. eqs. | 1 | 2 | 4 | 8 | 16 | 34 |

free parameters in $\delta_j$'s, we arrive at the minimums of $m$ of an order $2p$ scheme.

| Order $2p$ | 2 | 4 | 6 | 8 | 10 | 12 |
|---|---|---|---|---|---|---|
| $m \geq$ | 1 | 3 | 7 | 15 | 31 | 67 |

The approximation (4) consists of $m$ moves; at the end of the $j$th move

$$\mathbf{Q}(\delta_j\theta, \mathbf{Q}(\cdots, \mathbf{Q}(\delta_1\theta, \mathbf{g})\cdots)) \approx \mathbf{y}(\tau + c_j\theta)$$

where $c_j \stackrel{\text{def}}{=} \sum_{i=1}^{j} \delta_i$. It is possible for a scheme to have some $c_j < 0$ or $c_j > 1$, which means some of the intermediate moves may jump "out of bounds", outside $[\tau, \tau+\theta]$. Such "out of bounds" moves are permissible in orbit calculations, but may be harmful in situations when true solutions $\mathbf{y}(t)$ pass too near singularities: "out of bounds" moves may hit or cross the singularities, and thus jeopardize computations. In our searching for high order schemes (4), efforts have been made to keep all $0 \leq c_j \leq 1$, among other things. We found that keeping all $0 \leq c_j \leq 1$ is possible only when the number of stages $m$ is bigger than its minimum required for achieving a particular order by at least 2, in which case the determining equations are underdetermined and thus present room for choices. Considering that increasing $m$ implies increasing work, we always keep $m$ as small as possible while having $0 \leq c_j \leq 1$. Two other quantities we have attempted to minimize (globally if we can or locally) are

$$\max_{1 \leq j \leq m} |\delta_j| \quad \text{and} \quad \sum_{j=1}^{m} |\delta_j|. \tag{5}$$

The first one is the largest intermediate step-size and the second is the overall distance travelled.

## 4. Palindromic schemes

Palindromic schemes of orders up to 10 have been constructed in Li [8]. Some of them have been known in some special context as we shall comment. For ease of future references, we adopt notation `sIodrJ?` to denote an *I-Stage Order J Scheme*. (Thus `s1odr2` is the reflexive updating formula itself.) Analytic solutions can be found for order 4 schemes.

1. `s3odr4`: $m = 3$ and $\delta_1 = \delta_3 = \frac{1}{2-\sqrt[3]{2}}$, $\delta_2 = -\frac{\sqrt[3]{2}}{2-\sqrt[3]{2}} < 0$ for which $c_1 = \delta_1 > 1$, $c_2 = -\frac{\sqrt[3]{2}-1}{2-\sqrt[3]{2}}$, and $c_3 = 1$. This is the scheme that has been discovered in integrating separable Hamiltonian systems by Yoshida [13], in composing the implicit mid-point rule by Sanz-Serna and Abia [10], and in its most general context by Kahan [6].

2. `s5odr4`: $m = 5$ and $\delta_1 = \delta_2 = \delta_4 = \delta_5 = \frac{1}{4-\sqrt[3]{4}}$, $\delta_3 = -\frac{\sqrt[3]{4}}{4-\sqrt[3]{4}}$. Suzuki [11] had this scheme for exponential approximations. It has been known to the first author for quite a while, but as the minimizer to both quantities in (5) it is due to [8].
3. `s5odr4a` and `s5odr4b`: $m = 5$ and $\delta_1 = \frac{3\pm\sqrt{3}}{6} = \delta_5$, $\delta_2 = \frac{3\mp\sqrt{3}}{6} = \delta_4$, $\delta_3 = -1$ for which $c_1 = \frac{3\pm\sqrt{3}}{6}$, $c_2 = 1$, $c_3 = 0$, $c_4 = \frac{3\mp\sqrt{3}}{6}$, $c_5 = 1$. They are interesting because they embedded an order 2 scheme in it, and thus may be used with cheap error estimators.

Analytic solutions are not available for schemes of orders 6 and higher. In the **Appendix**, numerical values with 20 correct decimal digits are given. These constants as well as codes that compute them are available from NETLIB; see `http://www.netlib.org/ode/composition.txt`.

## 5. Linear stability properties

In the past, instances have been reported on successful applications of composition schemes to Hamiltonian systems, but we cautioned the reader that composition schemes should be used with care. They may be instable, even though the **Q** is stable. In Li [8], a linear stability theory has been developed for the above mentioned palindromic schemes, assuming **Q** is the implicit mid-point rule and thus *A-Stable* [7]. But the linear stability regions for (4) will have holes in the left half-plane as long as there are negative $\delta_j$'s. This suggests that composition schemes may not be suitable to integrate stiff systems.

The reader is referred to Li [8] for the linear stability regions for all palindromic schemes in this paper.

## 6. An example

We present a simple numerical example to illustrate the usage of our schemes. The example also serves as a confirmation that these schemes do behave with the claimed order of convergence. Consider **Lorenz Attractor**

$$\begin{pmatrix} y_1' \\ y_2' \\ y_3' \end{pmatrix} = \begin{pmatrix} -\sigma(y_1 - y_2) \\ -y_1 y_3 + r y_1 - y_2 \\ y_1 y_2 - b y_3 \end{pmatrix}, \tag{6}$$

where $\sigma = 10$, $r = 28$, and $b = 8/3$. For illustration only, we take $y_1(0) = 10$, $y_2(0) = -20$ and $y_3(0) = 20$ initially, and are interested in integrating the system from $t = 0$ to $t = 1$.

A second order reflexive updating formula is obtained via a technique so-called *Symmetrical Splitting* [6, 8]. Let $y_i$'s be the approximations at $t = \tau$. Then the approximations $Y_i$ at $t = \tau + \theta$ are obtained via solving a linear system

$$\begin{pmatrix} Y_1 - y_1 \\ Y_2 - y_2 \\ Y_3 - y_3 \end{pmatrix} / \theta = \begin{pmatrix} -\sigma\left((y_1 + Y_1)/2 - (y_2 + Y_2)/2\right) \\ -(y_1 Y_3 + Y_1 y_3)/2 + r(y_1 + Y_1)/2 - (y_2 + Y_2)/2 \\ -(y_1 Y_2 + Y_1 y_2)/2 - b(y_3 + Y_3)/2 \end{pmatrix} \tag{7}$$

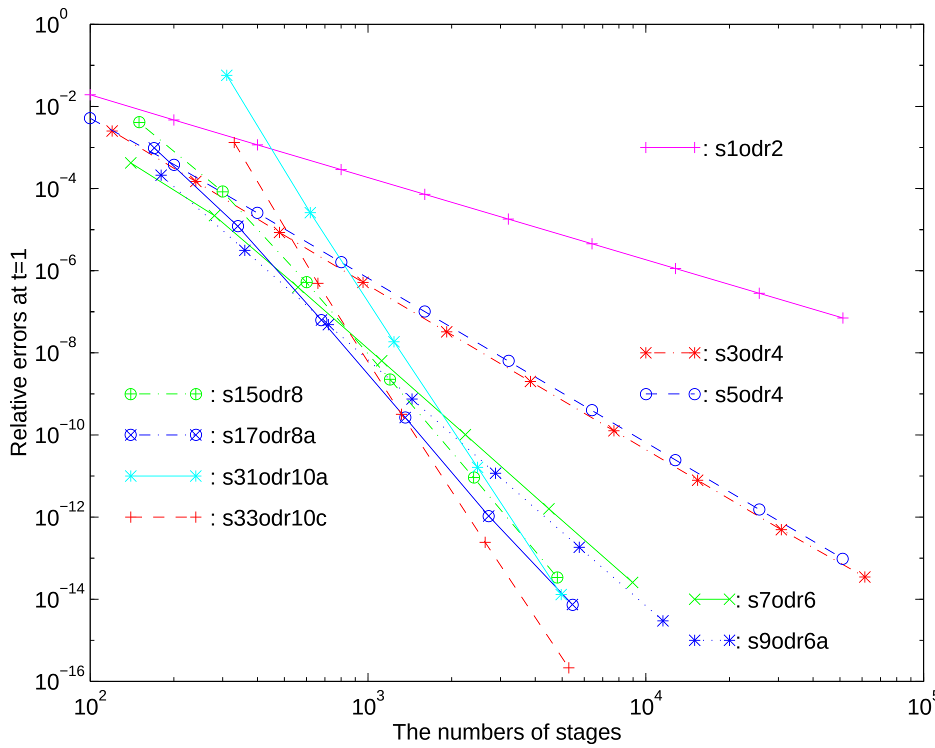


FIGURE 1. Relative errors of numerical solutions by palindromic schemes based on (8) ploted against costs (in the numbers of calls to (8)).

which is equivalent to

(8)

$$\left[ I - \frac{\theta}{2} \begin{pmatrix} -\sigma & \sigma & 0 \\ -y_3 + r & -1 & -y_1 \\ y_2 & y_1 & -b \end{pmatrix} \right] \begin{pmatrix} Y_1 - y_1 \\ Y_2 - y_2 \\ Y_3 - y_3 \end{pmatrix} = \theta \begin{pmatrix} -\sigma(y_1 - y_2) \\ -y_1 y_3 + r y_1 - y_2 \\ y_1 y_2 - b y_3 \end{pmatrix},$$

where $I$ is the $3 \times 3$ identity matrix. $\mathbf{Y} = \mathbf{Q}(\theta, \mathbf{y})$ obtained by solving (8) is reflexive since substitutions

$$\mathbf{y} \longleftarrow \mathbf{Y}, \quad , \mathbf{Y} \longleftarrow \mathbf{y}, \quad -\theta \longleftarrow \theta$$

leave (7) unchanged. It is worth mentioning that such $\mathbf{Q}(\theta, \mathbf{y})$ has an advantage over two conventional reflexive methods—*the trapezoidal rule* and *the implicit mid-point rule*—in that it requires solving no nonlinear systems but linear ones.

Once we have the $\mathbf{Q}(\theta, \mathbf{y})$, various palindromic schemes follow immediately. To keep this paper short, we choose to only present Figure 1 which plots the relative errors in numerical solutions at $t = 1$ against the numbers of calls to (8), where by relative errors we mean $|\alpha - \widetilde{\alpha}|/|\alpha|$ if $\widetilde{\alpha}$ is to approximate $\alpha$. Not all palindromic schemes in §4 and §A are included in the figure. This is because if we did, the figure would be a mess and not readable. But we point out that schemes not included behave similarly.

Two things need to be said about this example. First, the true solution to the system (6) is carefully computed using IBM's FORTRAN `REAL*16` and very small step-sizes. To 20 decimal digits, the true solution is

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}_{t=1} = \begin{pmatrix} \texttt{8.6356927098925060179D0} \\ \texttt{2.7986633879274570520D0} \\ \texttt{3.3360635089731421578D+1} \end{pmatrix}. \tag{9}$$

These digits are guaranteed correct by doing computations with different step-sizes and observing convergences.

Second, all computations are done in FORTRAN's `DOUBLE PRECISION`, and compensated summation technique is used. We briefly describe what we did with compensated summation technique. (For more discussion of compensated summation, see Kahan [5] and Higham [3].) The idea of the technique is to represent a number by two `double precision` floating point numbers such that the number is correctly represented to roughly 30 decimal digits. Take $y_1$ for an example. We represent $y_1$ by `(y1,yt1)`. As time advances from $\tau$ to $\tau + \theta$, $y_1$ is advanced to $Y_1$ and the difference $Y_1 - y_1$ (not $Y_1$ itself) is computed. Let the computed difference be `dy1`. Then $Y_1$ is represented as `(Y1, Yt1)` computed by

`Y1=(dy1+yt1)+y1` and `Yt1=((y1-Y1)+dy1)+yt1`.

Parentheses here must be fully respected. This technique turns out to be helpful in suppressing rounding errors sometimes. For example running `s9odr6a` for $\theta =$ `0.390625D-3` with/out compensated summation technique, we have the following relative errors in $y_i$'s at $t = 1$:

1. With compensated summation, `0.0000D+00,4.7604e-16,2.1299e-16`. For the $y_1$ component, it is due to pure luck.
2. Without compensated summation, `2.7152e-14, 1.6661e-14, 1.2353e-14`, less accurate by two decimal digits than with the technique.

## 7. Conclusions

We have presented constants for designing palindromic schemes of orders up to 10 from composing a reflexive (unconventional) scheme to possiblly increase its efficiency. Such schemes are very simple to implement, and may work much better than conventional schemes when they work. A simple example is included to illustrate the usage, as well as to verify the claimed orders of convergence of our schemes.

To keep this paper short, we left out discussions on important practical questions like *their stability properties*, *what orders are worth implementing?* The reader is referred to [8].

Finally, let's point out again that most of the material in this paper, including codes for the example, is available from NETLIB; see `http://www.netlib.org/ode/composition.txt`.

## Acknowledgement

We would like to thank Professor J. Demmel for his constructive suggestions.

## Appendix A. Palindromic schemes for orders 6 and higher

Among schemes that follow, `s7odr6` appeared in Yoshida [13], and `s15odr8` was also obtained by McLachlan [9] under different circumstances.

`s7odr6`

| | | | |
|---|---|---|---|
| $c_1$ | 0.78451361047755726382 | $\delta_1 = \delta_7$ | 0.78451361047755726382 |
| $c_2$ | 1.0200868238369153975 | $\delta_2 = \delta_6$ | 0.23557321335935813368 |
| $c_3$ | -0.15759316034195560944 | $\delta_3 = \delta_5$ | -1.1776799841788710069 |
| | | $\delta_4$ | 1.3151863206839112189 |

`s9odr6a`

| | | | |
|---|---|---|---|
| $c_1$ | 0.39216144400731413928 | $\delta_1 = \delta_9$ | 0.39216144400731413928 |
| $c_2$ | 0.72476058079667357788 | $\delta_2 = \delta_8$ | 0.33259913678935943860 |
| $c_3$ | 0.018514408239034218070 | $\delta_3 = \delta_7$ | -0.70624617255763935981 |
| $c_4$ | 0.10072800453258501830 | $\delta_4 = \delta_6$ | 0.082213596293550800230 |
| | | $\delta_5$ | 0.79854399093482996340 |

`s9odr6b`

| | | | |
|---|---|---|---|
| $c_1$ | 0.39103020330868478817 | $\delta_1 = \delta_9$ | 0.39103020330868478817 |
| $c_2$ | 0.72506749291982080566 | $\delta_2 = \delta_8$ | 0.33403728961113601749 |
| $c_3$ | 0.018840211732259462202 | $\delta_3 = \delta_7$ | -0.70622728118756134346 |
| $c_4$ | 0.10071776138031890797 | $\delta_4 = \delta_6$ | 0.081877549648059445768 |
| | | $\delta_5$ | 0.79856447723936218406 |

`s15odr8`

| | | | |
|---|---|---|---|
| $c_1$ | 0.74167036435061295345 | $\delta_1 = \delta_{15}$ | 0.74167036435061295345 |
| $c_2$ | 0.33256953855058135945 | $\delta_2 = \delta_{14}$ | -0.40910082580003159400 |
| $c_3$ | 0.52332424884681973940 | $\delta_3 = \delta_{13}$ | 0.19075471029623837995 |
| $c_4$ | -0.050538222269262527252 | $\delta_4 = \delta_{12}$ | -0.57386247111608226666 |
| $c_5$ | 0.24852595903439339659 | $\delta_5 = \delta_{11}$ | 0.29906418130365592384 |
| $c_6$ | 0.58315087727969158038 | $\delta_6 = \delta_{10}$ | 0.33462491824529818378 |
| $c_7$ | 0.89844396967645817701 | $\delta_7 = \delta_9$ | 0.31529309239676659663 |
| | | $\delta_8$ | -0.79688793935291635402 |

`s17odr8a`

| | | | |
|---|---|---|---|
| $c_1$ | 0.13020248308889008088 | $\delta_1 = \delta_{17}$ | 0.13020248308889008088 |
| $c_2$ | 0.69136546486399846544 | $\delta_2 = \delta_{16}$ | 0.56116298177510838456 |
| $c_3$ | 0.30189050221915117903 | $\delta_3 = \delta_{15}$ | -0.38947496264484728641 |
| $c_4$ | 0.46073240877430677993 | $\delta_4 = \delta_{14}$ | 0.15884190655515560090 |
| $c_5$ | 0.064828514641069202593 | $\delta_5 = \delta_{13}$ | -0.39590389413323757734 |
| $c_6$ | 0.24936815561938490968 | $\delta_6 = \delta_{12}$ | 0.18453964097831570709 |
| $c_7$ | 0.50774254330570695698 | $\delta_7 = \delta_{11}$ | 0.25837438768632204729 |
| $c_8$ | 0.80275426691501725585 | $\delta_8 = \delta_{10}$ | 0.29501172360931029887 |
| | | $\delta_9$ | -0.60550853383003451170 |

s17odr8b

| | | | |
|---|---|---|---|
| $c_1$ | 0.12713692773487857916 | $\delta_1 = \delta_{17}$ | 0.12713692773487857916 |
| $c_2$ | 0.68883946572368127888 | $\delta_2 = \delta_{16}$ | 0.56170253798880269972 |
| $c_3$ | 0.30630474577485109000 | $\delta_3 = \delta_{15}$ | -0.38253471994883018888 |
| $c_4$ | 0.46638080206949852119 | $\delta_4 = \delta_{14}$ | 0.16007605629464743119 |
| $c_5$ | 0.06456442774269155464 | $\delta_5 = \delta_{13}$ | -0.40181637432680696673 |
| $c_6$ | 0.25193114428497005171 | $\delta_6 = \delta_{12}$ | 0.18736671654227849724 |
| $c_7$ | 0.51263985349276245740 | $\delta_7 = \delta_{11}$ | 0.26070870920779240570 |
| $c_8$ | 0.80303724161792408129 | $\delta_8 = \delta_{10}$ | 0.29039738812516162389 |
| | | $\delta_9$ | -0.60607448323584816258 |

s31odr10a

| | | | |
|---|---|---|---|
| $c_1$ | -0.48159895600253002870 | $c_9$ | 0.13637459831059490870 |
| $c_2$ | -0.47796856284807043601 | $c_{10}$ | 0.32249749378157398757 |
| $c_3$ | 0.023834612739160966776 | $c_{11}$ | 0.55387077244595759390 |
| $c_4$ | 0.30681863898422351546 | $c_{12}$ | 0.031960406541771304852 |
| $c_5$ | 1.1138483179379457535 | $c_{13}$ | 0.78062154368676427278 |
| $c_6$ | 1.0877577373993535481 | $c_{14}$ | 0.84735805557736833031 |
| $c_7$ | 0.21489183593617283260 | $c_{15}$ | 0.043754811820660027146 |
| $c_8$ | -0.30884384468893298382 | | |
| $\delta_1 = \delta_{31}$ | -0.48159895600253002870 | $\delta_9 = \delta_{23}$ | 0.44521844299952789252 |
| $\delta_2 = \delta_{30}$ | 0.0036303931544595926879 | $\delta_{10} = \delta_{22}$ | 0.18612289547097907887 |
| $\delta_3 = \delta_{29}$ | 0.50180317558723140279 | $\delta_{11} = \delta_{21}$ | 0.23137327866438360633 |
| $\delta_4 = \delta_{28}$ | 0.28298402624506254868 | $\delta_{12} = \delta_{20}$ | -0.52191036590418628905 |
| $\delta_5 = \delta_{27}$ | 0.80702967895372223806 | $\delta_{13} = \delta_{19}$ | 0.74866113714499296793 |
| $\delta_6 = \delta_{26}$ | -0.026090580538592205447 | $\delta_{14} = \delta_{18}$ | 0.066736511890604057532 |
| $\delta_7 = \delta_{25}$ | -0.87286590146318071547 | $\delta_{15} = \delta_{17}$ | -0.80360324375670830316 |
| $\delta_8 = \delta_{24}$ | -0.52373568062510581643 | $\delta_{16}$ | 0.91249037635867994571 |

s31odr10b

| | | | |
|---|---|---|---|
| $c_1$ | 0.27338476926228452782 | $c_9$ | 0.61814916938393924433 |
| $c_2$ | 0.71926323428788736779 | $c_{10}$ | 0.13895907755995660185 |
| $c_3$ | 1.5514596627592504390 | $c_{11}$ | 0.30619982436039369094 |
| $c_4$ | 0.71749097720967101026 | $c_{12}$ | -0.56823168827336774213 |
| $c_5$ | 0.99640940777982295318 | $c_{13}$ | -1.0669665068095694000 |
| $c_6$ | 1.8867367882368482732 | $c_{14}$ | -0.47766114071982021148 |
| $c_7$ | 1.9434183030820939827 | $c_{15}$ | 0.35692823718900708627 |
| $c_8$ | 1.0860440949323051054 | | |
| $\delta_1 = \delta_{31}$ | 0.27338476926228452782 | $\delta_9 = \delta_{23}$ | -0.46789492554836586111 |
| $\delta_2 = \delta_{30}$ | 0.44587846502560283997 | $\delta_{10} = \delta_{22}$ | -0.47919009182398264249 |
| $\delta_3 = \delta_{29}$ | 0.83219642847136307126 | $\delta_{11} = \delta_{21}$ | 0.16724074680043708909 |
| $\delta_4 = \delta_{28}$ | -0.83396868554957942879 | $\delta_{12} = \delta_{20}$ | -0.87443151263376143307 |
| $\delta_5 = \delta_{27}$ | 0.27891843057015194293 | $\delta_{13} = \delta_{19}$ | -0.49873481853620165786 |
| $\delta_6 = \delta_{26}$ | 0.89032738045702532006 | $\delta_{14} = \delta_{18}$ | 0.58930536608974918851 |
| $\delta_7 = \delta_{25}$ | 0.056681514845245709418 | $\delta_{15} = \delta_{17}$ | 0.83458937790882729775 |
| $\delta_8 = \delta_{24}$ | -0.85737420814978887722 | $\delta_{16}$ | 0.28614352562198582747 |

s33odr10a

| | | | |
|---|---|---|---|
| $c_1$ | 0.070428877682658066880 | $c_9$ | 0.41100594684580454818 |
| $c_2$ | 0.94458539503619755729 | $c_{10}$ | 0.95487420737052878156 |
| $c_3$ | 1 | $c_{11}$ | 0.022351898086056138449 |
| $c_4$ | 0.93319952210120298840 | $c_{12}$ | 0.19195369692282078700 |
| $c_5$ | 0.30678643251320743247 | $c_{13}$ | 0.90803937270732642308 |
| $c_6$ | 0.54361264338849506120 | $c_{14}$ | 0.10787207023422068796 |
| $c_7$ | 0.12140200935679451910 | $c_{15}$ | 0.34565392315678839544 |
| $c_8$ | 0.36363143136720311159 | $c_{16}$ | 0.022350907648148961545 |
| $\delta_1 = \delta_{33}$ | 0.070428877682658066880 | $\delta_{10} = \delta_{24}$ | 0.54386826052472423338 |
| $\delta_2 = \delta_{32}$ | 0.87415651735353949041 | $\delta_{11} = \delta_{23}$ | -0.93252230928447264311 |
| $\delta_3 = \delta_{31}$ | 0.055414604963802442707 | $\delta_{12} = \delta_{22}$ | 0.16960179883676464855 |
| $\delta_4 = \delta_{30}$ | -0.066800477898797011598 | $\delta_{13} = \delta_{21}$ | 0.71608567578450563608 |
| $\delta_5 = \delta_{29}$ | -0.62641308958799555593 | $\delta_{14} = \delta_{20}$ | -0.80016730247310573512 |
| $\delta_6 = \delta_{28}$ | 0.23682621087528762872 | $\delta_{15} = \delta_{19}$ | 0.23778185292256770747 |
| $\delta_7 = \delta_{27}$ | -0.42221063403170054210 | $\delta_{16} = \delta_{18}$ | -0.32330301550863943389 |
| $\delta_8 = \delta_{26}$ | 0.24222942201040859249 | $\delta_{17}$ | 0.95529818470370207691 |
| $\delta_9 = \delta_{25}$ | 0.047374515478601436594 | | |

s33odr10b

| | | | |
|---|---|---|---|
| $c_1$ | 0.12282427644721572094 | $c_9$ | 0.45769021135686462033 |
| $c_2$ | 0.89927108535418012436 | $c_{10}$ | 0.90032429949679707981 |
| $c_3$ | 1.0480862308915230991 | $c_{11}$ | 0.080970924700860105168 |
| $c_4$ | 0.87569497135646242666 | $c_{12}$ | 0.21542566611838894562 |
| $c_5$ | 0.32823501353793778878 | $c_{13}$ | 0.85986805780855541100 |
| $c_6$ | 0.47336433681100706358 | $c_{14}$ | 0.14056656410653928542 |
| $c_7$ | 0.15771878527986245795 | $c_{15}$ | 0.35093558908002593152 |
| $c_8$ | 0.27858743617820117774 | $c_{16}$ | 0.081853639664320768579 |
| $\delta_1 = \delta_{33}$ | 0.12282427644721572094 | $\delta_{10} = \delta_{24}$ | 0.44263408813993245949 |
| $\delta_2 = \delta_{32}$ | 0.77644680890696440342 | $\delta_{11} = \delta_{23}$ | -0.81935337479593697464 |
| $\delta_3 = \delta_{31}$ | 0.14881514553734297479 | $\delta_{12} = \delta_{22}$ | 0.13445474141752884045 |
| $\delta_4 = \delta_{30}$ | -0.17239125953506067249 | $\delta_{13} = \delta_{21}$ | 0.64444239169016646538 |
| $\delta_5 = \delta_{29}$ | -0.54745995781852463787 | $\delta_{14} = \delta_{20}$ | -0.71930149370201612557 |
| $\delta_6 = \delta_{28}$ | 0.14512932327306927479 | $\delta_{15} = \delta_{19}$ | 0.21036902497348664610 |
| $\delta_7 = \delta_{27}$ | -0.31564555153114460562 | $\delta_{16} = \delta_{18}$ | -0.26908194941570516294 |
| $\delta_8 = \delta_{26}$ | 0.12086865089833871979 | $\delta_{17}$ | 0.83629272067135846284 |
| $\delta_9 = \delta_{25}$ | 0.17910277517866344258 | | |

s33odr10c

| | | | |
|---|---|---|---|
| $c_1$ | 0.12313526870982994083 | $c_9$ | 0.45728247090890761976 |
| $c_2$ | 0.89958508567920304603 | $c_{10}$ | 0.90046791756319334905 |
| $c_3$ | 1.0486399864748735022 | $c_{11}$ | 0.080978911880202504863 |
| $c_4$ | 0.87613237428093605795 | $c_{12}$ | 0.21480436926509833506 |
| $c_5$ | 0.32741996609293427854 | $c_{13}$ | 0.85989460450920438526 |
| $c_6$ | 0.47031762031135269954 | $c_{14}$ | 0.14053123280998377807 |
| $c_7$ | 0.15612568767148407956 | $c_{15}$ | 0.35004505094462027488 |
| $c_8$ | 0.28283512506709448979 | $c_{16}$ | 0.081763919538259755223 |
| $\delta_1 = \delta_{33}$ | 0.12313526870982994083 | $\delta_{10} = \delta_{24}$ | 0.44318544665428572929 |
| $\delta_2 = \delta_{32}$ | 0.77644981696937310520 | $\delta_{11} = \delta_{23}$ | -0.81948900568299084419 |
| $\delta_3 = \delta_{31}$ | 0.14905490079567045613 | $\delta_{12} = \delta_{22}$ | 0.13382545738489583020 |
| $\delta_4 = \delta_{30}$ | -0.17250761219393744420 | $\delta_{13} = \delta_{21}$ | 0.64509023524410605020 |
| $\delta_5 = \delta_{29}$ | -0.54871240818800177942 | $\delta_{14} = \delta_{20}$ | -0.71936337169922060719 |
| $\delta_6 = \delta_{28}$ | 0.14289765421841842100 | $\delta_{15} = \delta_{19}$ | -0.71936337169922060719 |
| $\delta_7 = \delta_{27}$ | -0.31419193263986861997 | $\delta_{16} = \delta_{18}$ | -0.26828113140636051966 |
| $\delta_8 = \delta_{26}$ | 0.12670943739561041022 | $\delta_{17}$ | 0.83647216092348048955 |
| $\delta_9 = \delta_{25}$ | 0.17444734584181312998 | | |

## References


[1] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I*, Springer-Verlag, New York, second ed., 1993. MR **94c:**65005

[2] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Springer-Verlag, New York, 1991. MR **92a:**65016

[3] N. J. Higham, *The accuracy of floating point summation*, SIAM J. Sci. Comput., 14 (1993), pp. 783–799. MR **94a:**65025

[4] W. Kahan, *Relaxation methods for solving systems of ordinary differential equations*, manuscript, CS Division, Department of EECS, University of California at Berkeley, Oct. 1977.

[5] ———, *Analysis and application of simply compensated summation*, work in progress, CS Division, Department of EECS, University of California at Berkeley, Oct. 1993.

[6] ———, *Unconventional numerical methods for trajectory calculations*, lectures notes, CS Division, Department of EECS, University of California at Berkeley, Oct. 1993.

[7] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, John Wiley & Sons, New York, 1991. MR **92i:**65114

[8] R.-C. Li, *Raising the Orders of Unconventional Schemes for Ordinary Differential Equations*, PhD thesis, Department of Mathematics, University of California at Berkeley, CA, 1995.

[9] R. I. McLachlan, *On the numerical integration of ordinary differential equations by symmetric composition methods*, SIAM Journal on Scientific Computing, 16 (1995), pp. 151–168. MR **95j:**65073

[10] J. M. Sanz-Serna and L. Abia, *Order conditions for canonical Runge-Kutta schemes*, SIAM Journal on Numerical Analysis, 28 (1991), pp. 1081–1096. MR **92e:**65103

[11] M. Suzuki, *General theory of fractal path integrals with applications to many-body theories and statistical physics*, Journal of Mathematical Physics, 32 (1991), pp. 400–407. MR **92k:**81096

[12] ———, *General theory of higher-order decomposition of exponential operators and symplectic integrators*, Physics Letters A, 165 (1992), pp. 387–395. MR **93b:**22011

[13] H. Yoshida, *Construction of higher order symplectic integrators*, Physics Letters A, 150 (1990), pp. 262–268. MR **91h:**70014



Computer Science Division and Department of Mathematics, University of California at Berkeley, Berkeley, California 94720

*E-mail address*: wkahan@cs.berkeley.edu

Mathematical Science Section, Oak Ridge National Laboratory, P.O. Box 2008, Bldg. 6012, Oak Ridge, Tennessee 37831-6367

*E-mail address*: na.rcli@na-net.ornl.gov