Cambridge International
Science Publishing
Cambridge CB1 6AZ
Great Britain

*Journal of Computational Methods in Sciences and Engineering*
vol. 2, no. 3, 2, pp. 1–3
ISSN 1472–7978

# Starting Methods for Two-Step Runge–Kutta Methods of Stage-Order 3 and Order 6

**J.H. Verner**[1,2]

Department of Mathematics, PIMS, Simon Fraser University, Burnaby, Canada, V5A 1S6

Received 18 February, 2003; accepted in revised form 4 April, 2004

*Abstract:* In [5], Jackiewicz and Tracogna proposed a general formulation of two step Runge–Kutta (TSRK) methods. Using formulas for two-step *pairs* of TSRK methods constructed in [6], Jackiewicz and Verner obtain results for order 8 pairs that fail to show this designated order. Hairer and Wanner [3] identify the problem by using B-series to formulate a complete set of order conditions for TSRK methods, and emphasize that special starting methods are necessary for the first step of implementation. They observe that for methods with stage order at least p-1, and design order p, starting methods of order at least p are sufficient. In this paper, the more general challenge to provide correct starting values for methods of low stage-order is met by showing how *perturbed* starting values should be selected for methods of order 6 and stage-order 3. The approach is sufficiently general that it may (and later will) be provided for such methods of higher orders. Evidence of the accompanying improvement in the implementation of TSRK methods illustrates that carefully designed starting methods are essential for efficient production codes based on methods of low stage-order.

*Keywords:*two-step Runge–Kutta methods, order reduction, starting methods, order conditions, implementation.

*Mathematics Subject Classification:* 65L05, 65L06, 65L20.

[1]jverner@pims.math.ca
[2]The work of this author was supported by the National Science and Engineering Council of Canada grant A8147.

## 1 Introduction

This article considers some explicit methods for approximating solutions to an initial value problem written as an autonomous system

$$\begin{cases} y'(x) = f(y(x)), \quad x \in [x_0, x_{end}], \\ y(x_0) = y_0, \end{cases} \tag{1}$$

where the derivative function $f : \mathbf{R}^n \to \mathbf{R}^n$ is assumed to be sufficiently continuous.

To exploit the advantages of accurate solutions obtained by Runge-Kutta methods for the numerical solution of (1), and the substantially smaller number of derivative evaluations per step required by linear multistep methods, various hybrid forms of these two types of method have appeared during the past four decades. Methods of a recent explicit family specified as two-step Runge–Kutta (TSRK) methods [5] are represented by the recursive formulas:

$$\begin{cases} Y_i^j = u_j y_{i-1} + (1-u_j) y_i + h \left( \sum_{k=1}^{s} a_{jk} f(Y_{i-1}^k) + \sum_{k=1}^{j-1} b_{jk} f(Y_i^k) \right), \\ \\ y_{i+1} = \theta y_{i-1} + (1-\theta) y_i + h \sum_{j=1}^{s} \left( v_j f(Y_{i-1}^j) + w_j f(Y_i^j) \right), \end{cases} \tag{2}$$

$j = 1, 2, \ldots, s$, $i = 1, 2, \ldots, N-1$ for a positive integer N, and a fixed stepsize $h = (x_{end} - x_0)/N$. In this paper, we restrict consideration to methods for which $u_j = \theta = 0$. For a vector $\mathbf{c}$ defined by $\mathbf{c} = (A + B)\mathbf{e}$ where A and B are the matrices of coefficients in (2) and $\mathbf{e}$ is a vector of units defined by $e_i = 1$, $i = 1, .., s$, the coefficients may be conveniently displayed in the form:

$$\begin{array}{c|c|c} \mathbf{c} & A & B \\ \hline & v^T & w^T \end{array} = \begin{array}{c|cccc|cccc} c_1 & a_{11} & a_{12} & \ldots & a_{1s} & & & & \\ c_2 & a_{21} & a_{22} & \ldots & a_{2s} & b_{21} & & & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \ldots & a_{ss} & b_{s1} & \ldots & b_{s,s-1} & \\ \hline & v_1 & v_2 & \ldots & v_s & w_1 & w_2 & \ldots & w_s \end{array} \tag{3}$$

Table 1: A TSRK as a Butcher tableau

To analyze these methods, we interpret $\{Y_i^j,\ j = 1, \ldots, s\}$ as approximations of local stage order $q \le p$ to $y(x_i + c_j h)$ and $y_i$ as an approximation of order $p$ to $y(x_i)$: this means that $\{Y_i^j = y(x_i + c_j h) + O(h^{min(q+1,p)}),\ j = 1, \ldots, s\}$, and $y_i = y(x_i) + O(h^p)$ respectively as $h \to 0$. In particular, the choice of $\mathbf{c}$ implies that each method has stage-order at least 1. While several modifications [6, 9] study implementation of these formulas with variable stepsizes in order to control error growth, this study focuses on a correct implementation of TSRK methods using fixed stepsizes in order to illustrate how leading terms of the error are propagated, and how they may be controlled to achieve the design order. Even so, these results will influence the implementation of corresponding TSRK pairs which use variable stepsizes, and further research in this direction may lead to improved algorithms.

As in [5, 6], the methods considered are explicit, and in particular are of type 1, so that $B$ is a lower triangular matrix. We assume the general form of order conditions derived by Jackiewicz and Tracogna [5] based on the approach of Albrecht [1]. This formulation uses the matrices A and B, the diagonal matrix C defined so that $\mathbf{c}$=C$\mathbf{e}$, and the identity matrix I. These order equations were solved directly by Jackiewicz and Verner [6] to obtain parametric families of TSRK pairs of orders (3,4), (5,6), and (7,8) requiring 3, 4 and 5 stages respectively. While the derivation of starting methods is confined here to those appropriate for selected four-stage TSRK formulas of order 6, the same approach with suitable modifications may be applied for corresponding improvements in TSRK methods of other orders.

## 2 Some illuminating experiments

Error estimating pairs of TSRK formulas of orders 7,8 constructed in [6] satisfied the order conditions, but on application to several problems, appeared to achieve only order 6. Because the methods were implemented using variable steps, the cause of this order reduction was not obvious. Philip Sharp suggested two tests using fixed order implementations that might help to resolve the dilemma. To illustrate the source of unexpected error, some known and new methods of order 6 are applied to three representative problems taken from the DETEST set [4].

1. A1: linear problem
$$y' = -y, \quad y(0) = 1$$
with solution $y(t) = e^{-t}$

2. A4: Logistic curve
$$y' = y(20 - y)/80 \quad y(0) = 1$$
with solution
$$y(t) = 20/(1 + 19e^{-t/4})$$

3. D1: Orbit problem
$$\begin{aligned} y_1'(t) &= y_3(t), & y_1(0) &= 0 \\ y_2'(t) &= y_4(t), & y_2(0) &= 0 \\ y_3'(t) &= -\frac{y_1(t)}{[y_1(t)^2 + y_2(t)^2]^{3/2}}, & y_3(0) &= 0 \\ y_4'(t) &= -\frac{y_2(t)}{[y_1(t)^2 + y_2(t)^2]^{3/2}}, & y_4(0) &= 0 \end{aligned}$$

To calibrate results from TSRK methods, each test was initially applied using an efficient eight-stage Runge–Kutta method (PDRK6) of order 6 obtained by Prince and Dormand [7]. In the first test, starting with initial values for the three initial value problems, approximations were obtained by this method with stepsizes $2^{-n}/5$, $n = 0, 1, \ldots, 8$, on the interval [0,20] (†[0,2$\pi$] on problem D1). Using sufficiently accurate values of the true solution, global errors (ge) were computed and used to estimate the achieved global order $p$, assuming that the error had the form $Kh^p$.

| | Linear Problem A1 | | Logistic Problem A4 | | Two Body Problem D1† | |
|---|---|---|---|---|---|---|
| h | ge | $p$ | ge | $p$ | ge | $p$ |
| 0.2 | $0.15636(10^{-16})$ | | $0.54090(10^{-12})$ | | $0.12150(10^{-8})$ | |
| 0.2/2 | $0.11175(10^{-18})$ | 7.12 | $0.85530(10^{-14})$ | 5.98 | $0.19598(10^{-10})$ | 5.97 |
| 0.2/4 | $0.83506(10^{-21})$ | 7.06 | $0.13447(10^{-15})$ | 5.99 | $0.31091(10^{-12})$ | 5.97 |
| 0.2/8 | $0.63805(10^{-23})$ | 7.03 | $0.21077(10^{-17})$ | 5.99 | $0.48941(10^{-14})$ | 5.98 |
| 0.2/16 | $0.49296(10^{-25})$ | 7.01 | $0.32985(10^{-19})$ | 5.99 | $0.76751(10^{-16})$ | 5.99 |
| 0.2/32 | $0.38299(10^{-27})$ | 7.00 | $0.51580(10^{-21})$ | 5.99 | $0.12014(10^{-17})$ | 5.99 |
| 0.2/64 | $0.29838(10^{-29})$ | 7.00 | $0.80626(10^{-23})$ | 5.99 | $0.18789(10^{-19})$ | 5.99 |
| 0.2/128 | $0.23279(10^{-31})$ | 7.00 | $0.12600(10^{-24})$ | 5.99 | $0.29372(10^{-21})$ | 5.99 |
| 0.2/256 | $0.18176(10^{-33})$ | 7.00 | $0.19732(10^{-26})$ | 5.99 | $0.45905(10^{-23})$ | 5.99 |

Table 2a. Order estimates using a fixed stepsize implementation of PDRK6 which has stage-order 1 and order 6.

In the second test, starting with the same initial values for problems A1 and A4, formal power series solutions in step length $h$ for the first three steps were obtained using the MAPLE computing environment.

| Taylor series of Error for Linear Problem | |
|---|---|
| RK step 1: | .0000248 $h^8$ + O($h^9$) |
| RK step 2: | .0000496 $h^8$ + O($h^9$) |
| RK step 3: | .0000744 $h^8$ + O($h^9$) |
| **Taylor series of Error for Logistic Problem** | |
| RK step 1: | - .146 $10^{-9}$ $h^7$ + O($h^8$) |
| RK step 2: | - .292 $10^{-9}$ $h^7$ + O($h^8$) |
| RK step 3: | - .438 $10^{-9}$ $h^7$ + O($h^8$) |

Table 2b: Leading error terms in three steps of PDRK6

Table 2a shows that PDRK6 achieves order 6 for A4 and D1, and order 7 for A1 (this method was designed to achieve order 7 for all linear, constant coefficient problems). Table 2b supports these results as leading error terms are of $O(h^7)$ for A4, and of $O(h^8)$ for A1, and both grow linearly.

For corresponding results from a TSRK method, we studied several stage-order 3 and order 6 methods derived using the approach in [6]. As the achieved order was clearly identified by TSRK methods using a small positive value of $c_1$, we selected the following method with $c_1 = 0.001$ for illustration. Additional values were required for starting, and the first test was applied using solution and derivative values over [0,h] obtained using RK4 over ten steps of length $h/10$. For the second test, exact solution and derivative starting values were obtained for A1 and A4, and used with this method.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\frac{1}{1000}$ | $-0.0110$ | 0.0437 | $-0.0454$ | 0.0137 | | | | |
| $\frac{23}{60}$ | $-0.0875$ | 0.6142 | $-1.0404$ | 0.2470 | $\frac{13}{20}$ | | | |
| $\frac{13}{20}$ | $-0.2542$ | 1.3786 | $-1.9025$ | 0.4353 | $\frac{5}{8}$ | 0.3677 | | |
| 1 | $-0.5849$ | 2.8724 | $-3.3735$ | $\frac{4}{5}$ | $\frac{1}{7}$ | $\frac{8}{9}$ | 0.2543 | |
| | $-0.0861$ | 0.7680 | $-1.6818$ | $\frac{13}{50}$ | 1.8380 | $-1.3422$ | 1.2358 | $\frac{1}{120}$ |

Table 3: Approximate coefficients for TSRK6

The results reported in Table 4a display less accuracy than for DPRK6, and only order 5 for the three problems shown; similar tests on other problems show similar results, although for a quadrature problem in which there is no dependence on errors in the stage values, the achieved order is 6. The results of Table 4b support this: even so, it is quite remarkable that the leading error terms which have $O(h^5)$ do not accumulate, but rather have an identical constant multiple from step to step, so that the global error is also a multiple of $h^5$. To understand why this occurs, we observe, that the propagated approximations $y_i$ have an error which is a multiple of $h^6$, and each new local error arises from a weighted sum of derivative values with errors of $O(h^5)$ added to this value. The coefficients of this weighted sum are sufficient that the error generated has exactly the same leading term as the corresponding error of the previous step. From the analysis, it turns out that this is to be expected, since the underlying design of a TSRK method is that in each step exactly the same error components in the stages are propagated from step to step in order that they may be annihilated by the final weight summation which yields the approximation to the solution at the grid points at the end of each step.

| h | Linear Problem A1 | | Logistic Problem A4 | | Two Body Problem D1† | |
|---|---|---|---|---|---|---|
| | ge | $p$ | ge | $p$ | ge | $p$ |
| 0.1 | $0.50710(10^{-15})$ | | $0.15090(10^{-10})$ | | $0.65613(10^{-9})$ | |
| 0.1/2 | $0.65912(10^{-17})$ | 6.18 | $0.46733(10^{-12})$ | 5.01 | $0.17914(10^{-10})$ | 5.19 |
| 0.1/4 | $0.76526(10^{-19})$ | 6.50 | $0.14513(10^{-13})$ | 5.00 | $0.50805(10^{-12})$ | 5.13 |
| 0.1/8 | $0.48130(10^{-21})$ | 7.31 | $0.45191(10^{-15})$ | 5.00 | $0.15132(10^{-13})$ | 5.06 |
| 0.1/16 | $0.13949(10^{-22})$ | 5.10 | $0.14095(10^{-16})$ | 5.00 | $0.46277(10^{-15})$ | 5.03 |
| 0.1/32 | $0.88229(10^{-24})$ | 3.98 | $0.44005(10^{-18})$ | 5.00 | $0.14321(10^{-16})$ | 5.01 |
| 0.1/64 | $0.34495(10^{-25})$ | 4.67 | $0.13745(10^{-19})$ | 5.00 | $0.44550(10^{-18})$ | 5.00 |
| 0.1/128 | $0.11858(10^{-26})$ | 4.86 | $0.42941(10^{-21})$ | 5.00 | $0.13891(10^{-19})$ | 5.00 |
| 0.1/256 | $0.38736(10^{-28})$ | 4.93 | $0.13417(10^{-22})$ | 5.00 | $0.43364(10^{-21})$ | 5.00 |

Table 4a. Order estimates using a fixed stepsize implementation of TSRK6 of order 6 with $c_1 = 0.001$ using 10 steps of RK4 for starting values.

Taylor series of Error for Linear Problem

| | |
|---|---|
| Starting step: | 0.00000 |
| TSRK step 2: | .00215 $h^5$ + .00393 $h^6$ - .00423 $h^7$ + $O(h^8)$ |
| TSRK step 3: | .00215 $h^5$ + .00465 $h^6$ - .0208 $h^7$ + $O(h^8)$ |

Taylor series of Error for Logistic Problem

| | |
|---|---|
| Starting step: | 0.0000 |
| TSRK step 2: | -.696 $10^{-6}$ $h^5$ + .367 $10^{-7}$ $h^6$ - .153 $10^{-6}$ $h^7$ + $O(h^8)$ |
| TSRK step 3: | -.696 $10^{-6}$ $h^5$ + .886 $10^{-7}$ $h^6$ - .395 $10^{-6}$ $h^7$ + $O(h^8)$ |

Table 4b: Leading error terms in two steps of TSRK6 with exact starting values

## 3 Order Conditions for a Starting Procedure

To motivate the derivation of a suitable starting procedure, we begin by outlining the approach taken by Jackiewicz and Tracogna [5] to derive order conditions for a TSRK method. Initially, the local truncation error of the propagated solution at $x_{i+1}$

$$\widehat{d}_{i+1} = y(x_{i+1}) - [y(x_i) + \ h \sum_{j=1}^{s} (v_j \, f(y(x_{i-1} + c_j h)) + w_j \, f(y(x_i + c_j h)))], \tag{4}$$

is expanded as a Taylor series with coefficients given as quadrature expressions

$$Q^{[k]} = \frac{1}{(k-1)!} \left( v^T (C - I)^{k-1} + w^T C^{k-1} - \frac{1}{k} \right) \mathbf{e}, \quad k = 1, 2, \ldots, p. \tag{5}$$

Similarly, a Taylor series expansion of the local truncation errors for the stage values $y(x_i + \mathbf{c}h) = [y(x_i + c_1 h), \ldots, y(x_i + c_s h)]^T$ defined as

$$d_{i+1} = y(x_i + \mathbf{c}h) - [y(x_i) + \ h \, (A \, f(y(x_{i-1} + \mathbf{c}h)) + B \, f(y(x_i + \mathbf{c}h)))] \, , \tag{6}$$

has coefficients given by the (vector) subquadrature expressions

$$\widetilde{C}^{[k]} = \frac{1}{(k-1)!} \left( A(C - I)^{k-1} + BC^{k-1} - \frac{C^k}{k} \right) \mathbf{e}, \quad k = 1, .., p. \tag{7}$$

For approximate stage values $Y_i = [Y_i^1, \ldots, Y_i^s]^T$ defined by (2), the errors defined by $q_{i+1} = y(x_i + \mathbf{c}h) - Y_i$, $i = 1, .., s$, must satisfy

$$q_{i+1} = hAt_i + hBt_{i+1} + hd_{i+1} + O(h^p). \tag{8}$$

Also, errors in derivative evaluations of these stages, $t_{i+1} = f(y(x_i + \mathbf{c}h)) - f(Y_i)$, must satisfy

$$v^T t_i + w^T t_{i+1} = O(h^p), \qquad h \to 0. \tag{9}$$

Then formal Taylor series in powers of h are obtained for each of $q_{i+1}$ and $t_{i+1}$. For coefficients of these formal series, (8) is used to obtain conditions which lead with further intricate manipulations of (9) to formal expressions of the order conditions. This development is provided separately for a single differential equation and for systems in [5].

In particular, the 37 order conditions which are necessary for a TSRK to have order 6 are derived and tabulated from these formal order conditions. The execution of one step of such a method uses solution values of order $p$ at two points, $x_i$ and $x_{i+1}$, and stage values of order $q$ ($\leq p$) at offstep points, to generate both solution and stage values one step later. Hence, solution values at $x_1$ and stage values for the first step must be selected by some procedure different from the TSRK formula. If starting stage values were chosen to be accurate to the design order, in particular that $q_1 \equiv y(x_0 + \mathbf{c}h) - Y_0 = O(h^p)$, then a careful study of the derivation in [5] would show that equation (8) fails to be valid in the first step if $q < p - 1$. Because $q_i = O(h^{q+1})$, when $q \leq p - 1$ for each subsequent step of the propagation, errors $q_1$ in the stage values of the first step must have a corresponding perturbation in terms of $O(h^{q+1})$ up to $O(h^{p-1})$. That is, we need solution and stage values for the initial step that are *perturbed* in exactly the same way that corresponding solution and stage values in subsequent steps deviate from their asymptotically correct Taylor expansions. In [2], Butcher developed such a strategy to implement so-called Runge–Kutta methods of effective order.

Accordingly, we extend the analysis in Jackiewicz and Tracogna [5] to determine how starting solution and derivative values may be selected in order to be consistent with the order conditions. The pairs derived in [6] yield coefficients which satisfy the order conditions necessary to achieve the correct order in each propagated step of a TSRK method. What is needed is a procedure for calculating each of the *starting* values $y_1$ and $Y_0$. The solution value $y_1$ at $x_1$ must be $O(h^{p+1})$, and this may be obtained by a standard Runge–Kutta method of order p. However, to propagate (2) correctly, the starting stage values of $Y_0$ must be selected so that

$$q_1 = hAt_0 + hBt_1 + hd_1 + O(h^p) \tag{10}$$

which may be rewritten as

$$\begin{aligned} Y_0 = y(x_0 + \mathbf{c}h) + \sum_{k=1}^{p-1} h^k \widetilde{C}^{[k]} y^{(k)}(x_0) \\ - hA[f(y(x_{-1} + \mathbf{c}h) - f(Y_{-1})] - hB[f(y(x_0 + \mathbf{c}h) - f(Y_0)] + O(h^p). \end{aligned} \tag{11}$$

For a TSRK method of order $p = 6$, and stage order $q = 3$ so that $\widetilde{C}^{[k]} = 0$, $0 \leq k \leq 3$, only coefficients of $h^4$ and $h^5$ in the first sum need to be considered. Hence, we need to find starting values $Y_0$ which satisfy

$$\begin{aligned} Y_0 = y(x_0 + \mathbf{c}h) + h^4 \widetilde{C}^{[4]} y^{(4)}(x_0) + h^5 \widetilde{C}^{[5]} y^{(5)}(x_0) \\ + hA[f(Y_{-1}) - f(y(x_{-1} + \mathbf{c}h))] + hB[f(Y_0) - f(y(x_0 + \mathbf{c}h))] + O(h^6). \end{aligned} \tag{12}$$

This formula and the analog for the expected value of $Y_{-1}$ can be expanded to give

$$Y_i = y(x_i + \mathbf{c}h) + h^4 \widetilde{C}^{[4]} y^{(4)}(x_i) + O(h^5), \quad i = -1, 0. \tag{13}$$

On recursive substitution of this expansion into the right side of (12), and then partial expansion about $y(x_0 + \mathbf{c}h)$ and $x_0$ leads to

$$\begin{aligned} Y_0 = {} & y(x_0 + \mathbf{c}h) + h^4 \widetilde{C}^{[4]} y^{(4)}(x_0) + h^5 \widetilde{C}^{[5]} y^{(5)}(x_0) \\ & + h(A + B) \frac{\partial f}{\partial y}(y(x_0 + \mathbf{c}h))[h^4 \widetilde{C}^{[4]} y^{(4)}(x_0)] + O(h^6). \end{aligned} \tag{14}$$

Finally, a series expansion in powers of h gives

$$\begin{aligned} Y_0 = {} & \sum_{k=0}^{5} \frac{h^k C^k \mathbf{e}}{k!} y^{(k)}(x_0) + h^4 \widetilde{C}^{[4]} y^{(4)}(x_0) + h^5 \widetilde{C}^{[5]} y^{(5)}(x_0) \\ & + h(A + B) \frac{\partial f}{\partial y}(y(x_0)) h^4 \widetilde{C}^{[4]} y^{(4)}(x_0) + O(h^6) \\ = {} & y(x_0) + h\mathbf{c}y'(x_0) + \frac{h^2 C^2 \mathbf{e}}{2!} y''(x_0) \\ & + \frac{h^3 C^3 \mathbf{e}}{3!} y^{(3)}(x_0) + \frac{h^4 C^4}{4!} \big[\mathbf{e} + 4! C^{-4} \widetilde{C}^{[4]}\big] y^{(4)}(x_0) \\ & + \frac{h^5 C^5}{5!} \Big[(\mathbf{e} + 5! C^{-5} \widetilde{C}^{[5]}) y^{(5)}(x_0) + 5! C^{-5} (A + B) \widetilde{C}^{[4]}\Big] \frac{\partial f}{\partial y}(y(x_0)) y^{(4)}(x_0) \\ & + O(h^6). \end{aligned} \tag{15}$$

This final form motivates the selection of starting stage values $Y_0$. Each component of $Y_0$ is just an approximation to the solution at an internal point of the initial step. Hence, we may formulate the following result:

**Theorem:** Approximate solution values for an initial value problem obtained using a two-step Runge-Kutta method of stage order 3 and order 6 will achieve this design order if starting values $Y_0^j,\ j = 1, .., 4$ are selected so that they agree up to terms of $\mathrm{O(h^5)}$ in expansion (15).

We show now how to derive a set of special Runge–Kutta methods which are designed specifically to obtain correct starting values for any well-behaved problem. A different method is derived for computing each of the four starting values. The parameters of each method denoted by $\{\overline{\mathbf{b}}, \overline{A}, \overline{\mathbf{c}}\}$ are computed to satisfy the standard order conditions up to order 3, and in addition, sets of perturbed order conditions of orders 4 and 5 which arise from the corresponding terms in expansion (15) above. These conditions can be easily tabulated using the matrix formulation for the TSRK methods, and corresponding matrices for the starting methods.

For a convenient representation of the order conditions, we define (vectors)

$$\widehat{C}^{[k]} = k! C^{-k} \widetilde{C}^{[k]}, \quad k = 4, 5, \tag{16}$$

$$\widehat{ApBC}^{[4]} = 5! C^{-5} (A + B) \widetilde{C}^{[4]}. \tag{17}$$

Observe that $C_j^{[4]}$ and $\widehat{ApBC}_j^{[4]}$ denote the components of these vectors which define conditions corresponding to the starting value at node $c_j,\ j = 1, .., 4$. Note that a *different* starting method is required for each stage.

Some comments on these special order conditions may clarify their derivation obtained by expanding (15) in terms of elementary differentials. Since the starting values are only used in the derivative evaluations of subsequent solution components, only terms up to order 5 need be considered. In Table 5, only terms of orders 4 and 5 are modified from the corresponding standard order conditions. For each, multiplication by the standard derivative $y^{(k)}(x_0)$ determines that $\widehat{C}_j^{[k]}$, $k = 4, 5$ must be added. As well, we add $\widehat{ApBC}_j^{[4]}$ to coefficients of those terms of order 5 for which the elementary differentials include $\partial \mathrm{f}/\partial \mathrm{y}\{\mathrm{y}(\mathrm{x}_0)\}$ – i.e. precisely those terms for which the Runge–Kutta order conditions begin with $\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}$. The left side of condition XI requires the dot product of the weight vector $\overline{\mathbf{b}}$ with the term by term *square* of the vector $\overline{AC}\mathbf{e}$ – this componentwise square is denoted in XI by a dot.

| | | | |
|---|---|---|---|
| *I.* | $\overline{\mathbf{b}}^{\mathbf{t}}\mathbf{e} = 1$ | *IX.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{C}^4\mathbf{e} = \dfrac{1+\widehat{C}_j^{[5]}}{5}$ |
| *II.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{C}\mathbf{e} = \frac{1}{2}$ | *X.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{C}^2\overline{AC}\mathbf{e} = \dfrac{1+\widehat{C}_j^{[5]}}{10}$ |
| *III.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{C}^2\mathbf{e} = \frac{1}{3}$ | *XI.* | $\overline{\mathbf{b}}^{\mathbf{t}}(.\overline{AC}\mathbf{e})^2 = \dfrac{1+\widehat{C}_j^{[5]}}{20}$ |
| *IV.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{AC}\mathbf{e} = \frac{1}{6}$ | *XII.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{CAC}^2\mathbf{e} = \dfrac{1+\widehat{C}_j^{[5]}}{15}$ |
| *V.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{C}^3\mathbf{e} = \dfrac{1+\widehat{C}_j^{[4]}}{4}$ | *XIII.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{CA}^2\overline{C}\mathbf{e} = \dfrac{1+\widehat{C}_j^{[5]}}{30}$ |
| *VI.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{CAC}\mathbf{e} = \dfrac{1+\widehat{C}_j^{[4]}}{8}$ | *XIV.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{AC}^3\mathbf{e} = \dfrac{1+\widehat{C}_j^{[5]}+\widehat{ApBC}_j^{[4]}}{20}$ |
| *VII.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{AC}^2\mathbf{e} = \dfrac{1+\widehat{C}_j^{[4]}}{12}$ | *XV.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{ACAC}\mathbf{e} = \dfrac{1+\widehat{C}_j^{[5]}+\widehat{ApBC}_j^{[4]}}{40}$ |
| *VIII.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^2\overline{C}\mathbf{e} = \dfrac{1+\widehat{C}_j^{[4]}}{24}$ | *XVI.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^2\overline{C}^2\mathbf{e} = \dfrac{1+\widehat{C}_j^{[5]}+\widehat{ApBC}_j^{[4]}}{60}$ |
| | | *XVII.* | $\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^3\overline{C}\mathbf{e} = \dfrac{1+\widehat{C}_j^{[5]}+\widehat{ApBC}_j^{[4]}}{120}$ |

Table 5: Order conditions to order 5 for the starting method for node $c_j$

## 4 Solving the Order Conditions

To obtain components of $Y_0$, we derive a set of four Runge–Kutta methods: we require coefficients $\{\overline{\mathbf{b}}, \overline{A}, \overline{\mathbf{c}}\}$ for one starting method satisfying the (perturbed) order conditions of Table 5 for each of four stages. For each method, we elected to use seven stages, and a strategy for solving the order conditions is adapted from that for 5,6 pairs in [10]. In particular, we will construct methods with stage order 2. In addition, we compute $y_1$ using a corresponding Runge–Kutta method of stage order 2 and conventional order 6 also derived using the strategy developed by Verner [10]. We continue with a detailed description of the algorithm for computing coefficients of the "internal" starting Runge–Kutta methods. For each internal node $c_j$: choose $\{(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^k)_2 = 0,\ k = 0, 1, 2, 3\}$, and then solve the four sets of *linear* equations separately for k=0,1,2,3,

$$(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^k)\overline{C}^{\nu-1}\mathbf{e} = \frac{1+\widehat{C}_j^{[\nu+k]}+(\widehat{ApBC}^{[\nu+k-1]})_j}{\nu(\nu+1)..(\nu+k)}, \quad \nu = 1, .., 6-k, \tag{18}$$

to obtain the (vector) homogeneous polynomials $\overline{\mathbf{b}}^{\mathbf{t}}$, $\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}$, $\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^2$, $\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^3$ respectively. (Note: $\hat{C}^{[k]} = 0$, $k = 1, 2, 3$, and for convenience, we selected $\hat{C}_j^{[6]} = 0$, and $\widehat{ApBC}_j^{[5]} = 0$, although any other values may be selected without decreasing the order of the methods obtained.) Now it may be easily shown that coefficients chosen to yield these homogeneous polynomials together with the stage-order conditions $\overline{A}\mathbf{e} = \overline{C}\mathbf{e}$ and $\overline{AC}\mathbf{e} = \overline{C}^2\mathbf{e}/2$ satisfy all but conditions XII and XIII of Table 5. With these choices, conditions XII and XIII hold simultaneously if

$$\begin{aligned}(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^2)_5 \left[(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A})_6 \textstyle\sum_{j=3}^4 \overline{b}_j(\overline{c}_j - \overline{c}_7)\overline{a}_{j2} - \overline{b}_6(\overline{c}_6 - \overline{c}_7)\sum_{j=3}^4(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A})_j\overline{a}_{j2}\right] \\ = \textstyle\sum_{j=3}^4(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^2)_j\overline{a}_{j2}\left[(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A})_6\overline{b}_5(\overline{c}_5 - \overline{c}_7) - \overline{b}_6(\overline{c}_6 - \overline{c}_7)(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A})_5\right] ,\end{aligned} \tag{19}$$

and this condition may be satisfied by using it to compute $\overline{a}_{4,2}$. Moreover, each of order conditions XII and XIII must hold separately (or else neither would be valid). For a standard Runge–Kutta method of order 6, the order conditions up to order 5 are those of Table 5 with $\hat{C}_j^{[k]} = 0$. In this case, order condition XII may be written (see [10])

$$\begin{aligned}3\int_0^1 c(c-\overline{c}_3)(c-1)^2dc\int_0^1 c(c-\overline{c}_3)(c-\overline{c}_4)(c-1)^2dc \\ = 2\int_0^1 c(c-\overline{c}_3)(c-1)^3dc\int_0^1 c(c-\overline{c}_3)(c-\overline{c}_4)(c-1)dc\end{aligned}$$

which reduces to

$$\overline{c}_4 = \frac{\overline{c}_3}{2 - 10\overline{c}_3 + 15\overline{c}_3^2}\,. \tag{20}$$

In contrast, the corresponding condition to satisfy XII (or else XIII when (19) holds) for a TSRK method is a substantial constraint on nodes $\overline{c}_3, \overline{c}_4, \overline{c}_5$. Using MAPLE to manipulate order condition XII in this case, we find for the values of $\widetilde{C}_j^{[4]}$, $\widetilde{C}_j^{[5]}$, $((A+B)\widetilde{C}^{[4]})_j$ determined from the selected TSRK method, that values of the nodes $\overline{c}_3$, $\overline{c}_4$, $\overline{c}_5$, of an acceptable starting method for node $c_j$ must be selected to annihilate the polynomial in these parameters which appears in the Appendix.

This polynomial imposes a condition on the nodes which together with the previous conditions is *sufficient* to obtain a starting method for node $c_j$. It is linear in $\overline{c}_5$, but unstable in this parameter: small changes in either $\overline{c}_3$ or $\overline{c}_4$ give large changes in $\overline{c}_5$. Fortunately, it is a stable quadratic in $\overline{c}_4$, and suitable distinct values of $\overline{c}_2, \overline{c}_3, \overline{c}_5, \overline{c}_6$ in [0,1] will yield a value of $\overline{c}_4$ lying inside the interval [0,1].

In summary, to compute the coefficients of each starting method, we need to (i) compute $\widetilde{C}_j^{[4]}$, $\widetilde{C}_j^{[5]}$, $((A+B)\widetilde{C}^{[4]})_j$ from the selected TSRK method (ii) select the arbitrary nodes $\overline{c}_2, \overline{c}_3, \overline{c}_5, \overline{c}_6$ in [0,1] (iii) compute $\overline{c}_4$ to annihilate $p(\overline{c}_3, \overline{c}_4, \overline{c}_5)$. (If $\overline{c}_4$, does not lie in [0,1], other choices of the arbitrary nodes are required.) (iv) Now, choose $\overline{a}_{3,2}$, $\overline{a}_{4,2}$, $\overline{a}_{4,3}$ to satisfy the two stage order conditions and (19) (v) compute the homogeneous polynomials using (18), and (vi) use these with the back-substitution (see [8, p. 1175] for example):

$$\overline{a}_{8-r,j} = \frac{(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^r)_j - \sum_{i=j+1}^{7-r}(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^{r-1})_i\overline{a}_{ij}}{(\overline{\mathbf{b}}^{\mathbf{t}}\overline{A}^{r-1})_{8-r}}\,, \quad r = 1, 2, 3, \quad j = 7-r, ..., 1, \tag{21}$$

to find the remaining entries of $\overline{A}$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | | | | | | | |
| $\frac{1}{8}$ | 0.125 | | | | | | |
| $\frac{4}{7}$ | −0.7346 | 1.3061 | | | | | |
| 0.2856 | −0.2957 | 0.6527 | −0.7140 | | | | |
| $\frac{3}{7}$ | −0.5152 | 0.9795 | −0.7141 | 0.3573 | | | |
| $\frac{3}{4}$ | −1.0180 | 1.7144 | 0.4344 | 0.1267 | −0.5075 | | |
| 1 | −1.2528 | 2.2862 | −0.8162 | −0.8975 | 1.0056 | 0.6746 | |
| | $-3.130E14$ | 0 | $7.464E15$ | $3.842E15$ | $-8.772E15$ | $-2.473E15$ | $2.517E14$ |

Table 6: Approximate coefficients of the starting method for node $c_1 = 1/1000$

For choices of small positive $c_1$, the weights are large, *but* in our experiments the corresponding starting methods obtained more clearly indicate the expected improvement in order. In order to illustrate this dependence on the starting methods, we have selected a TSRK method with $c_1 = 1/1000$. For this TSRK method in Table 3, the first (of four) internal Runge–Kutta starting methods has the coefficients in Table 6. (The free parameters are displayed as rational fractions.) Starting methods for the remaining three (larger) nodes, can be computed using the given algorithm. For these, the weights in $\mathbf{b}$ are of moderate size. (We conclude by noting that because $C^{-k}$ occurs in computing $\hat{C}^{[k]}$, $c_1$ must be different than zero. We have not found an alternative derivation which will allow $c_1 = 0$, and because the weights become large as $c_1 \to 0$, this may not be possible for this approach to computing $\mathrm{Y}_0$.)

Accurate values of the TSRK coefficients in Table 3 may be computed using the algorithm in [6]. For each of the four starting methods, we selected the (same) arbitrary nodes $\overline{c}_2$, $\overline{c}_3$, $\overline{c}_5$, and $\overline{c}_6$, shown in Table 6, but the computed coefficients (shown for example, for the first starting method in Table 6 as decimal approximations) are different for each starting method.

| | Linear Problem A1 | | Logistic Problem A4 | | Two Body Problem D1† | |
|---|---|---|---|---|---|---|
| h | ge | $p$ | ge | $p$ | ge | $p$ |
| 0.1 | $0.57864(10^{-15})$ | | $0.14751(10^{-11})$ | | $0.12865(10^{-8})$ | |
| 0.1/2 | $0.83695(10^{-17})$ | 6.11 | $0.24493(10^{-13})$ | 5.91 | $0.18354(10^{-10})$ | 6.13 |
| 0.1/4 | $0.12576(10^{-18})$ | 6.05 | $0.39401(10^{-15})$ | 5.95 | $0.27323(10^{-12})$ | 6.06 |
| 0.1/8 | $0.19267(10^{-20})$ | 6.02 | $0.62449(10^{-17})$ | 5.97 | $0.41639(10^{-14})$ | 6.03 |
| 0.1/16 | $0.29809(10^{-22})$ | 6.01 | $0.98268(10^{-19})$ | 5.98 | $0.64240(10^{-16})$ | 6.01 |
| 0.1/32 | $0.46347(10^{-24})$ | 6.00 | $0.15408(10^{-20})$ | 5.99 | $0.99734(10^{-18})$ | 6.00 |
| 0.1/64 | $0.72239(10^{-26})$ | 6.00 | $0.24118(10^{-22})$ | 5.99 | $0.15533(10^{-19})$ | 6.00 |
| 0.1/128 | $0.11273(10^{-27})$ | 6.00 | $0.37717(10^{-24})$ | 5.99 | $0.24232(10^{-21})$ | 6.00 |
| 0.1/256 | $0.17604(10^{-29})$ | 6.00 | $0.58941(10^{-26})$ | 5.99 | $0.37834(10^{-23})$ | 6.00 |

Table 7a. Order estimates using a fixed stepsize implementation of a TSRK method of order 6 and stage-order 3 with $c_1 = 0.001$, with four 7-stage perturbed Runge–Kutta methods for $Y_0$, and an RK6 for $y_1$.

We applied this method to the same initial value problems and tests of §2. Despite the large weights of the first starting method shown in Table 6, the expected improvement is clearly documented in Tables 7a and 7b: we achieved the design order of the TSRK methods in both tests for these and for other test problems. Moreover, the numerical error has been reduced from that in Table 4a with the use of these starting methods, even though this improved procedure is still not quite as efficient as the near optimal PDRK6 used for comparison. (Observe that each step of TSRK6 uses four derivative evaluations per step, while each step of PDRK6 requires eight deriva-

tives per step; to equilibrate the relative amounts of computation, we doubled the stepsize for each application of PDRK6.)

| Taylor series of Error for Linear Problem | |
|---|---|
| Starting step | - .000677 $h^7$ + O($h^8$) |
| TSRK step 2: | -.00499 $h^7$ + O($h^8$) |
| TSRK step 3: | -.0170 $h^7$ + O($h^8$) |
| **Taylor series of Error for Logistic Problem** | |
| Starting step: | .213 $10^{-7}$ $h^7$ + O($h^8$) |
| TSRK step 2: | -.154 $10^{-6}$ $h^7$ + O($h^8$) |
| TSRK step 3: | -.384 $10^{-6}$ $h^7$ + O($h^8$) |

Table 7b: Leading error terms in three steps of TSRK6 with starting methods

## 5 Conclusions

This study illustrates that special starting methods identified by Hairer and Wanner [3] can be designed and derived to achieve full order when TSRK methods of low stage-order are applied to smooth problems. The method used for illustration is not better than a near optimal conventional Runge–Kutta method. However, more improvement both in the selection of arbitrary parameters and in the design of starting methods may be possible, and hence further investigation of these methods is warranted.

We have yet to understand if starting methods exist for TSRK methods with $c_1 = 0$, if there are good starting methods for all TSRK methods of order 6 (the weights of the starting method for $c_1$ near zero are very large), and if there are suitable starting methods for TSRK methods of order 8 (for which serious deterioration of the order occurred when used with an adaptive stepsize). A more serious challenge that remains is to determine how such starting methods may be designed to accommodate a change of stepsize.

## References

[1] P. Albrecht, A new theoretical approach to Runge–Kutta methods, *SIAM J. Numer. Anal.* **24**, 391-406(1987).

[2] J.C. Butcher, The effective order of Runge–Kutta methods, Conf. on the Numerical Solution of Differential Equations, Dundee, *Lecture Notes in Mathematics* **109**, Springer, Berlin, 133-139(1969).

[3] E. Hairer and G. Wanner, Order conditions for general two-step Runge–Kutta methods, *SIAM J. Numer. Anal.* **34**, 2087-2089(1997).

[4] T.E. Hull, W.H. Enright, B.M. Fellen and A.E. Sedgwick, Comparing numerical methods for ordinary differential equations, *SIAM J. Numer. Anal.* **9**, 603-637(1972).

[5] Z. Jackiewicz and S. Tracogna, A general class of two-step Runge–Kutta methods for ordinary differential equations, *SIAM J. Numer. Anal.* **32**, 1390-1427(1995).

[6] Z. Jackiewicz and J.H. Verner, Derivation and implementation of two-step Runge–Kutta pairs, *Japan JIAM* **19**, 227-248(2002).

[7] P.J. Prince and J.R. Dormand, High order embedded Runge–Kutta formulae, *J. Comp. Appl. Math.* **7**, pp. 67-75(1981).

[8] P.W. Sharp and J.H. Verner, Completely imbedded Runge–Kutta pairs, *SIAM J. Numer. Anal.* **31**, 1169-1190(1994).

[9] S. Tracogna and B. Welfert, Two-step Runge–Kutta methods. Theory and practice, *BIT* **40**, 775-799(2000).

[10] J.H. Verner, Explicit Runge–Kutta methods with estimates of the local truncation error, *SIAM J. Numer. Anal.* **15**, 772-790(1978).

## Appendix

To solve the order conditions of the method for each starting derivative, a constraint is imposed on the nodes. To obtain nodes for each starting method to lie inside [0,1], select suitable values of $\overline{c}_3$ and $\overline{c}_5$, and then find a value of $\overline{c}_4$ which equates the following polynomial to zero.

$$
\begin{aligned}
&p(\overline{c}_3,\overline{c}_4,\overline{c}_5) = \\
&\overline{c}_3[-20\widetilde{C}_j^{[4]}+72\widetilde{C}_j^{[5]}-2880\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}+24480(\widetilde{C}_j^{[5]})^2+1036800(\widetilde{C}_j^{[5]})^3\\
&+((A+B)\widetilde{C}^{[4]})_j[12-1440\widetilde{C}_j^{[4]}+25920\widetilde{C}_j^{[5]}+2073600(\widetilde{C}_j^{[5]})^2]\\
&+((A+B)\widetilde{C}^{[4]})_j^2[1440+1036800\widetilde{C}_j^{[5]}]]+\\
&\overline{c}_3\overline{c}_4[-87\widetilde{C}_j^{[4]}+495\widetilde{C}_j^{[5]}-1800(\widetilde{C}_j^{[4]})^2+16200\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}+1036800\widetilde{C}_j^{[4]}(\widetilde{C}_j^{[5]})^2\\
&+43200(\widetilde{C}_j^{[5]})^2+((A+B)\widetilde{C}^{[4]})_j[75+360\widetilde{C}_j^{[4]}+43200\widetilde{C}_j^{[5]}+1036800\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}]]+\\
&\overline{c}_3\overline{c}_4\overline{c}_5[114\widetilde{C}_j^{[4]}-450\widetilde{C}_j^{[5]}+720(\widetilde{C}_j^{[4]})^2+5760\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}-18000(\widetilde{C}_j^{[5]})^2\\
&-86400\widetilde{C}_j^{[4]}(\widetilde{C}_j^{[5]})^2+86400(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]}]\\
&+((A+B)\widetilde{C}^{[4]})_j[-90+86400(\widetilde{C}_j^{[4]})^2-21600\widetilde{C}_j^{[5]}+8640\widetilde{C}_j^{[4]}+172800\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}]\\
&+((A+B)\widetilde{C}^{[4]})_j^2[-3600+259200\widetilde{C}_j^{[4]}]]+\\
&\overline{c}_3\overline{c}_5[27\widetilde{C}_j^{[4]}-75\widetilde{C}_j^{[5]}+360(\widetilde{C}_j^{[4]})^2-360\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}+((A+B)\widetilde{C}^{[4]})_j[-15+1080\widetilde{C}_j^{[4]}]]+\\
&\overline{c}_3\overline{c}_4^2[168\widetilde{C}_j^{[4]}-600\widetilde{C}_j^{[5]}+1440(\widetilde{C}_j^{[4]})^2+11520\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}-36000(\widetilde{C}_j^{[5]})^2\\
&-172800\widetilde{C}_j^{[4]}(\widetilde{C}_j^{[5]})^2+172800(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]}\\
&+((A+B)\widetilde{C}^{[4]})_j[-120+17280\widetilde{C}_j^{[4]}-43200\widetilde{C}_j^{[5]}+172800(\widetilde{C}_j^{[4]})^2+345600\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}]\\
&+((A+B)\widetilde{C}^{[4]})_j^2[-7200+518400\widetilde{C}_j^{[4]}]]+\\
&\overline{c}_3\overline{c}_4^2\overline{c}_5[-390\widetilde{C}_j^{[4]}+1350\widetilde{C}_j^{[5]}-10080(\widetilde{C}_j^{[4]})^2+7200\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}+72000(\widetilde{C}_j^{[5]})^2\\
&-86400(\widetilde{C}_j^{[4]})^3+86400(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]}\\
&+((A+B)\widetilde{C}^{[4]})_j[270-21600\widetilde{C}_j^{[4]}+86400\widetilde{C}_j^{[5]}-259200(\widetilde{C}_j^{[4]})^2]\\
&+14400((A+B)\widetilde{C}^{[4]})_j^2]+\\
&\overline{c}_3^2[27\widetilde{C}_j^{[4]}-75\widetilde{C}_j^{[5]}+360(\widetilde{C}_j^{[4]})^2-360\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}+((A+B)\widetilde{C}^{[4]})_j[-15+1080\widetilde{C}_j^{[4]}]]+\\
&\overline{c}_3^2\overline{c}_4[-114\widetilde{C}_j^{[4]}-330\widetilde{C}_j^{[5]}-3600(\widetilde{C}_j^{[4]})^2-48960\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}+104400(\widetilde{C}_j^{[5]})^2\\
&-604800(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]}+86400\widetilde{C}_j^{[4]}(\widetilde{C}_j^{[5]})^2\\
&+((A+B)\widetilde{C}^{[4]})_j[-30-8640\widetilde{C}_j^{[4]}+108000\widetilde{C}_j^{[5]}-86400(\widetilde{C}_j^{[4]})^2-172800\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}]\\
&+((A+B)\widetilde{C}^{[4]})_j^2[3600-259200\widetilde{C}_j^{[4]}]]+\\
&\overline{c}_3^2\overline{c}_5[-54\widetilde{C}_j^{[4]}+150\widetilde{C}_j^{[5]}-720(\widetilde{C}_j^{[4]})^2-5760\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}+18000(\widetilde{C}_j^{[5]})^2
\end{aligned}
$$

$$
\begin{aligned}
&- 86400(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]} + 86400\widetilde{C}_j^{[4]}(\widetilde{C}_j^{[5]})^2 \\
&+ ((A+B)\widetilde{C}^{[4]})_j[30 - 8640\widetilde{C}_j^{[4]} + 21600\widetilde{C}_j^{[5]} - 86400(\widetilde{C}_j^{[4]})^2 - 172800\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}] \\
&+ ((A+B)\widetilde{C}^{[4]})_j^2[3600 - 259200\widetilde{C}_j^{[4]}]] + \\
&\overline{c}_3^2\overline{c}_4\overline{c}_5[135\widetilde{C}_j^{[4]} + 225\widetilde{C}_j^{[5]} + 5040(\widetilde{C}_j^{[4]})^2 - 25200\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} + 72000(\widetilde{C}_j^{[5]})^2 \\
&+ 43200(\widetilde{C}_j^{[4]})^3 - 43200(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]} \\
&+ ((A+B)\widetilde{C}^{[4]})_j[45 - 10800\widetilde{C}_j^{[4]} + 86400\widetilde{C}_j^{[5]} + 129600(\widetilde{C}_j^{[4]})^2] \\
&+ 14400((A+B)\widetilde{C}^{[4]})_j^2] + \\
&\overline{c}_3^2\overline{c}_4^2[35\widetilde{C}_j^{[4]} + 525\widetilde{C}_j^{[5]} + 5040(\widetilde{C}_j^{[4]})^2 - 32400\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} + 108000(\widetilde{C}_j^{[5]})^2 \\
&+ 43200(\widetilde{C}_j^{[4]})^3 - 43200(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]} \\
&+ ((A+B)\widetilde{C}^{[4]})_j[105 - 18000\widetilde{C}_j^{[4]} + 129600\widetilde{C}_j^{[5]} + 129600(\widetilde{C}_j^{[4]})^2] \\
&+ 21600((A+B)\widetilde{C}^{[4]})_j^2] + \\
&\overline{c}_3^2\overline{c}_4^2\overline{c}_5[60\widetilde{C}_j^{[4]} - 1500\widetilde{C}_j^{[5]} + 7200(\widetilde{C}_j^{[4]})^2 - 64800\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} \\
&+ ((A+B)\widetilde{C}^{[4]})_j[-300 - 36000\widetilde{C}_j^{[4]}]] + \\
&\overline{c}_3^3\overline{c}_4[225\widetilde{C}_j^{[4]} - 225\widetilde{C}_j^{[5]} + 15120(\widetilde{C}_j^{[4]})^2 - 54000\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} + 108000(\widetilde{C}_j^{[5]})^2 \\
&+ 129600(\widetilde{C}_j^{[4]})^3 - 129600(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]} \\
&+ ((A+B)\widetilde{C}^{[4]})_j[-45 - 10800\widetilde{C}_j^{[4]} + 129600\widetilde{C}_j^{[5]} + 388800(\widetilde{C}_j^{[4]})^2] \\
&+ 21600((A+B)\widetilde{C}^{[4]})_j^2] + \\
&\overline{c}_3^3\overline{c}_4\overline{c}_5[-240\widetilde{C}_j^{[4]} - 28800\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} - 28800((A+B)\widetilde{C}^{[4]})_j\widetilde{C}_j^{[4]}] + \\
&\overline{c}_3^3\overline{c}_4^2[-240\widetilde{C}_j^{[4]} - 28800\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} - 28800((A+B)\widetilde{C}^{[4]})_j\widetilde{C}_j^{[4]}] + \\
&\overline{c}_3^3\overline{c}_4^2\overline{c}_5[600\widetilde{C}_j^{[4]} + 14400(\widetilde{C}_j^{[4]})^2] + \\
&\overline{c}_4[40\widetilde{C}_j^{[4]} - 144\widetilde{C}_j^{[5]} + 5760\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} - 48960(\widetilde{C}_j^{[5]})^2 - 2073600(\widetilde{C}_j^{[5]})^3 \\
&+ ((A+B)\widetilde{C}^{[4]})_j[-24 + 2880\widetilde{C}_j^{[4]} - 51840\widetilde{C}_j^{[5]} - 4147200(\widetilde{C}_j^{[5]})^2] \\
&+ ((A+B)\widetilde{C}^{[4]})_j^2[-2880 - 2073600\widetilde{C}_j^{[5]}]] + \\
&\overline{c}_4\overline{c}_5[-54\widetilde{C}_j^{[4]} + 150\widetilde{C}_j^{[5]} - 720(\widetilde{C}_j^{[4]})^2 + 720\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} \\
&+ ((A+B)\widetilde{C}^{[4]})_j[30 - 2160\widetilde{C}_j^{[4]}]] + \\
&\overline{c}_4^2[-54\widetilde{C}_j^{[4]} + 150\widetilde{C}_j^{[5]} - 720(\widetilde{C}_j^{[4]})^2 + 720\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} \\
&+ ((A+B)\widetilde{C}^{[4]})_j[30 - 2160\widetilde{C}_j^{[4]}]] + \\
&\overline{c}_4^2\overline{c}_5[108\widetilde{C}_j^{[4]} - 300\widetilde{C}_j^{[5]} + 1440(\widetilde{C}_j^{[4]})^2 + 11520\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]} - 36000(\widetilde{C}_j^{[5]})^2 \\
&- 172800\widetilde{C}_j^{[4]}(\widetilde{C}_j^{[5]})^2 + 172800(\widetilde{C}_j^{[4]})^2\widetilde{C}_j^{[5]} \\
&+ ((A+B)\widetilde{C}^{[4]})_j[-60 + 17280\widetilde{C}_j^{[4]} - 43200\widetilde{C}_j^{[5]} + 172800(\widetilde{C}_j^{[4]})^2 \\
&+ 345600\widetilde{C}_j^{[4]}\widetilde{C}_j^{[5]}] + ((A+B)\widetilde{C}^{[4]})_j^2[-7200 + 518400\widetilde{C}_j^{[4]}]]
\end{aligned}
$$