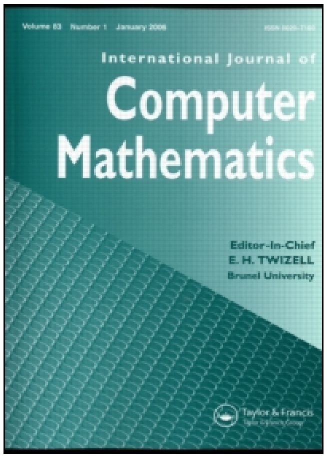


# International Journal of Computer Mathematics

Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713455451

## Non-linear singular systems using RK-Butcher algorithms

David J. Evans [a]; K. Murugesan [b]; S. Sekar [b]; Hyun-Min Kim [c]
[a] Parallelism, Algorithms and Architectures Research Centre, Department of Computer Science, Loughborough University. Loughborough, LE11 3TU. UK
[b] Department of Mathematics, National Institute of Technology. Tiruchirappalli, Tamil Nadu, 620 015. India
[c] Department of Mathematics, Pusan National University. Pusan, 609-735. Republic of Korea

To cite this Article: David J. Evans, K. Murugesan, S. Sekar and Hyun-Min Kim , 'Non-linear singular systems using RK-Butcher algorithms', International Journal of Computer Mathematics, 83:1, 131 - 142
To link to this article: DOI: 10.1080/00207160500069888
URL: http://dx.doi.org/10.1080/00207160500069888

*International Journal of Computer Mathematics*
Vol. 83, No. 1, January 2006, 131–142



# Non-linear singular systems using RK–Butcher algorithms

DAVID J. EVANS†, K. MURUGESAN*‡, S. SEKAR‡ and HYUN-MIN KIM§

†Parallelism, Algorithms and Architectures Research Centre, Department of Computer Science, Loughborough University, Loughborough LE11 3TU, UK
‡Department of Mathematics, National Institute of Technology, Tiruchirappalli-620 015, Tamil Nadu, India
§Department of Mathematics, Pusan National University, Pusan 609-735, Republic of Korea

*(Received 24 August 2004)*

The Runge–Kutta (RK)–Butcher algorithm is used to study time-invariant and time-varying non-linear singular systems. The results (discrete solutions) obtained using the RK method based on the arithmetic mean (RKAM), single-term Walsh series (STWS) and RK–Butcher algorithms are compared with the exact solutions of the non-linear singular systems for the time-invariant and time-varying cases. It is found that the solution obtained using the RK–Butcher algorithm is closer to the exact solutions of the non-linear singular systems. Stability regions for the RKAM, STWS and RK–Butcher algorithms are presented. Error graphs for discrete and exact solutions are presented in a graphical form to highlight the efficiency of this method. The RK–Butcher algorithm can easily be implemented using a digital computer and the solution can be obtained for any length of time for both time-invariant and time-varying cases for these non-linear singular systems, which is an added advantage of this algorithm.

*Keywords*: Time-invariant non-linear singular systems; Time-varying non-linear singular systems; STWS algorithm; RKAM algorithm; RK–Butcher algorithm

*C.R. Category*: G1.7

## 1. Introduction

Most realistic non-linear systems, in particular singular non-linear systems, do not admit any analytical solution and hence must be solved using a numerical technique. Conventional methods, such as the Euler, Taylor series and Adams–Moultan methods, are restricted to a very small step size in order to obtain a stable solution, which naturally requires much computer time. Many new methods have been developed to overcome this step-size constraint imposed by numerical stability, and these are reviewed by Butcher [1], Murugesan and colleagues [2–7] and Park *et al.* [8].

*Corresponding author. Email: murugu@nitt.edu

*International Journal of Computer Mathematics*
ISSN 0020-7160 print/ISSN 1029-0265 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207160500069888

Runge–Kutta (RK) methods have been used by many researchers [2–6, 9–16] to determine numerical solutions for problems, which are modelled as initial value problems (IVPs) involving differential equations, that arise in the fields of Science and Engineering. Although the RK method was introduced at the beginning of the twentieth century, research in this area is still very active and its applications are enormous because of its extending accuracy in the determination of approximate solutions and its flexibility.

RK methods have become very popular, both as computational techniques and in research applications [1, 14, 17]. The method was developed by Runge in the mid-1890s and extended by Kutta a few years later. They developed algorithms which solve differential equations efficiently and yet are the equivalent of approximating the exact solutions by matching $n$ terms of the Taylor series expansion.

RK algorithms have always been considered to be excellent tools for the numerical integration of ordinary differential equations (ODEs). The fact that RK methods are self-starting, easy to program and extremely accurate and versatile in ODE problems has led to their continuous analysis and use in mathematical research. The beauty of the RK pair is that it requires no extra function evaluations, which is the most time-consuming aspect of all ODE solvers. This breakthrough has initiated a search for RK algorithms of increasingly high-order for better error estimates.

Butcher [17] derived the best RK pair, together with an error estimate, and this is known as the RK–Butcher algorithm. It is nominally considered to be sixth order since it requires six function evaluations (it looks like a sixth-order method but in fact is a fifth-order method). In practice, the 'working order' is closer to 5 (fifth order), but the accuracy of the results obtained exceeds that of all other algorithms examined, including the RK–Fehlberg, RK–centroidal mean (RKCeM) and RK–arithmetic mean (RKAM) methods.

Bader [18, 19] introduced the RK–Butcher algorithm for finding the truncation error estimates and intrinsic accuracies and the early detection of stiffness in coupled differential equations that arises in theoretical chemistry problems. Recently, Murugesan *et al.* [7] and Park *et al.* [8] applied the RK–Butcher algorithms for finding the numerical solutions of an industrial robot arm control problem and optimal control of linear singular systems. In this article, we present a new approach for solving the time-invariant and time-varying non-linear singular systems using RK–Butcher algorithms with more accuracy.

## 2. RK–Butcher algorithms

The normal order of an RK algorithm is the approximate number of leading terms of an infinite Taylor series, which calculates the trajectory of a moving point, which was discussed by Shampine and Gordon [20]. The remainder of the infinite sum excluded is referred to as the local truncation error (LTE). RK algorithms are forward-looking predictors, that is, they use no information from preceding steps to predict the future position of a point. For this reason, they require a minimum of input data and consequently are very easy to program and simple to use.

The general $p$-stage Runge–Kutta method for solving an IVP is

$$\dot{y} = f(x, y) \tag{1}$$

with the initial condition $y(x_0) = y_0$ is defined by

$$y_{n+1} = y_n + h \sum_{i=1}^{p} b_i k_i$$

where

$$k_i = f\left(x_n + c_i h,\, y_n + h\sum_{j=1}^{p} a_{ij}k_j\right), \quad i = 1, 2, 3, \ldots, p$$

and

$$c_i = \sum_{j=1}^{p} a_{ij}; \quad i = 1, 2, \ldots, p$$

with $c$ and $b$ are $p$ dimensional vectors and $A(a_{ij})$ be the $p \times p$ matrix. Then the Butcher array is of the form

$$\begin{array}{c|ccccc}
c_1 & a_{11} & & & & \\
c_2 & a_{21} & a_{22} & & & \\
c_3 & a_{31} & a_{32} & & a_{33} & \\
. & . & . & & . & \\
. & . & . & & . & \\
. & . & . & & . & \\
. & . & . & & . & \\
c_p & a_{p1} & a_{p2} & \ldots & a_{p,p-1} & a_{p,p} \\
\hline
 & b_1 & b_2 & & b_{p-1} & b_p
\end{array}$$

then the RK–Butcher algorithm of the above equation (1) is of the form

$$\begin{aligned}
k_1 &= hf(x_n, y_n) \\
k_2 &= hf\left(x_n + \frac{h}{4}, y_n + \frac{k_1}{4}\right) \\
k_3 &= hf\left(x_n + \frac{h}{4}, y_n + \frac{k_1}{8} + \frac{k_2}{8}\right) \\
k_4 &= hf\left(x_n + \frac{h}{2}, y_n - \frac{k_2}{2} + k_3\right) \\
k_5 &= hf\left(x_n + \frac{3h}{4}, y_n + \frac{3k_1}{16} + \frac{9k_4}{16}\right) \\
k_6 &= hf\left(x_n + h, y_n - \frac{3k_1}{7} + \frac{2k_2}{7} + \frac{12k_3}{7} - \frac{12k_4}{7} + \frac{8k_5}{7}\right)
\end{aligned} \tag{2}$$

5th order predictor

$$y_{n+1} = y_n + \frac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)$$

4th order predictor

$$y^*_{n+1} = y_n + \frac{1}{6}(k_1 + 4k_4 + k_6)$$

Local Truncation Error Estimate (*EE*)

$$EE = y_{n+1} - y^*_{n+1}$$

Then the formation of the Butcher array of the above equation (2) takes the following form

$$\begin{array}{c|cccccc}
0 & & & & & & \\
\frac{1}{4} & \frac{1}{4} & & & & & \\
\frac{1}{4} & \frac{1}{8} & \frac{1}{8} & & & & \\
\frac{1}{2} & 0 & -\frac{1}{2} & 1 & & & \\
\frac{3}{4} & \frac{3}{16} & 0 & 0 & \frac{9}{16} & & \\
1 & -\frac{3}{7} & \frac{2}{7} & \frac{12}{7} & -\frac{12}{7} & \frac{8}{7} & \\
\hline
 & \frac{7}{90} & 0 & \frac{32}{90} & \frac{12}{90} & \frac{32}{90} & \frac{7}{90} \\
 & \frac{1}{6} & 0 & 0 & \frac{4}{6} & 0 & \frac{1}{6}
\end{array}$$

This Butcher array plays a vital role in the stability regions and is presented in the next section.

## 3. Stability regions

Consider the test equation $\dot{y} = \lambda y$ where $\lambda$ is a complex constant and it is used to determine the stability regions of these methods.

$$k_1 = f(y_n) = \lambda y_n$$

$$k_2 = f\left(y_n + \frac{hk_1}{4}\right) = \lambda y_n \left(1 + \frac{h\lambda}{4}\right)$$

$$k_3 = f\left(y_n + \frac{hk_1}{8} + \frac{hk_2}{8}\right) = \lambda y_n \left(1 + \frac{h\lambda}{8} + \frac{h\lambda}{8}\left(1 + \frac{h\lambda}{4}\right)\right)$$

$$k_4 = f\left(y_n - \frac{hk_2}{2} + hk_3\right) = \lambda y_n \left(1 - \frac{h\lambda}{2}\left(1 + \frac{h\lambda}{4}\right) + h\lambda\left(1 + \frac{h\lambda}{8} + \frac{h\lambda}{8}\left(1 + \frac{h\lambda}{4}\right)\right)\right)$$

$$k_5 = f\left(y_n - \frac{3hk_1}{16} + \frac{9hk_4}{16}\right) = \lambda y_n \left(1 + \frac{3h\lambda}{16} + \frac{9h\lambda}{16}\left(\begin{array}{l} 1 - \frac{h\lambda}{2}\left(1 + \frac{h\lambda}{4}\right) \\ +h\lambda\left(1 + \frac{h\lambda}{8} + \frac{h\lambda}{8}\left(1 + \frac{h\lambda}{4}\right)\right)\end{array}\right)\right)$$

$$k_6 = f\left(y_n - \frac{3hk_1}{7} + \frac{2hk_2}{7} + \frac{12hk_3}{7} - \frac{12hk_4}{7} + \frac{8hk_5}{7}\right)$$

$$= \lambda y_n \left(\begin{array}{l} 1 - \frac{3h\lambda}{7} + \frac{2h\lambda}{7}\left(1 + \frac{h\lambda}{4}\right) + \frac{12h\lambda}{7}\left(1 + \frac{h\lambda}{8} + \frac{h\lambda}{8}\left(1 + \frac{h\lambda}{4}\right)\right) \\ -\frac{12h\lambda}{7}\left(1 - \frac{h\lambda}{2}\left(1 + \frac{h\lambda}{4}\right) + h\lambda\left(1 + \frac{h\lambda}{8} + \frac{h\lambda}{8}\left(1 + \frac{h\lambda}{4}\right)\right)\right) \\ +\frac{8h\lambda}{7}\left(1 + \frac{3h\lambda}{16} + \frac{9h\lambda}{16}\left(1 - \frac{h\lambda}{2}\left(1 + \frac{h\lambda}{4}\right) + h\lambda\left(1 + \frac{h\lambda}{8} + \frac{h\lambda}{8}\left(1 + \frac{h\lambda}{4}\right)\right)\right)\right) \end{array}\right)$$

Substituting $z = h\lambda$ we get

$$k_1 = f(y_n) = \lambda y_n$$

$$k_2 = \lambda y_n\left(1 + \frac{z}{4}\right)$$

$$k_3 = \lambda y_n\left(1 + \frac{z}{8} + \frac{z}{8}\left(1 + \frac{z}{4}\right)\right)$$

$$k_4 = \lambda y_n\left(1 - \frac{z}{2}\left(1 + \frac{z}{4}\right) + z\left(1 + \frac{z}{8} + \frac{z}{8}\left(1 + \frac{z}{4}\right)\right)\right)$$

$$k_5 = \lambda y_n\left(1 + \frac{3z}{16} + \frac{9z}{16}\left(1 - \frac{z}{2}\left(1 + \frac{z}{4}\right) + z\left(1 + \frac{z}{8} + \frac{z}{8}\left(1 + \frac{z}{4}\right)\right)\right)\right)$$

$$k_6 = \lambda y_n \left(\begin{array}{l} 1 - \frac{3z}{7} + \frac{2z}{7}\left(1 + \frac{z}{4}\right) + \frac{12z}{7}\left(1 + \frac{z}{8} + \frac{z}{8}\left(1 + \frac{z}{4}\right)\right) \\ -\frac{12z}{7}\left(1 - \frac{z}{2}\left(1 + \frac{z}{4}\right) + z\left(1 + \frac{z}{8} + \frac{z}{8}\left(1 + \frac{z}{4}\right)\right)\right) \\ +\frac{8z}{7}\left(1 + \frac{3z}{16} + \frac{9z}{16}\left(1 - \frac{z}{2}\left(1 + \frac{z}{4}\right) + z\left(1 + \frac{z}{8} + \frac{z}{8}\left(1 + \frac{z}{4}\right)\right)\right)\right) \end{array}\right)$$

Then the 5th order predictor formula is

$$y_{n+1} = y_n + \frac{h}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6),$$

Substituting the values of $k_1, k_2, k_3, k_4, k_5$ and $k_6$ then we obtain

$$y_{n+1} = y_n + \frac{h\lambda y_n}{90}\left(90 + \frac{90}{2}z + \frac{30}{2}z^2 + \frac{30}{8}z^3 + \frac{30}{40}z^4 + \frac{30}{240}z^5\right)$$

divide both sides by $y_n$ then the stability polynomial $Q(z) = y_{n+1}/y_n$ is given as

$$Q(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!} + \frac{z^5}{5!} + \frac{z^6}{6!}.$$

Figure 1 show that a comparative study of the stability regions of the RKAM method, STWS and the RK–Butcher algorithm. In this stability region, the range for the real part of $\lambda$ in RKAM is $-3.463 < \text{Re}(\lambda) < 0.0$, STWS is $-3.284 < \text{Re}(\lambda) < 0.0$ and where as in the RK–Butcher algorithm it is $-2.780 < \text{Re}(\lambda) < 0.0$. It reveals that the RK–Butcher algorithm converges faster than the other two discussed methods RKAM and STWS.

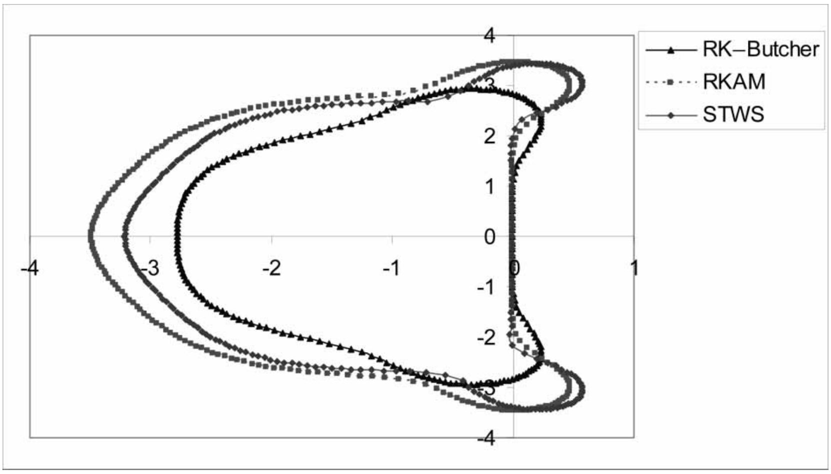


Figure 1. Stability regions for RKAM, STWS and RK–Butcher algorithms.

## 4. Non-linear singular systems

Consider the time invariant non-linear singular system of the form

$$K\dot{x}(t) = Ax(t) + f(x(t)) \tag{3}$$

with $x(0) = x_0$, where $K$ is an $n \times n$ singular matrix, $A$ is an $n \times n$ matrix, $x(t)$ is an $n$-state vector and $f$ is an '$n$' vector function. In order to make the above system (3) time varying case some of the components (not necessarily all the elements) in the system (3) are converted to time varying and then the system will be of the following form

$$K(t)\dot{x}(t) = A(t)x(t) + f(x(t)) \tag{4}$$

with $x(0) = x_0$, where this $K(t)$ is an $n \times n$ singular matrix, $A(t)$ is an $n \times n$ matrix, $x(t)$ is an $n$-state vector and $f$ is an '$n$' vector function. The time varying singular non-linear systems are much more difficult to solve than the time invariant systems. Therefore, many authors have tried various transform methods to over come these difficulties. In this article, we introduce RK–Butcher algorithms with more accuracy to solve these time-invariant and time-varying singular non-linear systems.

## 5. Numerical examples

In this section, two examples are presented, one is for time invariant and the other is for time varying case. Numerical solutions are obtained using three methods like RKAM method, STWS and RK–Butcher algorithms.

### 5.1 *Example*

Consider the following time invariant nonlinear singular system (Campbell [21], Lin and Yang [22])

$$K = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix} \quad \text{and} \quad f(x(t)) = \begin{bmatrix} 0 \\ -x^2 \end{bmatrix} \tag{5}$$

with initial condition $x(0) = [0 \quad 0]^T$.

Table 1. Solutions for time-invariant system (5) for various values of '$x_1$'.

| | | Discrete solution $x_1$-values | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S. No. | Time | Exact solutions | RKAM solutions | RKAM error | STWS solutions | STWS error | RK–Butcher solutions | RK–Butcher error |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.25 | −0.25 | −0.25640 | 0.00640 | −0.254 | 0.004 | −0.25002 | 0.00002 |
| 3 | 0.5 | −0.50 | −0.50295 | 0.00295 | −0.504 | 0.004 | −0.50007 | 0.00007 |
| 4 | 0.75 | −0.75 | −0.75673 | 0.00673 | −0.754 | 0.004 | −0.75009 | 0.00009 |
| 5 | 1 | −1.00 | −1.00729 | 0.00729 | −1.004 | 0.004 | −1.00014 | 0.00014 |
| 6 | 1.25 | −1.25 | −1.25651 | 0.00651 | −1.254 | 0.004 | −1.25017 | 0.00017 |
| 7 | 1.5 | −1.50 | −1.50652 | 0.00652 | −1.504 | 0.004 | −1.50019 | 0.00019 |
| 8 | 1.75 | −1.75 | −1.75239 | 0.00239 | −1.754 | 0.004 | −1.75022 | 0.00022 |
| 9 | 2 | −2.00 | −2.00200 | 0.00200 | −2.004 | 0.004 | −2.00026 | 0.00026 |

The exact solutions are

$$x_1(t) = -t$$
$$x_2(t) = \frac{t^2}{2} \tag{6}$$

The results (discrete solutions) obtained using RKAM, STWS and RK–Butcher algorithms (with step size time $t = 0.25$) along with the exact solutions and its absolute errors between them are calculated and are presented in tables 1 and 2. To highlight the efficiency of the RK–Butcher algorithms and to distinguish the effect of the errors in accordance with the exact solutions, a graphical representation is presented in figures 2 and 3 for selected values of '$x_1$' and '$x_2$', using three-dimensional effect.

## 5.2 *Example*

Consider the time varying non-linear singular system of the following form (Hsiao and Wang [23] and Sepehrian and Razzaghi [24])

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & t^2 \\ 0 & 0 & 0 \end{bmatrix} \dot{x}(t) = \begin{bmatrix} t x_1(t) + x_2(t) \\ \exp(t) x_1(t) x_2(t) \\ x_2(t)(x_1(t) + x_3(t)) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 2t^2 \exp(-t) \\ 0 \end{bmatrix}, \tag{7}$$

Table 2. Solutions for time-invariant system (5) for various values of '$x_2$'.

| | | Discrete solution $x_1$-values | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S. No. | Time | Exact solutions | RKAM solutions | RKAM error | STWS solutions | STWS error | RK–Butcher solutions | RK–Butcher error |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.25 | 0.031 | 0.031974 | 0.000974 | 0.032 | 0.001 | 0.031002 | 0.000002 |
| 3 | 0.5 | 0.125 | 0.125492 | 0.000492 | 0.127 | 0.002 | 0.125007 | 0.000007 |
| 4 | 0.75 | 0.281 | 0.281510 | 0.000510 | 0.285 | 0.004 | 0.281009 | 0.000009 |
| 5 | 1 | 0.500 | 0.500759 | 0.000759 | 0.505 | 0.005 | 0.500014 | 0.000014 |
| 6 | 1.25 | 0.781 | 0.781651 | 0.000651 | 0.787 | 0.006 | 0.781017 | 0.000017 |
| 7 | 1.5 | 1.125 | 1.125658 | 0.000658 | 1.132 | 0.007 | 1.125019 | 0.000019 |
| 8 | 1.75 | 1.531 | 1.531662 | 0.000662 | 1.540 | 0.009 | 1.531022 | 0.000022 |
| 9 | 2 | 2.000 | 2.000982 | 0.000982 | 2.010 | 0.010 | 2.000026 | 0.000026 |

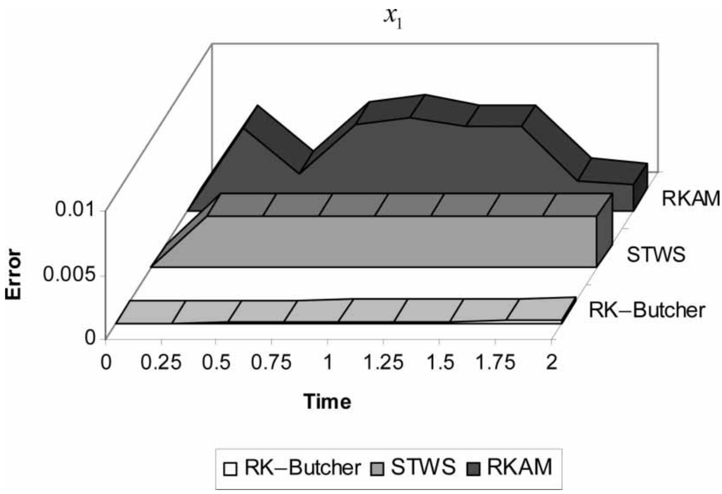


Figure 2. Error graph for '$x_1$' at various time intervals.

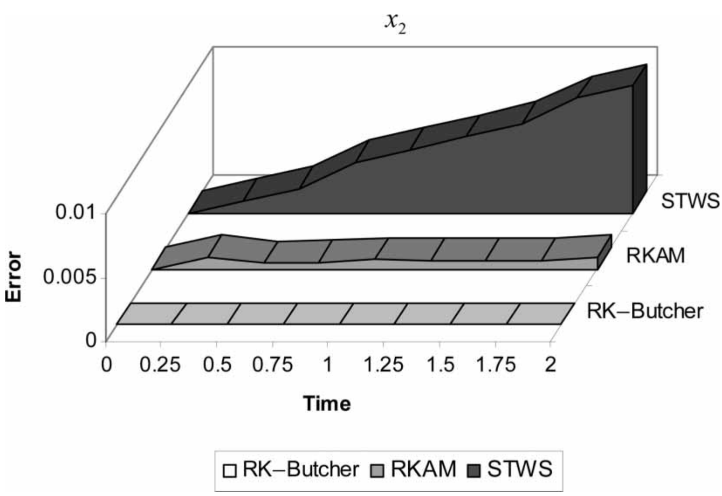


Figure 3. Error graph for '$x_2$' at various time intervals.

with initial condition

$$x(0) = \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix}.$$

The exact solutions are

$$x(t) = \begin{bmatrix} 2\exp(-t)(1-t) \\ t^2\exp(-t) \\ -2\exp(-t)(1-t) \end{bmatrix} \quad (8)$$

The results (discrete solutions) were obtained using the RKAM method; STWS and RK–Butcher algorithms (with step size time $t = 0.25$) along with the exact solutions and the absolute errors between them were calculated and are presented in tables 3–5. To highlight the efficiency of the RK–Butcher algorithms and to distinguish the effect of the errors in

Table 3. Solutions for time varying system (7) at various values of '$x_1$'.

| | | Discrete solution $x_1$-values | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S. No. | Time | Exact solutions | RKAM solutions | RKAM error | STWS solutions | STWS error | RK–Butcher solutions | RK–Butcher error |
| 1 | 0 | 2 | 2 | 0 | 2 | 0 | 2 | 0 |
| 2 | 0.25 | 0.778801 | 0.779801 | 0.001 | 0.783458 | 0.00466 | 0.778805 | 4E–06 |
| 3 | 0.5 | 0 | −0.00935 | 0.00935 | −0.09452 | 0.09452 | 0.000006 | 6E–06 |
| 4 | 0.75 | −0.47236 | −0.47423 | 0.00187 | −0.49452 | 0.02216 | −0.47239 | 3E–05 |
| 5 | 1 | −0.73575 | −0.73857 | 0.00282 | −0.76504 | 0.02929 | −0.73579 | 4E–05 |
| 6 | 1.25 | −0.85951 | −0.86015 | 0.00064 | −0.89451 | 0.035 | −0.85956 | 5E–05 |
| 7 | 1.5 | −0.89252 | −0.89658 | 0.00406 | −0.94892 | 0.0564 | −0.89258 | 6E–05 |
| 8 | 1.75 | −0.86886 | −0.86836 | 0.0005 | −0.92765 | 0.05879 | −0.86893 | 7E–05 |
| 9 | 2 | −0.81201 | −0.81394 | 0.00193 | −0.88310 | 0.07109 | −0.81209 | 8E–05 |

Table 4. Solutions for time-varying system (7) at various values of '$x_2$'.

| | | Discrete solution $x_2$-values | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S. No. | Time | Exact solutions | RKAM solutions | RKAM error | STWS solutions | STWS error | RK–Butcher solutions | RK–Butcher error |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.25 | 0.048675 | 0.048875 | 0.0002 | 0.054867 | 0.00619 | 0.048676 | 1E–06 |
| 3 | 0.5 | 0.151632 | 0.152032 | 0.0004 | 0.175132 | 0.0235 | 0.151634 | 2E–06 |
| 4 | 0.75 | 0.265706 | 0.266306 | 0.0006 | 0.296506 | 0.0308 | 0.265709 | 3E–06 |
| 5 | 1 | 0.367879 | 0.368679 | 0.0008 | 0.403879 | 0.036 | 0.367883 | 4E–06 |
| 6 | 1.25 | 0.447663 | 0.448663 | 0.001 | 0.494763 | 0.0471 | 0.447667 | 4E–06 |
| 7 | 1.5 | 0.502042 | 0.508642 | 0.0066 | 0.552042 | 0.05 | 0.502048 | 6E–06 |
| 8 | 1.75 | 0.532182 | 0.539882 | 0.0077 | 0.605382 | 0.0732 | 0.532189 | 7E–06 |
| 9 | 2 | 0.541341 | 0.550041 | 0.0087 | 0.635141 | 0.0938 | 0.541349 | 8E–06 |

accordance with the exact solutions, a graphical representation is presented in figures 4–6 for selected values of '$x_1$', '$x_2$' and '$x_3$', using three-dimensional effect.

## 6. Conclusions

The obtained results (discrete solutions) of the non-linear singular systems for time-invariant and time-varying cases show that the RK–Butcher algorithm works well for finding the state

Table 5. Solutions for time-varying system (7) at various values of '$x_3$'.

| | | Discrete solution $x_3$-values | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S. No. | Time | Exact solutions | RKAM solutions | RKAM error | STWS solutions | STWS error | RK–Butcher solutions | RK–Butcher error |
| 1 | 0 | −2 | −2 | 0 | −2 | 0 | −2 | 0 |
| 2 | 0.25 | −0.77880 | −0.77890 | 1E–04 | −0.78880 | 0.01 | −0.77881 | 1E–05 |
| 3 | 0.5 | 0 | 0.000089 | 0.00009 | 0.000009 | 9E–06 | 0 | 0 |
| 4 | 0.75 | 0.472366 | 0.472466 | 1E–04 | 0.472367 | 1E–06 | 0.472367 | 1E–06 |
| 5 | 1 | 0.735758 | 0.735958 | 0.0002 | 0.735760 | 2E–06 | 0.735760 | 2E–06 |
| 6 | 1.25 | 0.859514 | 0.859914 | 0.0004 | 0.859517 | 3E–06 | 0.859517 | 3E–06 |
| 7 | 1.5 | 0.892521 | 0.892921 | 0.0004 | 0.892525 | 4E–06 | 0.892525 | 4E–06 |
| 8 | 1.75 | 0.868869 | 0.869569 | 0.0007 | 0.868875 | 6E–06 | 0.868876 | 7E–06 |
| 9 | 2 | 0.812011 | 0.813011 | 0.001 | 0.812019 | 8E–06 | 0.812019 | 8E–06 |

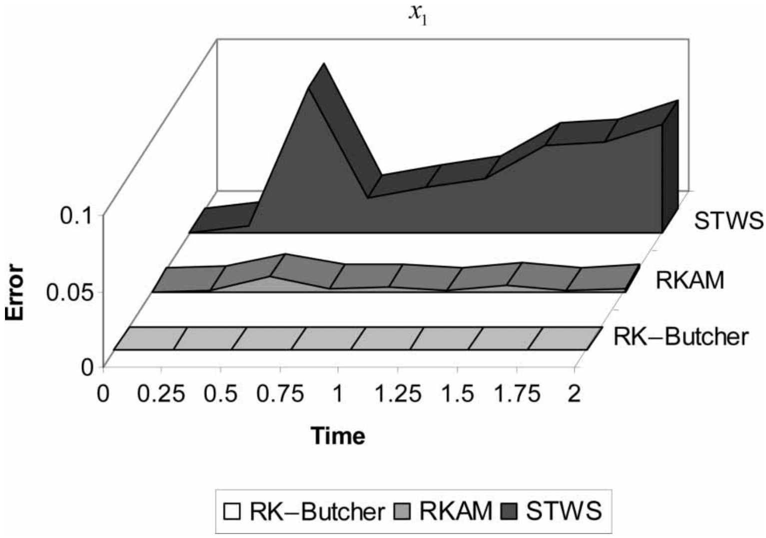


Figure 4. Error graph for '$x_1$' at various time intervals for time-varying case.

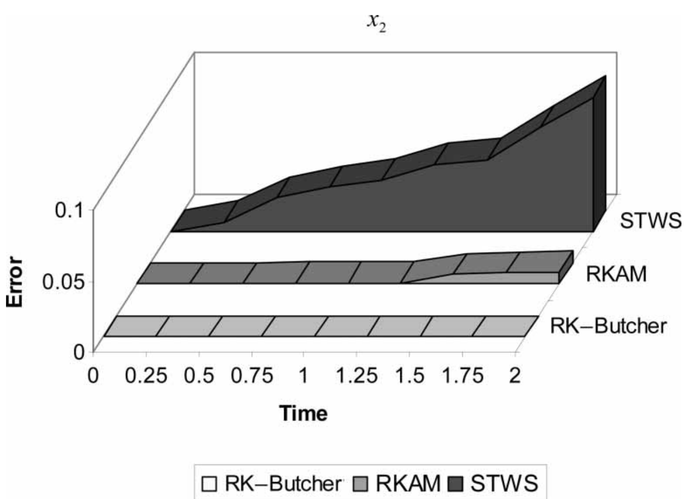


Figure 5. Error graph for '$x_2$' at various time intervals for time-varying case.

vector. From tables 1–5, we can observe that for most of the time intervals, the absolute error is less (almost no error) in the RK–Butcher algorithms when compared to the Runge–Kutta Arithmetic Mean (RKAM) method, Single Term Walsh Series (STWS) Technique, which yields a small error, along with the exact solutions. From figure 1 can be noted that the stability region of the RK–Butcher algorithm is smaller than the stability region of RKAM and STWS methods. It reveals that the RK–Butcher algorithm converges faster than the other two methods discussed here. From figures 2–6, one can predict that the error is much less in the RK–Butcher algorithm when compared to the RKAM and STWS methods discussed by Balachandran and Murugesan [25] and Sepehrian and Razzaghi [24]. Hence the RK–Butcher algorithm is more suitable for studying the time-invariant and time-varying non-linear singular systems.

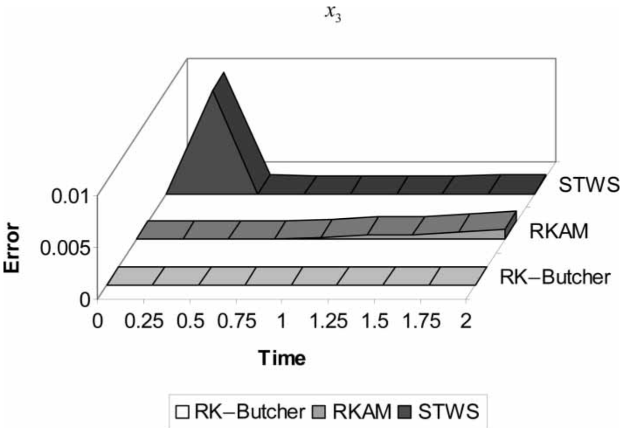


Figure 6. Error graph for '$x_3$' at various time intervals for time-varying case.

## Acknowledgements

One of the authors Dr. K. Murugesan would like to thank Professor Jong Yeoul Park, Department of Mathematics, Pusan National University, Pusan 609-735, Republic of Korea, for his kind help during the stay in Pusan National University, and also, he would like to thank the Korean Science and Engineering Foundation (KOSEF), Republic of Korea, for the financial assistance provided to him under the Brain Pool Program.

## References

[1] Butcher, J.C., 2003, *Numerical Method for Ordinary Differential Equations* (Chichester, U.K.: John Wiley & Sons).
[2] Murugesan, K., Paul Dhayabaran, D. and Evans, D.J., 1999, Analysis of different second order systems via Runge-Kutta method. *International Journal of Computer Mathematics*, **70**, 477–493.
[3] Murugesan, K., Paul Dhayabaran, D. and Evans, D.J., 1999, Analysis of second order multivariable linear system using Single Term Walsh Series technique and Runge-Kutta method. *International Journal of Computer Mathematics*, **72**, 367–374.
[4] Murugesan, K., Paul Dhayabaran, D. and Evans, D.J., 2000, Analysis of non-linear singular system from fluid dynamics using extended Runge-Kutta methods. *International Journal of Computer Mathematics*, **76**, 239–266.
[5] Murugesan, K., Paul Dhayabaran, D., Henry Amirtharaj, E.C. and Evans, D.J., 2001, A comparison of extended Runge-Kutta formulae based on variety of means to solve system of IVPs. *International Journal of Computer Mathematics*, **78**, 225–252.
[6] Murugesan, K., Paul Dhayabaran, D., Henry Amirtharaj, E.C. and Evans, D.J., 2002, A fourth order embedded Runge-Kutta RKACeM(4,4) method based on arithmetic and centroidal means with error control. *International Journal of Computer Mathematics*, **79**(2), 247–269.
[7] Murugesan, K., Sekar, S., Murugesh, V. and Park, J.Y., 2004, Numerical solution of an industrial robot arm control problem using the RK–Butcher algorithm. *International Journal of Computer Applications in Technology*, **19**(2), 132–138.
[8] Park, J.Y., Evans, D.J., Murugesan, K., Sekar, S. and Murugesh, V., 2004, Optimal control of singular systems using the RK–Butcher algorithm. *International Journal of Computer Mathematics*, **81**(2), 239–249
[9] Alexander, R.K. and J.J. Coyle, 1990, Runge-Kutta methods for differential-algebric systems. *SIAM Journal on Numerical Analysis*, **27**(3) 736–752.
[10] Evans, D.J., 1991, A new 4th Order Runge-Kutta method for initial value problems with error control. *International Journal of Computer Mathematics*, **139**, 217–227.
[11] Evans, D.J. and Yaakub, A.R., 1996, A new fifth order weighted Runge-Kutta formula. *International Journal of Computer Mathematics*, **59**, 227–243.
[12] Evans, D.J. and Yaakub, A.R., 1998, Weighted fifth order Runge-Kutta formulas for second order differential equations. *International Journal of Computer Mathematics*, **70**, 233–239.
[13] Lambert, J.D., 1991, *Numerical Methods for Ordinary Differential Systems. The Initial Value Problem* (Chichester, U.K.: John Wiley & Sons).
[14] Shampine, L.F., 1994, *Numerical Solution of Ordinary Differential Equations* (New York, U.S.A.: Chapman & Hall).

[15] Yaakub, A.R. and Evans, D.J., 1999, A fourth orders Runge-Kutta RK (4, 4) method with error control. *International Journal of Computer Mathematics*, **71**, 383–411.
[16] Yaakub, A.R. and Evans, D.J., 1999, New Runge-Kutta starters of multi-step methods. *International Journal of Computer Mathematics*, **71**, 99–104.
[17] Butcher, J.C., 1987, *The Numerical Analysis of Ordinary Differential Equations*: *Runge-Kutta and General Linear Methods* (Chichester, U.K.: John Wiley & Sons).
[18] Bader, M., 1987, A comparative study of new truncation error estimates and intrinsic accuracies of some higher order Runge-Kutta algorithms. *Computers and Chemistry*, **11** 121–124.
[19] Bader, M., 1998, A new technique for the early detection of stiffness in coupled differential equations and application to standard Runge-Kutta algorithms. *Theoretical Chemistry Accounts*, **99**, 215–219.
[20] Shampine, L.F. and Gordon, M.K., 1975, *Computer Solution of Ordinary Differential Equations – The Initial Value Problem* (San Francisco, U.S.A.: W.H. Freeman & Co).
[21] Campbell, S.L., 1982, *Singular System of Differential Equations* (London: Pitman).
[22] Lin, J.Y. and Yang, Z.H., 1988, Existence and uniqueness of solutions for non-linear singular (descriptor) systems. *International Journal of Systems Science*, **19**, 2179–2184.
[23] Hsiao, C.H. and Wang, W.J., 1999, State analysis of time-varying singular nonlinear systems via Haar wavelets. *Mathematics and Computers in Simulation*, **51**, 91–100.
[24] Sepehrian, B. and Razzaghi, M., 2003, Solution of time-varying singular nonlinear systems by Single-Term Walsh Series. *Mathematical Problems in Engineering*, **3**, 129–136.
[25] Balachandran, K. and Murugesan, K., 1990, Analysis of nonlinear singular systems via STWS method. *International Journal of Computer Mathematics*, **36**, 9–12.