Two Embedded Pairs for Solve Directly Third-Order Ordinary Differential Equation by Using Runge-Kutta Type Method (RKTGD)

Firas A. Fawzi, Zeyad M. Abdullah, Nazar K. Hussein

Department of Mathematics, Faculty of Computer Science and Mathematics, University of Tikrit, Iraq
Email: firasadil01@tu.edu.iq

Abstract. In this paper, two pairs of embedded Runge-Kutta (RK) type techniques for straightforwardly tackling third-order ordinary differential equations (ODEs) of the form $v'' = f(x, v, v')$ signified as RKTGD strategies were proposed and explored. Relying on the order conditions, the primary pair with mathematical order 4 and 3 was called RKTGD4(3), while different has order 5 and 4, and was named RKTGD5(4). The new strategies were determined so that the higher-order techniques were exact and the lower order techniques would bring about the best error estimates. At that point, variables step-size codes were created to support the methods and utilized in solving a lot of third-order problems. Comparisons were made between mathematical results and existing embedded RK pairs within the scientific literature, that require the problems to be reduced into a system of first-order ODEs, and the effectiveness of the new RKTGD pairs have appeared.

1. Introduction
In the present paper, the numerical integration of 3rd-order ODEs and initial conditions of the structure

$$v''(t) = f(x, v(t), v'(t)), \quad v(t_0) = v_0, \quad v'(t_0) = v'_0, \quad t \geq t_0$$

(1)

was considered, where $v, v' \in \mathbb{R}^d, f : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a continuous vector-valued function that doesn’t explicitly depend on the second derivatives. Some examples of such problems are found in electromagnetic waves, thin film flow and gravity driven flow [1-3]. A number of engineers, scientists, and researchers usually solve problem (1) by transforming the third-order ODEs into a system of first-order equations three times the dimension. See examples in the references [4-8]. These methods are multi-step methods, therefore, they require starting values when utilized in solving problem (1). However, it is more efficient if the problem can be solved directly with the aid of numerical methods. This paper is focused on the use of a one-step embedded explicit process, in particular, the RK type methods for solve directly third-order ODEs of the problem (1) which don’t need the beginning values. The regular sort of new procedure with $m$-stage for solving initial value problems (IVPs) (1) can be formulated in [9]:

$$v_{n+1} = v_n + h v'_n + \frac{h^2}{2} v''_n + h^3 \sum_{i=1}^{m} b_i k_i,$$

(2)

$$v'_{n+1} = v'_n + h v''_n + h^2 \sum_{i=1}^{m} b_i' k'_i,$$

(3)
\[ v''_{n+1} = v''_n + h \sum_{i=1}^{m} b_i' k_i, \]  
where

\[ k_i = f(x_n + c_i h, v_n + c_i h v'_n + \frac{h^2}{2} c_i^2 v''_n + h^3 \sum_{j=1}^{i-1} a_{ij} k_j, v'_n + c_i h v''_n + h^2 \sum_{j=1}^{i-1} \bar{a}_{ij} k_j), \]  
for \( i = 2, 3, ..., m \).

RKTGD method parameters \( b_i, b'_i, b''_i, a_{ij}, \bar{a}_{ij} \) and \( c_i \) of the new technique assumed to be realistic, \( m \) is the number of phases of the technique. Then, this method is explicit when \( a_{ij} = 0 \) for \( i \leq j \). To decide the coefficients of the technique given by (2-5), the RKTGD strategy expression is extended by utilizing Taylor’s series expansion \([10, 11]\). After being simplified mathematically, this expansion is then compared to the solution was given by Taylor’s series expansion. At that point, this process requires a great deal of mathematical and numeric calculations, however utilizing programming like Maple as in \([10]\). General order conditions for the RKTGD strategy can be gotten from the direct expansion of the local truncation error. Likewise, efficient RK codes frequently utilize embedded pair of request \( p(q) \) where \( p \) is greater or equal to \( q + 1 \), examples of these are seen in \([12-16]\). This study is significantly dedicated to additional work in this exploration field of study. Especially, it is pointed tat deriving the \( p(q) \) pairs of explicit RKTGD techniques utilized in variable step size codes to give a tinny error estimation, and it’s based on the techniques \((c,A,b,b',b'')\) of order \( p \), and \((c,A,b,b',b'')\) of order \( q \).

For Butcher Tableau, the embedded pair can be expressed as follows:

\[
\begin{array}{c|c|c}
  c & A & \bar{A} \\
  \hline
  b^T & b'^T & b''^T \\
  \hline
  \hat{b}^T & \hat{b}'^T & \hat{b}''^T \\
\end{array}
\]  

The essential reason for building the embedded pairs of explicit RKTGD strategy is to obtain a low value local error estimate that would be utilized in the step-size algorithm of the vector. Obviously, the method will calculate \( v_{n+1}, v'_{n+1} \) and \( v''_{n+1} \) to approximate \( v(x_{n+1}), v'(x_{n+1}) \) and \( v''(x_{n+1}) \), where \( v_{n+1} \) is the measured solution and \( v(x_{n+1}) \) is an accurate solution.

2. CONSTRUCTION OF THE METHODS

The order conditions of RKTGD method are derived up to order six by using the Taylor expansion and Maple package. For the new system, the local truncation errors or the order terms for \( m \)-stage up to order six can be expressed (see \([9]\)). The construction of the embedded Runge-Kutta method is an active research area producing continuous improvements to the existing codes. We are concerned with the derivation of \( p(q) \) pairs of explicit RKTGD methods which are used in variable step size codes to provide a cheap error estimation. The following simplifying assumption is used to minimize the number of equations to solve:

\[ \sum_{j=1}^{m} \bar{a}_{ij} = \frac{c_i^2}{2} \]  

To develop efficient pairs, the following techniques are used:

(a) For both the higher and lower order RKTGD formulas, the \( \| r^{(p+1)}_q \|_2 \) and \( \| \xi^{(p+1)}_q \|_2 \) quantities should be as minimal as possible, where

\[
\| r^{(p+1)}_q \|_2 = \sqrt{\sum_{i+1}^{n+1} (r_{ij}^{(p+1)})^2 + \sum_{i+1}^{n+1} (\bar{r}_{ij}^{(p+1)})^2 + \sum_{i+1}^{n+1} (\beta_{ij}^{(p+1)})^2},
\]

and
\[ \| \tau^{(q+1)} \|_2 = \sqrt{\sum_{i+1}^{n+1} (\tau_i^{(q+1)})^2 + \sum_{i+1}^{n+1} (\tau_i^{(q+1)})^2 + \sum_{i+1}^{n+1} (\tau_i^{(q+1)})^2}, \]

Where \( \tau^{(p+1)} \), \( \tau^{(p+1)} \) and \( \tau^{(p+1)} \) are the local truncation error norms for \( y', y'' \) and \( y''' \) respectively, \( \tau^{(p+1)} \) is the global local truncation error.

(b) The calculation of the local error at point \( x_{n+1} \) is determined using the formula:

\[ LTE = \max \{ \| \delta_{n+1}^{(y)} \|_\infty, \| \delta_{n+1}^{(y )} \|_\infty, \| \delta_{n+1}^{(y'' )} \|_\infty \}, \]

where

\[ \delta_{n+1} = \tilde{y}_{n+1} - y_{n+1}, \]
\[ \delta'_{n+1} = \tilde{y}'_{n+1} - y'_{n+1}, \]
\[ \delta''_{n+1} = \tilde{y}''_{n+1} - y''_{n+1}. \]

The solutions using the higher order formula are \( y_{n+1}, y'_{n+1} \) and \( y''_{n+1} \), while the solutions using the lower order formula are \( \tilde{y}_{n+1}, \tilde{y}'_{n+1} \) and \( \tilde{y}''_{n+1} \). These local error estimates (LTE) can be used by the standard formula as given in [17] to control the step size \( h \),

\[ h_{n+1} = 0.9h_{n} \left( \frac{LTE}{\tau_{g}} \right), \]

where 0.9 is considered to be a safety factor and represents the estimate of local error at each step. \( Tol \) represents the necessary accuracy, which is the maximum permissible local error. If \( LTE \leq Tol \), local extrapolation (or higher-order mode) method will be accepted; this means that a more precise approximation will be used to advance the integration and \( h \) would be updated using the formula (8).

On the other hand, if \( LTE > Tol \), the step will be rejected and the step-size would be halved.

3. The Derivation of 4(3) pair

A three-stage RKTG technique of order four is derived in this section. The conditions of the algebraic order up to order four are used (see [9]) and the simplifying assumption (7). Then, the system of equations can be reduced to 13 nonlinear equations with 17 unknowns and left with four degrees of freedom by simplifying assumptions. The solution of the equation system results simultaneously in a solution in terms of four free parameters, which are \( a_3, a_2, a_1 \) and \( b_2 \):

\[ a_{21} = \frac{9 - 24c_3 + 16c_3^2}{8(c_3^3 - 12c_3^2 + 4)}, \quad a_{31} = \frac{c_3(9 - c_3^2 - 20c_3^2 + 14c_3 - 3)}{-3 + 4c_3}, \]
\[ a_{32} = -\frac{c_3(18 - c_3^2 - 36c_3^2 + 25c_3 - 6)}{-3 + 4c_3}, a_{21} = a_{21}, a_{31} = a_{31}, \]
\[ a_{32} = -\frac{-12a_{31} + 16a_{31}c_3 + 423a_{21}c_3^4 - 864a_{21}c_3^3 + 567a_{12}c_3^2 - 128a_{21}c_3}{-12 + 16c_3}, \]
\[ + \frac{50c_3^3 - 36c_3^2 + 9c_3 - 24c_3^2}{-12 + 16c_3}, \]
\[ b_1' = \frac{6c_3^2 - 6c_3 + 1}{8(-3 + 4c_3)} + \frac{1}{c_3}, b_2' = \frac{3c_3 + 2 + 48b_2c_3 - 36b_2}{24c_3(3c_3 - 2)}, b_3' = \frac{-1 + c_3}{6c_3(3 - 8c_3^2 + 6c_3^2)}, c_1 = 0, \]
\[ c_2 = \frac{1}{2(3c_3 - 2)}, \]
\[ b_2 = b_2, b_1 = \frac{-12c_3^2 + 11c_3 + 72b_2c_3^2 - 96b_2c_3 - 2 + 36b_2}{24c_3(3c_3 - 2)}, \]
\[ b_3 = \frac{-3c_3 + 2 + 48b_2c_3 - 36b_2}{24c_3(3c_3 - 2)}, \]
\[ b_1'' = \frac{6c_3^2 - 6c_3 + 1}{8(-3 + 4c_3)} + \frac{1}{c_3}, b_2'' = \frac{2(7c_3^2 - 54c_3^2 + 36c_3 - 8)}{3(-50c_3^2 + 36c_3 - 9 + 24c_3^2)}, b_3'' = \frac{1}{6c_3(3 - 8c_3 + 6c_3^2)}, \]

and setting \( c_3 = 1 \) and \( a_{31} = 0 \) gives:
By using the `minimize` command in Maple, then \( a_{21} = 0.0325000022021653 \) and \( b_2 = 0.100000024350969 \), and the minimum value \( \| \tau^{(5)}_g \|_2 \) is 0.0586320476692477264. Again for the optimized values, \( a_{21} = \frac{3}{100} \) and \( b_2 = \frac{1}{10} \) were chosen.

Therefore, the coefficients denoted by RKTGD4 of the three-stage fourth-order RKTGD method can be defined as follows:

\[
\begin{align*}
    c_2 &= \frac{1}{2}, c_3 = 1, a_{21} = \frac{3}{100}, a_{32} = \frac{13}{100}, a_{21} = \frac{1}{8}, a_{32} = \frac{1}{2}, b_1 = \frac{3}{40}, b_2 = \frac{1}{10} \\
    b_3 &= -\frac{1}{120}, b'_1 = \frac{1}{6}, b'_2 = \frac{1}{3}, b'_3 = 0, b''_1 = \frac{1}{6}, b''_2 = \frac{2}{3}, b''_3 = -\frac{1}{6}
\end{align*}
\]

A three-stage \((s=3)\) third-order \((q=4)\) embedded formula can now be derived considering the above solutions for \( A \) and \( c \). Solving the third-order equations for \( v(t) \), \( v'(t) \) (see [9]) simultaneously yields a solution for \( b_i \) in \( b_i \) terms, while \( b'_i \) and \( b''_i \) are the same as the fourth-order formula. Three free parameters of the solution in terms of \( b'_3 \), \( b_2 \) and \( b_3 \) are obtained as follows:

\[
\begin{align*}
    b'_1 &= \frac{1}{6} - b_2 - b_3, b_2 = b_2, b_3 = b_3, b'_1 = \frac{1}{6} + b'_3, \quad b'_2 = \frac{1}{3} - 2 b'_3, \quad b'_3 = b'_3, \\
    b''_1 &= \frac{1}{6}, b''_2 = \frac{2}{3}, b''_3 = \frac{1}{6}
\end{align*}
\]

Here, take \( b'_3 = \frac{1}{16} \) then:

\[
\| \tau^{(4)}_g \|_2 = \frac{1}{120} \sqrt{3600 b_2^2 + 14400 b_2 b_3 - 600 b_2 + 14400 b_3^2 - 1200 b_3 + 45}
\]

Two free parameters \( b_2 \) and \( b_3 \) do not appear in the solution now, but for fourth-order conditions, they appear in the local truncation error. According to Dormand [9], the free parameters \( b_2 \) and \( b_3 \) can be determined by minimizing the local truncation error norms \( \| \tau^{(4)}_g \|_2 \). By using the `minimize` command in Maple, \( b_2 = 0.416666666666741 \), \( b_3 = -0.166666666666519 \) were obtained, and the minimum error is 0.0372677996249964963. For the optimized value in fractional form, \( b_2 = \frac{4}{10} \) and \( b_3 = -\frac{2}{10} \) were chosen.

Finally, all the three-stage embedded RKTGD system coefficients denoted by RKTGD4(3) can be written in the following way (see Table 1):

**Table I.** The RKTGD4(3) Method.
4. The Derivation of (5(4)) pair

The four-stage RKTGD method of order five is derived in this section, and the algebraic conditions (see [9]) and assumption simplification (7) are used. Then, 31 nonlinear equations consist of the resulting system of equations with 24 unknown variables to be solved. Using the simplified assumption (7), the system of equations is reduced to 20 equations with 24 unknowns and 4 degrees of freedom remaining. Solving the system and the solution family simultaneously in the form of \(a_{41}, a_{42}, a_{43}\) and \(c_4\) gives the following:

\[
\bar{a}_{21} = \frac{c_4^2}{2(4c_4 - 1)^2}, \quad \bar{a}_{31} = -\frac{2(-71c_4^2 + 48c_4 - 16 + 20c_4^3)}{125c_4(10c_4^2 - 12c_4 + 3)}, \quad \bar{a}_{32} = \frac{125c_4(10c_4^2 - 12c_4 + 3)}{2((-4 + 5c_4)(11c_4 - 4)(4c_4 - 1))}
\]

\[
\bar{a}_{41} = \frac{30c_4^3 - 59c_4^2 + 40c_4 - 8}{4}, \quad \bar{a}_{42} = \frac{c_4(38c_4 - 59c_4^2 - 8 + 30 30c_4^3)(4c_4 - 1)}{11c_4 - 4}, \quad a_{21} = \frac{a_{41} + a_{42} + a_{43}}{16c_4^2 - 8c_4 + 1)(4c_4 - 1)},
\]

\[
a_{31} = \frac{-50a_{41}c_4^2 - 226a_{43}c_4^3 + 20a_{42}c_4 + 108a_{43}c_4 - 16a_{43} + 16c_4^2 + 87c_4^3}{625(10c_4^2 - 12c_4 + 3)(-1 + 2c_4)c_4^3}
\]

\[
+ \frac{105a_{42}c_4^2 + 100a_{43}c_4^3 + 100a_{42}c_4^3 + 73c_4^3 - 110c_4^6}{625(10c_4^2 - 12c_4 + 3)(-1 + 2c_4)c_4^3} \quad a_{32} = \frac{2(11c_4 - 4)(-40c_4^5 + 62c_4^4 - 29c_4^3 + 4c_4^2 + 16a_{43}c_4 + 5a_{43}c_4 - 4a_{43}}{652(10c_4^2 - 12c_4 + 3)(-1 + 2c_4)c_4^3},
\]

\[
b_1 = \frac{-17c_4^3 + 1056b_2c_4^3 + 120c_4^2 - 912b_2c_4^2 - 27c_4 + 192c_4b_2}{5(72c_4^2 - 21c_4 + 2 - 240b_2c_4^2 - 80c_4^3 + 480b_2c_4^3)},
\]

\[
b_3 = \frac{96c_4(4c_4 - 1)}{(4c_4 - 1)^2(-4 + 5c_4)}, \quad b_4 = \frac{-16c_4^2 + 8c_4 - 1 + 132b_2c_4^2 - 48c_4b_2}{(4c_4 - 1)^2(-4 + 5c_4)}, \quad b_5'' = \frac{2c_4^2 + 4c_4 + 1}{48c_4^3},
\]

\[
b_2 = \frac{(4c_4 - 1)^2(16c_4^2 + 8c_4 - 1)}{24c_4^2(-1 + 2c_4)(11c_4 - 4)}, \quad b_3'' = \frac{125(10c_4^2 - 12c_4 + 3)}{48(-4 + 5c_4)(11c_4 - 4)}, \quad b_4'' = \frac{24c_4^2(-1 + 2c_4)(-4 + 5c_4)}{5296c_4^3}.
\]

Letting \(a_{41} = a_{42} = a_{43} = 0\) and \(c_4 = \frac{2}{3}\), the local truncation error in one free parameter is given by:
By using the \texttt{minimize} command in Maple, \( b_2 = 0.129642210144928 \) was obtained, and the minimum error is 0.0391272028545436926. For the optimized value in fractional form, \( b_2 = \frac{1}{10} \) was chosen.

Finally, all the coefficients of four-stage fifth-order RKT method denoted by RKTG5 can be written as follows:

\[
c_2 = \frac{2}{5}, c_3 = \frac{4}{5}, c_4 = \frac{2}{3}, a_{21} = 0, a_{31} = \frac{4}{124}, a_{32} = \frac{4}{125}, \bar{a}_{21} = \frac{2}{25}, \bar{a}_{31} = \frac{4}{25},
\]

\[
b_1 = \frac{23}{192}, b_2 = \frac{25}{64}, b_3 = \frac{25}{192}, b_4 = -\frac{9}{64}, b_{1}' = \frac{23}{192}, b_{2}' = \frac{125}{192}, b_{3}' = \frac{125}{192}, b_{4}' = -\frac{27}{64}.
\]

A three-stage (s=3) fourth-order (q=4) embedded formula may now be derived based on the above solution for values of A and c. Solving the third and fourth-order equations for \( v \) (see [9]), so that the resulting equation system consists of 2 nonlinear equations with 4 variable unknowns to be solved. Solving the method at the same time gives:

\[
b_1 = b_1, b_2 = \frac{25}{96} + \frac{1}{2} b_3, b_3 = b_3, b_4 = \frac{3}{2} b_1 - \frac{3}{2} b_3 - \frac{3}{32}.
\]

Choosing \( b_1 = \frac{1}{20} \), another free parameter \( b_3 \) was found by minimize error norm \( || y_n^{(4)} ||_2 \).

where

\[
|| y_n^{(4)} ||_2 = \frac{1}{3000} \sqrt{2520 - 12800 b_3 + 6264 b_3^2}
\]

Here, one free parameter \( b_3 \) did not appear in the solution but it appeared in the local truncation error. According to Dormand [11]. The free \( b_3 \) parameter can be calculated by minimizing the local truncation error and obtaining \( b_3 = \frac{1}{10} \) by using Maple. For \( v' \), the equation of order-four were solved and simplifying assumption (7) was used. The simplifying assumption reduces the equation system to 3 nonlinear equations with 4 unknowns, simultaneously solving the system gives:

\[
s_1 = b_1, s_2 = \frac{95}{96} - 5 b_1, b_3 = -5 b_1 + \frac{35}{48}, b_4 = 9 b_1 - \frac{30}{32}
\]

where

\[
|| y_n^{(4)} ||_2 = \frac{1}{9000} \sqrt{155302 - 1405056 b_1 + 9326592 b_1^2}
\]

It can be noticed that there is one free parameter \( b_1 \), using the minimize command in Maple, then \( b_1 = 0.0753252635046047 \) was obtained and for optimized value, \( b_1 = \frac{7}{100} \) was chosen. While for \( v_0 \), the equations until order four were solved, and using simplifying assumption to reduce the system of equations to 4 nonlinear equations with 4 unknowns; then \( \tilde{b}_1 \) is the same as the fifth-order formula. Finally, all the coefficients of four-stage embedded RKT method denoted by RKTGD5(4) can be written as follows (see Table II):

Table II. The RKTGD5(4) Method.
5. Numerical Experiments

In this section, some of the issues involving $v''' = f(x, v, v')$ are tested. When the same set of problems are reduced to a system of first-order equations and solved using the existing RK pairs of the same order, comparisons are made between the numerical results and the results obtained.

- RKTGD4(3): In this paper, the Runge-Kutta type 4(3) pair derived.
- RKTGD5(4): In this paper, the Runge-Kutta type 5(4) pair derived.
- RK4(3)Z: 4(3) pair of Runge-Kutta, provided by Hairer et. al [18].
- RK4(3)M: Merson-derived pair type 4(3) of Runge-Kutta as in Hairer et. al [18].
- RK4(3)D: Dormand-derived Runge-Kutta Type 4(3) pair [11].
- RK4(3)F: Fehlberg-derived Runge-Kutta Type 4(3) pair [19].
- RKB5(4): Pair of Runge-Kutta 5(4) introduced in Butcher [20].
- RKF5(4): Fehlberg-derived Runge-Kutta 5(4) pair as given in [19].
- DOPRI5(4): Pair of Runge-Kutta 5(4) with FSAL properties obtained from Dormand and Prince [12].

Problem 1. (Homogeneous Linear Problem)

$$y'''(x) = -y(x) + 2y'(x), y(0) = 0, y'(0) = 1, y''(0) = 1,$$

Theoretical solution:

$$y(x) = 2e^x - (-1 + \frac{2}{3} \sqrt{5})e^{\frac{1}{2}(\sqrt{5}+1)x} + (-1 + \frac{2}{3} \sqrt{5})e^{-\frac{1}{2}(\sqrt{5}+1)x}.$$

Problem 2. (Linear System)

$$y''(x) = \frac{1}{2} e^{3x} y_3(x) y'_2(x), \quad y'''(x) = \frac{8}{3} e^{2x} y_1(x) y'_3(x), \quad y''(x) = 27 y_2(x) y'_1(x)$$

$$y_1(0) = 1, y'_1(0) = -1, y''_1(0) = 1, \quad y_2(0) = 1, y'_2(0) = -2, y''_2(0) = 4, \quad y_3(0) = 1, y'_3(0) = -3, y''_3(0) = 9,$$

Theoretical solution:

$$y_1(x) = e^{-x}, \quad y_2(x) = e^{2x}, \quad y_3(x) = e^{3x}.$$

Problem 3. (Homogeneous Non-Linear Problem)

$$y'''(t) = \frac{3y'(t)}{4(y(t))^4}$$

$$y(0) = 1, y'(0) = \frac{1}{2}, y''(0) = \frac{1}{4},$$

Theoretical solution:

$$y(x) = \sqrt{x + 1}.$$
Comparison between RKTGD4(3), RK4(3)Z, RK4(3)M, RK4(3)D and RK4(3)F with respect to problems 1 to 3 was made and results are displayed in Figures 1 to 3.

**Figure 1.** Comparison for RKTGD4(3), RK4(3)Z, RK4(3)M, RK4(3)D and RK4(3)F problem 1 with $X_{end}=5$

**Figure 2.** Comparison for RKTGD4(3), RK4(3)Z, RK4(3)M, RK4(3)D and RK4(3)F problem 2 with $X_{end}=2$
Figure 3. Comparison for RKTGD4(3), RK4(3)Z, RK4(3)M, RK4(3)D and RK4(3)F problem 3 with $X_{end}=10$

Comparison between RKTGD5(4), RK5(4)B, RK5(4)F, and DOPRI5(4) with respect to problems 1 to 3 was made and results are displayed in Figures 4 to 6.

Figure 4. Comparison for RKTGD5(4), RK5(4)B, RK5(4)F, and DOPRI5(4) problem 1 with $X_{end}=5$

Figure 5. Comparison for RKTGD5(4), RK5(4)B, RK5(4)F, and DOPRI5(4) problem 2 with $X_{rnd} = 3$. 

Figure 6. Comparison for RKTGD5(4), RK5(4)B, RK5(4)F, and DOPRI5(4) problem 3 with $X_{end}=10$. 
Figure 6. Comparison for RKTGD5(4), RK5(4)B, RK5(4)F, and DOPRI5(4) problem 3 with $X_{end}=10$.

6. Conclusion
Two pairs of embedded RKTGD methods for directly solving third-order ODEs using the variable step size codes have been suggested and discussed in this study. The methods are called, respectively, RKTGD4(3) and RKTGD5(4). Step-size codes of variables based on the methods were created and used to solve the third-order ODEs. The comparison was made with the Runge-Kutta methods of the current single step, as other direct methods are multi-step in nature. Numerical outcomes of the same RKTGD and RK pairs of the same order were also compared. The results showed that the RKTGD4(3) is more efficient than the Runge–Kutta 4(3) pairs like the RK4(3)Z method (Hairer et al [18]), RK4(3)M (Merson [18]), RK4(3)D (Dormand [11]), and RK4(3)F (Fehlberg [19]) for all the problems tested. This new pair is more efficient for RKTGD5(4) compared to RK5(4)B (Butcher [20]), RK5(4)F (Fehlberg as in [19]), and RK5(4)D given by Dormand and Prince [12]. It has been shown from the numerical results that, compared to the existing embedded RK pairs of the same order in the literature, the new methods are more efficient. Hence, it can be concluded that in solving third order ODEs, the RKTGD pairs are computationally more efficient.

References
[1] T G Myers 1998 Thin films with high surface tension SIAM Rev 40 (3) pp 441-462
[2] E Momoniat, Symmetries 2009 first integrals and phase planes of a third-order ordinary differential equation from thin film flow Math. Comput. Model 49 (1-2) pp 215-225
[3] B R Duffy, S K Wilson 1997 A third-order differential equation arising in thin-film flows and relevant to Tanner’s law Math. Lett. 10 (3) pp 63-68
[4] N Waeleh, Z A Majid, F Ismail 2011 A new algorithm for solving higher order IVPs of ODEs Appl. Math. Sci. 5 (56) pp 2795-2805
[5] D O Awoyemi and O M Idowu 2005 A class of hybrid collocation methods for third-order ordinary differential equations International Journal of Computer Mathematics 82 (10) pp1287-1293
[6] F A Fawzi, N Senu, F Ismail and Z A Majid 2018 A New Integrator of Runge-Kutta Type for Directly Solving General Third-order ODEs with Application to Thin Film Flow Problem Applied Mathematics & Information Sciences 12 (4) pp 775-784
[7] S N Jator 2011 Solving second order initial value problems by a hybrid multi-step method without predictors Appl. Math. Comput 217 (8) pp 4036-4046
[8] F A F Alshareeeda 2017 Runge-Kutta type methods for solving third-order ordinary differential equations and first-order oscillatory problems Ph.D. Thesis Universiti Putra Malaysia
[9] F A Fawzi and N Senu and F Ismail 2018 An efficient of direct integrator of Runge-Kutta type method for solving $y''' = f(x,y,y')$ with application to thin film flow problem international journal of pure and applied mathematics 120 pp 27-50

[10] W Gander and D Gruntz 1999 Derivation of numerical methods using computer algebra SIAM Review 41 (3) pp 577-593

[11] J R Dormand 1996 Numerical Methods for Differential Equations A Computational Approach, CRC Press, Boca Raton, Fla, USA

[12] J R Dormand, P J Prince 1980 A family of embedded Runge-Kutta formulae J. Comput. Appl. Math 6 (1) pp 19-26

[13] F A Fawzi, N Senu, F Ismail and Z A Majid 2016 An embedded 6 (5) pair of explicit Runge-Kutta method for periodic ivps Far East Journal of Mathematical Sciences 100 (11) pp 1841

[14] J R Dormand, P J Prince 1981 High order embedded Runge-Kutta formulae J. Comput. Appl. Math 7 (1) pp 67-75

[15] N Senu, M Mechee, F Ismail and Z Siri 2014 Embedded explicit Runge-Kutta type methods for directly solving special third order differential equations $y''' = f(x,y)$ Applied Mathematics and Computation 240 (2014) pp 33-43

[16] M El-Mikkawy, R El-Desouky 2003 A new optimized non-FSAL embedded Runge-Kutta-Nyström algorithm of orders 6 and 4 in six stages Applied Mathematics and Computation 145 pp 33-43

[17] N Senu, M Suleiman, F Ismail 2009 An embedded explicit Runge-Kutta-Nyström method for solving oscillatory problems Physica Scripta 80 pp 015005

[18] E Hairer, S P Nørsett, G Wanner 1993 Solving Ordinary Differential Equations I: Nonstiff Problem 8 Springer Berlin Germany 2nd edition

[19] E Fehlberg 1969 Lower order classical Runge-Kutta formulas with step-size control and their application to some heat transfer problems NASATR R-315

[20] J C Butcher 2008 Numerical Methods for Ordinary Differential Equations John Wiley & Sons Chichester UK 2nd edition