DIRECT ONE-STEP METHOD FOR SOLVING
THIRD-ORDER BOUNDARY VALUE PROBLEMS

Athraa Abdulsalam\textsuperscript{1,3}, Norazak Senu\textsuperscript{1,2}, Zanariah Abdul Majid\textsuperscript{1,2}

\textsuperscript{1}Department of Mathematics
Universiti Putra Malaysia
43400 UPM Serdang, Selangor, MALAYSIA
\textsuperscript{2}Institute for Mathematical Research
Universiti Putra Malaysia
43400 UPM Serdang, Selangor, MALAYSIA
\textsuperscript{3}Department of Mathematics
University of Baghdad
Al-Jadrriya, Baghdad, IRAQ

Abstract: A direct explicit Runge-Kutta type (RKT) method via shooting technique to approximate analytical solutions to the third-order two-point boundary value problems (BVPs) with boundary condition type I and II are proposed. In this paper first, a three-stage fourth-order direct explicit Runge-Kutta type method denoted as RKT3s4 is constructed. A new algorithm of shooting technique for solving two-point BVPs for third-order ordinary differential equations (ODEs) is presented.

AMS Subject Classification: 65L05, 65L06, 65L20
Key Words: direct Runge-Kutta method, two-point boundary value problem, shooting technique, third-order ordinary differential equation

Received: August 16, 2018 © 2019 Academic Publications

\textsuperscript{§}Correspondence author
1. Introduction

Numerous problems in mathematics can be formulated in the form of differential equations, an initial value problem (IVP) is an ordinary differential equation (ODE) whose boundary conditions are specified at a single point, which can be found in mathematical modeling of real-life problems [1]-[3]. There is also another class of the ODE which is the boundary value problem (BVP), a BVP differs from an IVP in that the boundary conditions are specified at more than one point and in that solutions of the differential equation over an interval, satisfying the boundary conditions at the endpoints, are required ([4], p.1). BVP arises in several branches of engineering and applied sciences including fluid dynamics and chemical reactions, elastic beams, etc. (see [5], p.7 - p.27).

In this work, we concentrate on finding an approximate solution to two-point BVPs with boundary conditions type I and II of the form

\[ u''' = f(t,u), \quad a < t < b, \]

with boundary conditions:

(a) Type I

\[ u(a) = \alpha, \quad u'(a) = \beta, \quad u(b) = \gamma. \]

(b) Type II

\[ u(a) = \alpha, \quad u'(a) = \beta, \quad u'(b) = \lambda. \]

where \( a, b, \alpha, \beta, \gamma, \) and \( \lambda \) are constants, the proof of existence and uniqueness of solutions to two-point BVPs of third-order ODE is possible, see Henrici [6]. For analytical solutions of the IVPs and BVPs, analytical methods are seldom used since most of the problems encountered were difficult, with either complex differential equations or complex boundary conditions ([7], [8]) or sometimes finding an analytical solution to some ODE applications is complicated or impossible, therefore recourse to numerical methods in such cases is almost the only choice. For several years, various numerical methods have been derived to handle the IVPs and BVPs, which is a subject can be treated separately. The numerical procedures for the solution of the IVP can be classified into two major groups: one-step methods and multi-step methods. One of the advantages of one-step methods is that it can change the step size easily at different \( t \) ([4], p.10). Two such methods of the one-step are Taylor’s method and Runge-Kutta (RK) method. The numerical algorithm of the RK method is considered the most widely used scheme, due to its low truncation error ([4], p.8). For the
multi-step, such methods are Predictor-Corrector methods, Adams-Moulton method, Adams-Bashforth method, and Trapezium rule method. Similarly, for the numerical procedures for the solution of the BVP, there exist several techniques, such as Finite-Difference method ([9], [10]), Shooting method ([5], [11]), the quasilinearization method ([12], [13]), the monotone iterative method ([14], [15]), and the variational iteration method ([16], [17]). Burden and Faires [1] have used RK method (after transferring it into the system of first-order) via shooting technique to solve second-order two-point linear BVP with Dirichlet boundary condition. However, many researchers ([18], [19]) have shown that using direct RK approach to solve higher-order ODE without reducing it first to a system of first-order is superior and more efficient than the conventional RK. Where there is no need to increase the number of equations and calculating more function evaluations which lead to a time-consuming process and more human effort as in classical RK.

Therefore, the purpose of this study is to construct a direct and effective method of Runge-Kutta type with less computation time and function evaluations to solve two-point BVPs of third-order with boundary conditions type I and II, algorithm of shooting technique was offered to develop the approximate analytical solutions.

The organizing of this paper is as follows: In Section 2, the construction of the explicit RKT3s4 method is presented. The explanation of the new shooting technique algorithm is given in Section 3. In Section 4, four problems numerically examined the efficiency of the RKT3s4 method as compared to the existing method and the last section, deals with the conclusions.

2. Construction of the Explicit RKT3s4 Method

In this section, an explicit three-stage RKT method of order-four will be derived. The general $\mu$-stage RKT method for the differential equation

$$u''' = f(t, u(t))$$

given by

$$u_{n+1} = u_n + hu'_n + \frac{h^2}{2}u''_n + h^3\sum_{i=1}^{\mu} b_i \kappa_i,$$  \hspace{1cm} (5)

$$u'_{n+1} = u'_n + hu''_n + h^2\sum_{i=1}^{\mu} \hat{b}_i \kappa_i,$$  \hspace{1cm} (6)
\[ u''_{n+1} = u''_n + h \sum_{i=1}^{\mu} \hat{b}_i \kappa_i, \tag{7} \]

where

\[ \kappa_1 = f(t_n, u_n), \]
\[ \kappa_i = f(t_n + c_i h, u_n + c_i h u'_n + \frac{h^2}{2} c_i^2 u''_n + h^3 \sum_{j=1}^{\mu} a_{ij} \kappa_j), \tag{8} \]

for \( i = 2, 3, \ldots, \mu \), where \( c_i, a_{ij}, b_i, \hat{b}_i, \) and \( \hat{\hat{b}}_i \) for \( i = 1, 2, \ldots, \mu \) and \( j = 1, 2, \ldots, \mu \) are the parameters of the RKT method and they supposed to be real. RKT method is said to be explicit when \( a_{ij} = 0 \) for \( i \leq j \), otherwise it is an implicit method. RKT method (5)-(8) can be written by the well-known Butcher tableau as follows (see Table 1):

<table>
<thead>
<tr>
<th>( c_i )</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_2 )</td>
<td>( a_{21} )</td>
</tr>
<tr>
<td>( c_3 )</td>
<td>( a_{31} )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( c_\mu )</td>
<td>( a_{\mu,1} )</td>
</tr>
<tr>
<td>\hline</td>
<td></td>
</tr>
<tr>
<td>( b_1 )</td>
<td>( b_2 )</td>
</tr>
<tr>
<td>( \hat{b}_1 )</td>
<td>( \hat{b}_2 )</td>
</tr>
<tr>
<td>( \hat{\hat{b}}_1 )</td>
<td>( \hat{\hat{b}}_2 )</td>
</tr>
</tbody>
</table>

Table 1: \( \mu \)-stage explicit RKT method.

To derive the RKT3s4 method, we will use the order conditions that have given by Mechee et al. [20] up to fifth-order for \( u, u', \) and \( u'' \). The order conditions of \( \mu \)-stage RKT method up to fifth-order as given in [20] are given as follows.

The order conditions for \( u \):

Order 3:
\[ \sum b_i = \frac{1}{6}, \tag{9} \]

Order 4:
\[ \sum b_i c_i = \frac{1}{24}, \tag{10} \]
Order 5:

\[ \sum b_i c_i^2 = \frac{1}{60}. \] (11)

The order conditions for \( u' \):

Order 2:

\[ \sum \hat{b}_i = \frac{1}{2}, \] (12)

Order 3:

\[ \sum \hat{b}_i c_i = \frac{1}{6}, \] (13)

Order 4:

\[ \sum \hat{b}_i c_i^2 = \frac{1}{12}, \] (14)

Order 5:

\[ \sum \hat{b}_i c_i^3 = \frac{1}{20}, \sum \hat{b}_i a_{ij} = \frac{1}{120}. \] (15)

The order conditions for \( u'' \):

Order 1:

\[ \sum \hat{b}_i = 1, \] (16)

Order 2:

\[ \sum \hat{b}_i c_i = \frac{1}{2}, \] (17)

Order 3:

\[ \sum \hat{b}_i c_i^2 = \frac{1}{2}, \] (18)

Order 4:

\[ \sum \hat{b}_i c_i^3 = \frac{1}{4}, \sum \hat{b}_i a_{ij} = \frac{1}{24}, \] (19)

Order 5:

\[ \sum \hat{b}_i c_i^4 = \frac{1}{5}, \sum \hat{b}_i a_{ij} c_j = \frac{1}{120}, \sum \hat{b}_i a_{ij} c_i = \frac{1}{30}. \] (20)

Now, assume that \( c_3 = 1 \), as a result, we will have a system of nonlinear equations, consisting of ten nonlinear equations with fourteen unknown variables that have not yet been resolved, as follows:

\[ b_1 + b_2 + b_3 = \frac{1}{6}, \] (21)

\[ b_2 c_2 + b_3 c_3 = \frac{1}{24}, \] (22)

\[ \hat{b}_1 + \hat{b}_2 + \hat{b}_3 = \frac{1}{2}. \] (23)
\[ \hat{b}_2 c_2 + \hat{b}_3 c_3 = \frac{1}{6}, \] (24)
\[ \hat{b}_2 c_2^2 + \hat{b}_3 c_3^2 = \frac{1}{12}, \] (25)
\[ \hat{b}_1 + \hat{b}_2 + \hat{b}_3 = 1, \] (26)
\[ \hat{b}_2 c_2 + \hat{b}_3 c_3 = \frac{1}{2}, \] (27)
\[ \hat{b}_2 c_2^2 + \hat{b}_3 c_3^2 = \frac{1}{3}, \] (28)
\[ \hat{b}_2 c_2^3 + \hat{b}_3 c_3^3 = \frac{1}{4}, \] (29)
\[ \hat{b}_2 a_{2,1} + \hat{b}_3 a_{3,1} + \hat{b}_3 a_{3,2} = \frac{1}{24}. \] (30)

Accordingly, the system has a solution based on three free parameters \( b_2, a_{2,1}, \) and \( a_{3,1} \) as follows:

\[ \hat{b}_1 = \frac{1}{6}, \] (31)
\[ \hat{b}_2 = \frac{2}{3}, \] (32)
\[ \hat{b}_3 = \frac{1}{6}, \] (33)
\[ a_{3,2} = -4 a_{2,1} - a_{3,1} + \frac{1}{4}, \] (34)
\[ b_1 = \frac{1}{8} - \frac{1}{2} b_2, \] (35)
\[ b_3 = -\frac{1}{2} b_2 + \frac{1}{24}, \] (36)
\[ c_2 = \frac{1}{2}, \] (37)
\[ c_3 = 1, \] (38)
\[ \hat{b}_1 = \frac{1}{6}, \] (39)
\[ \hat{b}_2 = \frac{1}{3}, \] (40)
\[ \hat{b}_3 = 0. \] (41)

Based on Dormand [21], the free parameters are chosen by minimizing the error equations. The global error of the fifth-order conditions is defined as
Table 2: Butcher tableau for RKT3s4 method.

\[
\begin{array}{cccc}
0 & 1 & 2 & 140 \\
1 & 1/2 & 10 & 0 \\
3 & 1/3 & 10 & -1/120 \\
1/6 & 1/3 & 0 \\
1/6 & 2/3 & 1/6 \\
\end{array}
\]

3. Shooting Method for Linear BVP

Shooting technique is used to convert the BVP to IVPs. The idea of shooting technique is to obtain the missing initial value until the boundary condition at
the other end converges to its correct value. When we use the shooting method, we transform (1) into IVP of the form

\[ u''' = f(t, u), \quad a < t < b, \]
\[ u(a) = \alpha, \quad u'(a) = \beta, \quad u''(a) = \lambda_1, \]

where \( \lambda_1 \) is any number. Then the resulting IVP will be solved using RKT method.

**Reduction to Three IVPs:**

The solution of a linear two-point BVP is associated with the formation of a linear combination of the solutions to three IVPs.

The form of the IVPs as follows.

Suppose that \( \psi(t) \) is the unique solution to the IVP

\[ \psi''' = f_1(t, \psi), \]
\[ f_1(t, \psi) = q(t)\psi(t) + g(t), \quad \text{with} \quad \psi(a) = \alpha, \quad \psi'(a) = 0, \quad \psi''(a) = 0. \] (44)

Suppose that \( \rho(t) \) is the unique solution to the IVP

\[ \rho''' = f_2(t, \rho), \]
\[ f_2(t, \rho) = q(t)\rho(t), \quad \text{with} \quad \rho(a) = 0, \quad d\rho'(a) = 1, \quad \rho''(a) = 0. \] (45)

Suppose that \( \vartheta(t) \) is the unique solution to the IVP

\[ \vartheta''' = f_3(t, \vartheta), \]
\[ f_3(t, \vartheta) = q(t)\vartheta(t), \quad \text{with} \quad \vartheta(a) = 0, \quad \vartheta'(a) = 0, \quad \vartheta''(a) = 1. \] (46)

Then the linear combination

\[ u(t) = \psi(t) + \theta_1 \rho(t) + \theta_2 \vartheta(t), \] (47)

is a solution to the BVP (1).

For the boundary condition type I, the solution \( u(t) \) in equation (47) takes on the boundary values. Then the linear combination

\[ u(a) = \psi(a) + \theta_1 \rho(a) + \theta_2 \vartheta(a), \] (48)
\[ u(a) = \alpha. \] (49)
\[ u'(a) = \psi'(a) + \theta_1 \rho'(a) + \theta_2 \vartheta'(a), \] (50)
\[ u'(a) = \theta_1. \]  
\[ u(b) = \psi(b) + \theta_1 \rho(b) + \theta_2 \vartheta(b). \]  

Imposing the boundary conditions \( u'(a) = \beta \) and \( u(b) = \gamma \) in (51) and (52) produces \( \theta_1 = \beta \) and \( \theta_2 = \frac{\gamma - \psi(b) - \beta \rho(b)}{\vartheta(b)} \). Therefore, if \( \vartheta(b) \neq 0 \), the unique solution of the two-point BVP (1) with boundary condition type I is given by:

\[ u(t) = \psi(t) + \beta \rho(t) + \frac{\gamma - \psi(b) - \beta \rho(b)}{\vartheta(b)} \vartheta(t). \]  

For the boundary condition type II, the solution \( u(t) \) in equation (47) takes on the boundary values. Then the linear combination

\[ u(a) = \psi(a) + \theta_1 \rho(a) + \theta_2 \vartheta(a), \]  
\[ u(a) = \alpha, \]  
\[ u'(a) = \psi'(a) + \theta_1 \rho'(a) + \theta_2 \vartheta'(a), \]  
\[ u'(a) = \theta_1. \]  
\[ u'(b) = \psi'(b) + \theta_1 \rho'(b) + \theta_2 \vartheta'(b). \]  

Imposing the boundary conditions \( u'(a) = \beta \) and \( u'(b) = \lambda \) in (57) and (58) produces \( \theta_1 = \beta \) and \( \theta_2 = \frac{\gamma - \psi'(b) - \beta \rho'(b)}{\vartheta'(b)} \). Therefore, if \( \vartheta'(b) \neq 0 \), the unique solution of the two-point BVP (1) with boundary condition type II is given by:

\[ u(t) = \psi(t) + \beta \rho(t) + \frac{\gamma - \psi'(b) - \beta \rho'(b)}{\vartheta'(b)} \vartheta(t). \]  

**Algorithm 1: RKT Method via Linear Shooting Technique:**

To approximate the solution of BVP (1) with boundary condition type I:

**INPUT:** \( \alpha, \beta, \gamma \) boundary conditions; \( a, b \) endpoints; \( N \) number of subintervals.

**OUTPUT:** approximations \( \varphi_{1,i} \) to \( u(t_i) \); \( \varphi_{2,i} \) to \( u'(t_i) \); \( \varphi_{3,i} \) to \( u''(t_i) \) for each \( i = 0, 1, ..., N \).

**Step 1:** Set \( h = (b - a)/N \);

\[ \psi_{1,0} = \alpha; \]  
\[ \psi_{2,0} = 0; \]  
\[ \psi_{3,0} = 0; \]  
\[ \rho_{1,0} = 0; \]
\[ \rho_{2,0} = 1; \]
\[ \rho_{3,0} = 0; \]
\[ \vartheta_{1,0} = 0; \]
\[ \vartheta_{2,0} = 0; \]
\[ \vartheta_{3,0} = 1. \]

Step 2: For \( i = 0, \ldots, N - 1 \) do Step 3 and Step 4.
(RKD method is used in Step 3 and Step 4.)
Step 3: Set \( t = a + ih \).
Step 4: Set

\[
\kappa_1 = f_1(t, \psi_{1,i});
\]
\[
\kappa_i = f_1(t + c_i h, \psi_{1,i} + c_i h \psi_{2,i} + \frac{h^2}{2} c_i^2 \psi_{3,i} + h^3 \sum_{j=1}^{i-1} a_{ij} \kappa_j);
\]
\[
\psi_{1,i+1} = \psi_{1,i} + h \psi_{2,i} + \frac{h^2}{2} \psi_{3,i} + h^3 \sum_{i=1}^{\mu} \hat{b}_i \kappa_i;
\]
\[
\psi_{2,i+1} = \psi_{2,i} + h \psi_{3,i} + h^2 \sum_{i=1}^{\mu} \hat{b}_i \kappa_i;
\]
\[
\psi_{3,i+1} = \psi_{3,i} + h \sum_{i=1}^{\mu} \hat{b}_i \kappa_i;
\]
\[
\bar{\kappa}_1 = f_2(t, \rho_{1,i});
\]
\[
\bar{\kappa}_i = f_2(t + c_i h, \rho_{1,i} + c_i h \rho_{2,i} + \frac{h^2}{2} c_i^2 \rho_{3,i} + h^3 \sum_{j=1}^{i-1} a_{ij} \bar{\kappa}_j);
\]
\[
\rho_{1,i+1} = \rho_{1,i} + h \rho_{2,i} + \frac{h^2}{2} \rho_{3,i} + h^3 \sum_{i=1}^{\mu} b_i \bar{\kappa}_i;
\]
\[
\rho_{2,i+1} = \rho_{2,i} + h \rho_{3,i} + h^2 \sum_{i=1}^{\mu} \hat{b}_i \bar{\kappa}_i;
\]
\[
\rho_{3,i+1} = \rho_{3,i} + h \sum_{i=1}^{\mu} \hat{b}_i \bar{\kappa}_i;
\]
\[
\bar{\kappa}_1 = f_3(t, \vartheta_{1,i});
\]
\[
\kappa_i = f_3(t + c_i h, \vartheta_{1,i} + c_i h \vartheta_{2,i} + \frac{h^2}{2} c_i^2 \vartheta_{3,i} + h^3 \sum_{j=1}^{i-1} a_{ij} \kappa_j);
\]

\[
\vartheta_{1,i+1} = \vartheta_{1,i} + h \vartheta_{2,i} + \frac{h^2}{2} \vartheta_{3,i} + h^3 \sum_{i=1}^{\mu} b_i \kappa_i;
\]

\[
\vartheta_{2,i+1} = \vartheta_{2,i} + h \vartheta_{3,i} + h^2 \sum_{i=1}^{\mu} \hat{b}_i \kappa_i;
\]

\[
\vartheta_{3,i+1} = \vartheta_{3,i} + h \sum_{i=1}^{\hat{\mu}} \hat{\kappa}_i;
\]

Step 5: (For boundary condition type I)

set \( \varphi_{1,0} = \alpha \);

\( \varphi_{2,0} = \beta \);

\( \varphi_{3,0} = \frac{(\gamma - \psi_{1,N} - \varphi_{2,0}\rho_{1,N})}{\vartheta_{1,N}} \);

OUTPUT \((a, \varphi_{1,0}, \varphi_{2,0}, \varphi_{3,0})\).

Step 5: (For boundary condition type II)

set \( \varphi_{1,0} = \alpha \);

\( \varphi_{2,0} = \beta \);

\( \varphi_{3,0} = \frac{(\gamma - \psi_{2,N} - \varphi_{2,0}\rho_{2,N})}{\vartheta_{2,N}} \);

OUTPUT \((a, \varphi_{1,0}, \varphi_{2,0}, \varphi_{3,0})\).

Step 6: For \( i = 1, \ldots, N \) set

\[
\Omega_1 = \psi_{1,i} + \varphi_{2,0} \rho_{1,i} + \varphi_{3,0} \vartheta_{1,i};
\]

\[
\Omega_2 = \psi_{2,i} + \varphi_{2,0} \rho_{2,i} + \varphi_{3,0} \vartheta_{2,i};
\]

\[
\Omega_3 = \psi_{3,i} + \varphi_{2,0} \rho_{3,i} + \varphi_{3,0} \vartheta_{3,i};
\]

\( t = a + ih \);

OUTPUT \((t, \Omega_1, \Omega_2, \Omega_3)\).

Step 7: Complete.
4. Numerical Results

In this section, we selected four problems to test the performance of the RKT3s4 method in terms of accuracy and effectiveness. The first two problems are BVP problems, while the second problems are a class of BVP called self-adjoint singularly perturbed boundary value problems (SPBVPs).

For the numerical comparisons, we chose the following fourth-order RK types methods of comparisons for the first and second problems, whereas for SPBVPs we wanted to examine whether the new method could solve this type of problem or not, for this we compared with Quartic B-spline methods. Taking into consideration that the Quartic B-spline and Runge-Kutta type methods have different behavior.

All calculations were performed using the code written by us in C program.

- **RKT3s4**: The three-stage fourth-order explicit RKT method derived in this paper;
- **RKD4M**: The three-stage fourth-order RKT method of [23];
- **RK4**: The four-stage fourth-order RK method given in [22];
- **RK4Z**: The five-stage fourth-order RK method of [24];
- **F.N**: The number of function evaluations;
- **MAXE**: Max (|y(tn) - yn|) which is the maximum between absolute errors of the exact solutions and the computed solutions;

**Problem 1.** (See Arshad et al. [25]) Consider the inhomogeneous linear two-point BVP

\[ u''' = tu + (t^3 - 2t^2 - 5t - 3)e^t, \quad 0 \leq t \leq 1, \]
\[ u(0) = 0, \quad u'(0) = 1, \quad u(1) = 0. \]

The analytic solution is given by

\[ u(t) = t(1 - t)e^t. \]

**Problem 2.** (See Abd El-Salam et al. [26]) Consider the inhomogeneous linear two-point BVP

\[ u''' + u = (t - 4) \sin t + (1 - t) \cos t, \quad 0 \leq t \leq 1, \]
\[ u(0) = 0, \quad u'(0) = -1, \quad u'(1) = \sin(1). \]
The analytic solution is given by
\[ u(t) = (t - 1) \sin t. \]

**Problem 3.** (See Saini et al. [27]) Consider the inhomogeneous two-point SPBVP

\[
- \epsilon u''' + u = 6 \epsilon (1 - t)^5 t^3 - 6 \epsilon^2 \left(6 (1 - t)^5 - 90 (1 - t)^4 t + 180 (1 - t)^3 t^2 - 60 (1 - t)^2 t^3\right),
\]

\[ u(0) = 0, \quad u'(0) = 0, \quad u(1) = 0. \]

The analytic solution is given by
\[ u(t) = 6 t^3 \epsilon (1 - t)^5. \]

**Problem 4.** (See Saini et al. [27]) Consider the inhomogeneous two-point SPBVP

\[
- \epsilon u''' + u = 81 \epsilon^2 \cos 3t + 3 \epsilon \sin 3t,
\]

\[ u(0) = 0, \quad u'(0) = 9 \epsilon, \quad u(1) = 3 \epsilon \sin(3). \]

The analytic solution is given by
\[ u(t) = 3 \epsilon \sin 3t. \]

It is easy to note through Figures 1, 2 that RKT3s4 method performs well when integrating third-order BVP compared to RKD4M, RK4, and RK4Z methods and we can observe from Tables 3, 4 that the numerical results for the RKT3s4 agree to one decimal place when compared with the fourth-order RK methods (RKD4M, RK4, and RK4Z). Add to that, the figures confirm that RKT3s4 method requires fewer function evaluations than RK4 and RK4Z methods. That is because when we solve problems (1), (2) using RK4 and RK4Z methods, we need to reduce them to a system of first-order BVPs which is three times the dimension. Therefore, using a direct method means skipping the step which involves solving the system of linear differential equations which can
save a large amount of work, in terms of the number of function evaluations. As for solving third order SPBVP with respect to different values of $\epsilon$, we observed that the results obtained in Tables 5, 8 showed the efficiency of the new method. Figures 3, 4 are also visualized comparing the given tables. Thus, the RKT3s4 method is very efficient and accurate to evaluate according to the given problems and boundary conditions.

### Table 3: Maximum errors and number of function evaluations of Problem 1.

<table>
<thead>
<tr>
<th>$N$</th>
<th>Methods</th>
<th>F.N</th>
<th>MAXE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>RKT3s4</td>
<td>12</td>
<td>4.290020890 (-6)</td>
</tr>
<tr>
<td></td>
<td>RKD4M</td>
<td>12</td>
<td>2.572186843 (-5)</td>
</tr>
<tr>
<td></td>
<td>RK4</td>
<td>48</td>
<td>4.653033201 (-5)</td>
</tr>
<tr>
<td></td>
<td>RK4Z</td>
<td>60</td>
<td>4.653033201 (-5)</td>
</tr>
<tr>
<td>8</td>
<td>RKT3s4</td>
<td>24</td>
<td>2.314759256 (-7)</td>
</tr>
<tr>
<td></td>
<td>RKD4M</td>
<td>24</td>
<td>1.595541387 (-6)</td>
</tr>
<tr>
<td></td>
<td>RK4</td>
<td>96</td>
<td>3.384321949 (-6)</td>
</tr>
<tr>
<td></td>
<td>RK4Z</td>
<td>120</td>
<td>3.384321949 (-6)</td>
</tr>
<tr>
<td>16</td>
<td>RKT3s4</td>
<td>48</td>
<td>8.975345855 (-9)</td>
</tr>
<tr>
<td></td>
<td>RKD4M</td>
<td>48</td>
<td>9.853887617 (-8)</td>
</tr>
<tr>
<td></td>
<td>RK4</td>
<td>192</td>
<td>2.202264262 (-7)</td>
</tr>
<tr>
<td></td>
<td>RK4Z</td>
<td>240</td>
<td>2.202264262 (-7)</td>
</tr>
<tr>
<td>32</td>
<td>RKT3s4</td>
<td>96</td>
<td>4.998855219 (-10)</td>
</tr>
<tr>
<td></td>
<td>RKD4M</td>
<td>96</td>
<td>5.856636787 (-9)</td>
</tr>
<tr>
<td></td>
<td>RK4</td>
<td>384</td>
<td>1.406407724 (-8)</td>
</tr>
<tr>
<td></td>
<td>RK4Z</td>
<td>480</td>
<td>1.406407724 (-8)</td>
</tr>
</tbody>
</table>

5. **Conclusion**

The two-point boundary value problems of third-order ordinary differential equations with boundary conditions type I and type II, can be solved using the direct method of Runge-Kutta via shooting technique. The experiments have shown that the new method worked well, as expected, which is to say that using a direct method to solve the higher-order ordinary differential equations is not only most efficient in terms of the absolute maximum global error but is
Table 4: Maximum errors and number of function evaluations of Problem 2.

<table>
<thead>
<tr>
<th>$N$</th>
<th>Methods</th>
<th>F.N</th>
<th>MAXE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>RKT3s4</td>
<td>12</td>
<td>1.967463006 (-6)</td>
</tr>
<tr>
<td>8</td>
<td>RKD4M</td>
<td>12</td>
<td>1.799932279 (-5)</td>
</tr>
<tr>
<td></td>
<td>RK4</td>
<td>48</td>
<td>1.159839228 (-4)</td>
</tr>
<tr>
<td></td>
<td>RK4Z</td>
<td>60</td>
<td>1.159839228 (-4)</td>
</tr>
<tr>
<td>16</td>
<td>RKT3s4</td>
<td>24</td>
<td>1.869702460 (-7)</td>
</tr>
<tr>
<td></td>
<td>RKD4M</td>
<td>24</td>
<td>1.549408651 (-6)</td>
</tr>
<tr>
<td></td>
<td>RK4</td>
<td>96</td>
<td>7.166950172 (-6)</td>
</tr>
<tr>
<td></td>
<td>RK4Z</td>
<td>120</td>
<td>7.166950172 (-6)</td>
</tr>
<tr>
<td>32</td>
<td>RKT3s4</td>
<td>48</td>
<td>5.482110596 (-8)</td>
</tr>
<tr>
<td></td>
<td>RKD4M</td>
<td>48</td>
<td>1.543556326 (-7)</td>
</tr>
<tr>
<td></td>
<td>RK4</td>
<td>192</td>
<td>4.432972907 (-7)</td>
</tr>
<tr>
<td></td>
<td>RK4Z</td>
<td>240</td>
<td>4.432972907 (-7)</td>
</tr>
<tr>
<td>64</td>
<td>RKT3s4</td>
<td>96</td>
<td>8.907090702 (-9)</td>
</tr>
<tr>
<td></td>
<td>RKD4M</td>
<td>96</td>
<td>1.693722997 (-8)</td>
</tr>
<tr>
<td></td>
<td>RK4</td>
<td>384</td>
<td>2.752917824 (-8)</td>
</tr>
<tr>
<td></td>
<td>RK4Z</td>
<td>480</td>
<td>2.752917824 (-8)</td>
</tr>
</tbody>
</table>

Table 5: Maximum error of RKT3s4 of Problem 3.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\epsilon = 1/16$</th>
<th>$\epsilon = 1/32$</th>
<th>$\epsilon = 1/64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$6.5 \times 10^{-6}$</td>
<td>$2.8 \times 10^{-6}$</td>
<td>$1.0 \times 10^{-6}$</td>
</tr>
<tr>
<td>20</td>
<td>$4.2 \times 10^{-7}$</td>
<td>$1.7 \times 10^{-7}$</td>
<td>$6.6 \times 10^{-8}$</td>
</tr>
<tr>
<td>40</td>
<td>$2.6 \times 10^{-8}$</td>
<td>$1.1 \times 10^{-8}$</td>
<td>$4.1 \times 10^{-9}$</td>
</tr>
</tbody>
</table>

more efficient as it can save a large amount of work, in terms of the number of function evaluations.
<table>
<thead>
<tr>
<th>$N$</th>
<th>$\epsilon = 1/16$</th>
<th>$\epsilon = 1/32$</th>
<th>$\epsilon = 1/64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$4.7 \times 10^{-4}$</td>
<td>$1.9 \times 10^{-4}$</td>
<td>$8.0 \times 10^{-5}$</td>
</tr>
<tr>
<td>20</td>
<td>$1.1 \times 10^{-4}$</td>
<td>$4.7 \times 10^{-5}$</td>
<td>$1.9 \times 10^{-5}$</td>
</tr>
<tr>
<td>40</td>
<td>$2.6 \times 10^{-5}$</td>
<td>$1.2 \times 10^{-5}$</td>
<td>$4.8 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 6: Maximum error of Problem 3 in Saini [27].

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\epsilon = 1/16$</th>
<th>$\epsilon = 1/32$</th>
<th>$\epsilon = 1/64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$2.9 \times 10^{-3}$</td>
<td>$9.2 \times 10^{-4}$</td>
<td>$1.4 \times 10^{-4}$</td>
</tr>
<tr>
<td>20</td>
<td>$1.2 \times 10^{-4}$</td>
<td>$3.8 \times 10^{-5}$</td>
<td>$6.8 \times 10^{-6}$</td>
</tr>
<tr>
<td>40</td>
<td>$6.4 \times 10^{-6}$</td>
<td>$2.1 \times 10^{-6}$</td>
<td>$4.6 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Table 7: Maximum error of Problem 3 in Akram [28].

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\epsilon = 1/16$</th>
<th>$\epsilon = 1/32$</th>
<th>$\epsilon = 1/64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$1.9 \times 10^{-5}$</td>
<td>$3.7 \times 10^{-5}$</td>
<td>$7.0 \times 10^{-5}$</td>
</tr>
<tr>
<td>20</td>
<td>$2.6 \times 10^{-6}$</td>
<td>$5.0 \times 10^{-6}$</td>
<td>$9.9 \times 10^{-6}$</td>
</tr>
<tr>
<td>40</td>
<td>$3.4 \times 10^{-7}$</td>
<td>$6.6 \times 10^{-7}$</td>
<td>$1.3 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 8: Maximum error of RKT3s4 of Problem 4.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\epsilon = 1/16$</th>
<th>$\epsilon = 1/32$</th>
<th>$\epsilon = 1/64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$2.4 \times 10^{-4}$</td>
<td>$1.0 \times 10^{-4}$</td>
<td>$4.0 \times 10^{-5}$</td>
</tr>
<tr>
<td>20</td>
<td>$6.1 \times 10^{-5}$</td>
<td>$2.6 \times 10^{-5}$</td>
<td>$1.0 \times 10^{-6}$</td>
</tr>
<tr>
<td>40</td>
<td>$1.5 \times 10^{-5}$</td>
<td>$6.4 \times 10^{-6}$</td>
<td>$2.5 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 9: Maximum error of Problem 4 in Saini [27].

Acknowledgements

This study has been supported by the Fundamental Research Grant Scheme (Ref. No. FRGS/1/2018/STG06/UPM/02/2) awarded by the Malaysia Min-
<table>
<thead>
<tr>
<th>N</th>
<th>$\epsilon = 1/16$</th>
<th>$\epsilon = 1/32$</th>
<th>$\epsilon = 1/64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$1.3 \times 10^{-2}$</td>
<td>$3.2 \times 10^{-3}$</td>
<td>$3.4 \times 10^{-4}$</td>
</tr>
<tr>
<td>20</td>
<td>$1.1 \times 10^{-3}$</td>
<td>$2.7 \times 10^{-4}$</td>
<td>$2.2 \times 10^{-5}$</td>
</tr>
<tr>
<td>40</td>
<td>$7.8 \times 10^{-5}$</td>
<td>$1.8 \times 10^{-5}$</td>
<td>$1.1 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 10: Maximum error of Problem 4 in Akram [28].

Figure 1: The efficiency curve for RKT3s4, RKD4M, RK4, and RK4Z for Problem 1 with $N = 2^i, i = 2, 3, 4, 5$.

References


Figure 2: The efficiency curve for RKT3s4, RKD4M, RK4, and RK4Z for Problem 2 with $N = 2^i$, $i = 2, 3, 4, 5$.


Figure 3: Comparison figure between RKT3s4, Saini, and Akram methods.


[12] B. Ahmad, J.J. Nieto and N. Shahzad, Generalized quasilinearization


