Kragujevac Journal of Mathematics
Volume 51(9) (2027), Pages 1575–1598.

ENHANCED EFFICIENCY IN SOLVING SPECIAL SECOND-ORDER LINEAR BVPs USING A SEVENTH-ORDER TWO-STEP RUNGE-KUTTA-NYSTRÖM METHOD VIA LINEAR SHOOTING

ATHRAA ABDULSALAM[1,2], NORAZAK SENU[2,3], M. Z. SAGHIR[4], HAZIZAH MOHD IJAM[5], AND MAYS BASIM[6]

Abstract. This study presents a seventh-order multi-step Runge-Kutta-Nyström (MSRKN) method and investigates its use as an initial value problem (IVP) integrator within the classical linear shooting framework for special second-order linear two-point boundary value problems (BVPs) under Dirichlet, Neumann, and Robin boundary conditions. The proposed method consists of six stages and requires four function evaluations per step due to the FSAL (first stage as last) property. In the shooting procedure, the BVP is reduced to two associated IVPs, which are solved using the proposed method, and their numerical solutions are combined to reconstruct the BVP solution. A linear stability analysis of the MSRKN method is provided, and numerical results illustrate its effectiveness compared to methods with similar characteristics in the literature.

1. Introduction

Boundary value problems arise in a wide range of scientific applications and are used to describe numerous chemical, physical, and biological phenomena. Depending on the problem, different types of boundary conditions may be imposed, including Dirichlet, Neumann, and Robin boundary conditions, related studies can be found in [1–5]. In the present study, we focus on special second-order linear two-point BVPs subject to these three types of boundary conditions. The theoretical and numerical

Key words and phrases. Multi-step Runge-Kutta-Nyström methods, Explicit methods, Boundary value problems, Shooting method, Second-order differential equations.

2020 *Mathematics Subject Classification.* Primary: 34B05. Secondary: 65L06, 34A12.

DOI

Received: February 08, 2026.

Accepted: October 06, 2026.

study of BVPs has been a prominent topic in the scientific literature. Under suitable conditions on the differential equation and boundary conditions, the existence and uniqueness of solutions can be established (see [6, 7]). However, analytical solutions are often difficult to obtain, and various numerical methods have therefore been developed for solving BVPs [3]. Efficiency and accuracy are important considerations in the development of such numerical methods, particularly when repeated function evaluations are required. Several numerical approaches have been developed for solving BVPs, including finite difference methods [8, 9], block methods [5, 10, 11], collocation methods [12, 13], finite element and Galerkin-type methods [14, 15], and spectral or pseudo-spectral methods [16]. Shooting methods are another widely used approach for BVPs [17], while multiple shooting methods have also been developed to improve the numerical treatment of problems for which single shooting may become sensitive or unstable [18]. In shooting methods, a BVP is transformed into one or more IVPs, which can then be solved using suitable IVP solvers, such as linear multistep or Runge-Kutta methods. For nonlinear BVPs, nonlinear shooting generally requires an iterative procedure and suitable initial guesses. In contrast, for linear BVPs, the linear shooting method uses the superposition principle to construct the BVP solution from solutions of associated IVPs. The present study is concerned with this linear shooting formulation. Although shooting methods can be computationally attractive because of their relatively small system size, numerical instability may arise when the associated IVPs contain growing modes, even when the original BVP is well-posed [19–21]. Finite difference approaches transform the continuous differential equation into a system of algebraic equations whose solution provides approximations at the nodal points. Although such methods are widely used for BVPs, the development and implementation of higher-order schemes may become more involved [10, 20]. Several studies have also considered methods that treat second-order BVPs directly, without first reducing them to equivalent systems of first-order equations [2, 5, 10]. Li and Wang [22] introduced the general formulation of MSRKN methods for special second-order IVPs and derived the corresponding order conditions based on Nyström-trees. They also constructed explicit two-step RKN methods of orders five and six with the FSAL property (for related developments on multi-step RKN methods for second-order IVPs, we refer the reader to [23–26]). Although these methods showed good efficiency, their construction was limited to orders five and six. This motivates the construction of a higher-order MSRKN method that can provide higher accuracy while preserving the computational advantages of the two-step RKN structure with the FSAL property. Moreover, the literature on Runge-Kutta-Nyström methods for solving boundary value problems is relatively limited. Therefore, in this study, we investigate the use of a higher-order MSRKN method as an IVP integrator within the linear shooting framework for special second-order linear two-point BVPs of the form

$$g''(x) = u(x)g(x) + r(x), \quad x \in [a, b], \tag{1.1}$$

subject to:

(a) Dirichlet boundary conditions

$$g(a) = \alpha, \quad g(b) = \beta, \tag{1.2}$$

(b) Neumann boundary conditions

$$g'(a) = \beta_1, \quad g'(b) = \beta_2, \tag{1.3}$$

(c) Robin boundary conditions

$$g'(a) + D_1 g(a) = \alpha, \quad g'(b) + D_2 g(b) = \beta, \tag{1.4}$$

where a, b, α, β, β_1, β_2, D_1 and D_2 are constants, while $u(x)$ and $r(x)$ are continuous real-valued functions on $[a, b]$. In this study, we construct a seventh-order explicit two-step RKN method. The method consists of six stages and requires four function evaluations per step due to the FSAL property. Although the proposed method can be applied to both linear and nonlinear special second-order IVPs of the form $g'' = f(x, g)$, its application to BVPs in the present study is restricted to linear problems because the linear shooting formulation relies on the principle of superposition. The following outlines the structure of this paper. Section 2 presents the general formulation and fundamental concepts of the MSRKN method, as well as the construction of the new method. Section 3 discusses the convergence, consistency, and stability of the proposed MSRKN method. Section 4 illustrates the shooting technique and details the coding process for implementing the MSRKN method within the linear shooting framework. In Section 5, we examine several test problems to verify the method's validity and efficacy. Finally, Section 6 summarizes the study's findings.

2. Fundamental Concepts

2.1. **The Formulation of MSRKN Method.** Li and Wang [22] defined multi-step RKN methods as follows.

Definition 2.1. A Runge-Kutta-Nyström method with $\jmath$-step and ς-stage, formulated for solving special second-order IVPs of the form $g'' = f(x, g)$, is defined as:

$$\begin{aligned}
G_i &= \sum_{\ell=1}^{\jmath} r_{i\ell} g_{n-\ell+1} + h \sum_{\ell=1}^{\jmath} p_{i\ell} g'_{n-\ell+1} + h^2 \sum_{j=1}^{\varsigma} a_{ij} f(x_n + c_j h, G_j), \\
g_{n+1} &= \sum_{\ell=1}^{\jmath} q_\ell g_{n-\ell+1} + h \sum_{\ell=1}^{\jmath} w_\ell g'_{n-\ell+1} + h^2 \sum_{i=1}^{\varsigma} b_i f(x_n + c_i h, G_i), \\
g'_{n+1} &= \sum_{\ell=1}^{\jmath} \nu_\ell g'_{n-\ell+1} + h \sum_{i=1}^{\varsigma} \bar{b}_i f(x_n + c_i h, G_i),
\end{aligned} \tag{2.1}$$

where $c_i, a_{ij}, r_{i\ell}, b_i, p_{i\ell}, q_\ell, \nu_\ell, w_\ell$, and $\bar{b}_i$ for $\ell = 1, \ldots, \jmath$, $i, j = 1, 2, \ldots, \varsigma$ are real numbers.

The Butcher tableau shown in Table 1 provides a compact representation of the general coefficients in formulation (2.1).

TABLE 1. General Butcher tableau for the $\jmath$-step ς-stage RKN method defined by (2.1).

$$\begin{array}{c|ccc|ccc|ccc} c_1 & r_{11} & \dots & r_{1\jmath} & p_{11} & \dots & p_{1\jmath} & a_{11} & \dots & a_{1\varsigma} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_\varsigma & r_{\varsigma 1} & \dots & r_{\varsigma\jmath} & p_{\varsigma 1} & \dots & p_{\varsigma\jmath} & a_{\varsigma 1} & \dots & a_{\varsigma\varsigma} \\ \hline & q_1 & \dots & q_\jmath & w_1 & \dots & w_\jmath & b_1 & \dots & b_\varsigma \\ & & & & \nu_1 & \dots & \nu_\jmath & \bar{b}_1 & \dots & \bar{b}_\varsigma \end{array}$$

It is common to rewrite the non-autonomous problem $g'' = f(x, g)$ in an equivalent autonomous form, as shown below:

$$\begin{pmatrix} x \\ g \end{pmatrix}'' = \begin{pmatrix} 0 \\ f(x, g) \end{pmatrix}. \tag{2.2}$$

Applying method (2.1) to either the original non-autonomous problem (1.1) or its autonomous counterpart (2.2) yields equivalent numerical solution under these conditions:

$$\begin{aligned} &\sum_{\ell=1}^{\jmath} r_{i\ell} = 1, \quad \sum_{\ell=1}^{\jmath} q_\ell = 1, \quad \sum_{\ell=1}^{\jmath} \nu_\ell = 1, \\ &\sum_{\ell=1}^{\jmath} r_{i\ell}(1-\ell) + \sum_{\ell=1}^{\jmath} p_{i\ell} = c_i, \quad \sum_{\ell=1}^{\jmath} q_\ell(1-\ell) + \sum_{\ell=1}^{\jmath} w_\ell = 1, \end{aligned} \tag{2.3}$$

for more details refer to [22]. For the remainder of this article, under the conditions (2.3), our attention is restricted to the autonomous case, written in the form below:

$$g'' = f(g), \quad x \in [a, b]. \tag{2.4}$$

Abdulsalam et al. [27] used the Nyström series and Nyström tree theories to extend the order conditions of the MSRKN method up to order eight.

Theorem 2.1. *For exact starting values* (*the local assumptions*)

$$g_{n-\ell+1} = g(x_{n-\ell+1}), \quad g'_{n-\ell+1} = g'(x_{n-\ell+1}), \quad \ell = 1, \dots, \jmath,$$

the MSRKN method (2.1) *has order* p, $p \geq 2$, *if and only if*

$$1 = \sum_{\ell=1}^{\jmath} q_\ell (1-\ell)^{\rho(t)} + \sum_{\ell=1}^{\jmath} w_\ell (1-\ell)^{\rho(t)-1} \rho(t) + \sum_{i=1}^{\varsigma} b_i \Psi_i''(t), \quad p \geq \rho(t), \tag{2.5}$$

$$\rho(t) = \sum_{\ell=1}^{\jmath} \nu_\ell (1-\ell)^{\rho(t)-1} \rho(t) + \sum_{i=1}^{\varsigma} \bar{b}_i \Psi_i''(t), \quad p+1 \geq \rho(t), \tag{2.6}$$

where t *is in Nyström trees* (*see* [22]).

Using the following simplifying conditions provided in [22, Equation (13)]

(2.7)
$$\sum_{j=1}^{\varsigma} a_{ij}c_j^{\alpha} = \frac{1}{(\alpha+1)(\alpha+2)}\left(c_i^{\alpha+2} - \sum_{\ell=1}^{\jmath} r_{i\ell}\,(1-\ell)^{\alpha+2} - \sum_{\ell=1}^{\jmath} p_{i\ell}\,(1-\ell)^{\alpha+1}\,(\alpha+2)\right),$$

for $0 \le \alpha \le q_1 - 2$, where q_1 is the stage order of G_i. Setting $\alpha = 0, \ldots, 5$ in (2.7) yields the six simplifying conditions for G_i which are used in constructing MSRKN methods.

With the six simplifying conditions from (2.7), Theorem 2.1 provides the eighth-order conditions of MSRKN methods as follows.

The order conditions for g_{n+1}:

$$\sum_{\ell=1}^{\jmath} q_\ell\,(1-\ell)^{\alpha+2} + (\alpha+2)\sum_{\ell=1}^{\jmath} w_\ell\,(1-\ell)^{\alpha+1} + (\alpha+2)(\alpha+1)\sum_{i=1}^{\varsigma} b_i c_i^{\alpha} = 1, \tag{2.8}$$

where $0 \le \alpha \le 6$. The order conditions for g'_{n+1}:

$$\sum_{\ell=1}^{\jmath} \nu_\ell\,(1-\ell)^{\alpha+1} + (\alpha+1)\sum_{i=1}^{\varsigma} \bar{b}_i c_i^{\alpha} = 1, \quad 0 \le \alpha \le 7. \tag{2.9}$$

Additionally, for the pth-order MSRKN method (2.1), the coefficients of the leading term associated with the local truncation error of g_{n+1} are given by

(2.10)
$$er_{p+1}(t_j) = \frac{\alpha(t_j)}{\rho(t_j)!}\left[1 - \sum_{\ell=1}^{\jmath} q_\ell(1-\ell)^{\rho(t_j)} - \sum_{\ell=1}^{\jmath} w_\ell(1-\ell)^{\rho(t_j)-1}\rho(t_j) - \sum_{i=1}^{\varsigma} b_i\Psi_i''(t_j)\right],$$

for $\rho(t_j) = p+1$, with the corresponding error constant

$$C_{p+1} = \|(er_{p+1}(t_1), \ldots, er_{p+1}(t_k))\|_2, \tag{2.11}$$

where k is the number of trees of order $p+1$

Similarly, the coefficients of the leading term corresponding to the local truncation error of g'_{n+1} in the pth-order MSRKN method (2.1) are given by

$$er'_{p+1}(t_j) = \frac{\alpha(t_j)}{\rho(t_j)!}\left[\rho(t_j) - \sum_{\ell=1}^{\jmath} \nu_\ell(1-\ell)^{\rho(t_j)-1}\rho(t_j) - \sum_{i=1}^{\varsigma} \bar{b}_i\Psi_i''(t_j)\right], \tag{2.12}$$

for $\rho(t_j) = p+2$, with error constant

$$C'_{p+1} = \|(er'_{p+1}(t_1), \ldots, er'_{p+1}(t_m))\|_2, \tag{2.13}$$

where m refers to the number of trees of order $p+2$.

2.2. **The Construction of MSRKN Method.** This subsection focuses on constructing an explicit MSRKN method, specifically, the construction of a seventh-order two-step RKN method in the case of $\varsigma = 6$ and $\jmath = 2$. Take a look at the explicit two-step RKN methods with the FSAL property, as outlined in the coefficients table Table 2.2.

-1	0	1	0	0	0					
0	1	0	0	0	0	0				
c_3	r_{31}	r_{32}	p_{31}	p_{32}	a_{31}	a_{32}	0			
c_4	r_{41}	r_{42}	p_{41}	p_{42}	a_{41}	a_{42}	a_{43}	0		
c_5	r_{51}	r_{52}	p_{51}	p_{52}	a_{51}	a_{52}	a_{53}	a_{54}	0	
1	q_1	q_2	w_1	w_2	b_1	b_2	b_3	b_4	b_5	0
	q_1	q_2	w_1	w_2	b_1	b_2	b_3	b_4	b_5	0
			ν_1	ν_2	$\bar{b}_1$	$\bar{b}_2$	$\bar{b}_3$	$\bar{b}_4$	$\bar{b}_5$	$\bar{b}_6$

TABLE 2.

To find the parameters of this method, (2.3) gives the following essential relation

$$\begin{aligned} &r_{31}+r_{32}=1, \quad r_{41}+r_{42}=1, \quad r_{51}+r_{52}=1, \quad q_1+q_2=1,\\ &\nu_1+\nu_2=1, \quad -r_{32}+p_{31}+p_{32}=c_3, \quad -r_{42}+p_{41}+p_{42}=c_4,\\ &-r_{52}+p_{51}+p_{52}=c_5, \quad -q_2+w_1+w_2=1. \end{aligned} \tag{2.14}$$

By setting $c_1=-1$ and $c_2=0$, and treating c_3, c_4, c_5, and q_2 as free coefficients, we solve the equations (2.14) along with the algebraic condition (2.8) and (2.9) by taking $\alpha=0,\ldots,5$ and $\alpha=0,\ldots,6$, respectively. Additionally, the first four simplifying conditions (2.7) at $\alpha=0,\ldots,3$ for the third internal stage (to give the values of a_{3j}), the first five simplifying conditions (2.7) at $\alpha=0,\ldots,4$ for the fourth internal stage (to give the values of a_{4j}), and the first six simplifying conditions (2.7) at $\alpha=0,\ldots,5$ for the fifth internal stage (to give the values of a_{5j}), by using Solve command in Maple. This yields a four-parametric family of methods with c_3, c_4, c_5, and q_2 as free parameters. Next, we aim to choose suitable values for these free coefficients. Some studies select the free parameters by minimizing the error constants of the method (see [28,29]). In this study, we investigated the simultaneous minimization of the error constants C_8 and C_8', defined by (2.11) and (2.13), respectively. For convenience, we define the combined error measure as

$$GE_8=\sqrt{C_8^2+(C_8')^2}. \tag{2.15}$$

The results showed that minimizing the error constants alone does not necessarily give a better method. Although very small values of C_8 and C_8' can be obtained, some of the resulting coefficients become considerably large and the stability interval is reduced. For example, an unconstrained simultaneous minimization reduced GE_8 from approximately 1.5193×10^{-4} for the adopted parameter set to 4.5524×10^{-6}. However, the maximum absolute value of the method coefficients increased from approximately 4.13 to 178.36, while the stability interval decreased from $(0,0.8200)$ to approximately $(0,0.1668)$.

We also considered a constrained simultaneous optimization to control the growth of the method coefficients. By restricting the maximum absolute value of the coefficients

to 5, the combined error measure was reduced to approximately

$$GE_8 = 4.9070 \times 10^{-5}.$$

These results show that the selection of the free parameters should not depend only on minimizing the error constants, but the magnitude of the method coefficients and the stability properties should also be taken into consideration. Therefore, we choose the parameter values

$$c_3 = -\frac{6}{10}, \qquad c_4 = \frac{1}{5}, \qquad c_5 = \frac{7}{10}, \qquad q_2 = -\frac{2}{100}.$$

For these selected values, the error constants obtained from (2.11) and (2.13) are

$$C_8 = 4.6887 \times 10^{-5}, \qquad C_8' = 1.4451 \times 10^{-4},$$

which give $GE_8 = 1.5193 \times 10^{-4}$. Substituting these values into the corresponding coefficient expressions results in a seventh-order explicit two-step RKN method defined by the following coefficients:

$$\begin{aligned}
&a_{31} = \frac{54}{3125}, \quad a_{32} = \frac{36}{3125}, \quad a_{41} = \frac{36}{3125}, \quad a_{42} = \frac{24}{625}, \quad a_{43} = \frac{6}{125},\\
&a_{51} = -\frac{737405753}{2361600000}, \quad a_{52} = -\frac{414789851}{295200000}, \quad a_{53} = -\frac{321865369}{188928000},\\
&a_{54} = \frac{129757243}{188928000}, \quad b_1 = \frac{13313}{367200}, \quad b_2 = \frac{2329}{18900}, \quad b_3 = \frac{4349}{22464}, \quad b_4 = \frac{2801}{8640},\\
&b_5 = \frac{9704}{69615}, \quad \bar{b}_1 = \frac{59}{25092}, \quad \bar{b}_2 = \frac{4}{2583}, \quad \bar{b}_3 = \frac{2875}{153504}, \quad \bar{b}_4 = \frac{2675}{5904},\\
&\bar{b}_5 = \frac{6400}{13923}, \quad \bar{b}_6 = \frac{1051}{11808}, \quad p_{31} = -\frac{336}{3125}, \quad p_{32} = -\frac{594}{3125}, \quad p_{41} = -\frac{432}{3125},\\
&p_{42} = -\frac{294}{3125}, \quad p_{51} = \frac{67715879}{16400000}, \quad p_{52} = -\frac{20740867}{8200000}, \quad w_1 = \frac{101}{150},\\
&w_2 = \frac{23}{75}, \quad r_{31} = \frac{992}{3125}, \quad r_{32} = -\frac{2133}{3125}, \quad r_{41} = \frac{3024}{3125}, \quad r_{42} = -\frac{101}{3125},\\
&r_{51} = \frac{329171}{3280000}, \quad r_{52} = -\frac{2950829}{3280000}, \quad q_1 = \frac{51}{50}, \quad \nu_1 = \frac{40}{41}, \quad \nu_2 = \frac{1}{41}.
\end{aligned} \tag{2.16}$$

The method defined by (2.16) is denoted by TSRKN7.

3. Analysis of the MSRKN Method

In numerical analysis, the focus is both on designing numerical methods and on analyzing their behavior. This analysis typically focuses on three key concepts: stability (whether errors are damped out), order (how accurately the method approximates the solution), and convergence (whether the method approaches the exact solution).

Convergence. A numerical method is considered convergent if its computed solution approaches the exact solution of the differential equation as the step size h tends to zero ($h \to 0$). Convergence is an essential property of any practical numerical scheme.

Consistency and Order. The relationship between consistency and convergence is fundamental. For first-order ordinary differential equations (ODEs), Lambert [30] established that a numerical method must accurately represent the differential system to be convergent. This accuracy is formally assessed through the concept of consistency, which is verified by replacing the numerical solution with the exact solution. A necessary condition for consistency is that the method's order p satisfies $p \geq 1$. This requirement also extends to numerical methods for higher-order ODEs, as confirmed by Fatunla [31].

The MSRKN method proposed in this work is consistent, as it achieves seventh order ($p = 7 \geq 1$). However, consistency alone does not guarantee convergence. For the MSRKN method considered as an IVP integrator, consistency together with zero-stability establishes convergence under the standard assumptions for multistep methods. The convergence of the complete linear shooting procedure, including the boundary reconstruction, is discussed in Section 4.

Stability. To study zero-stability, we apply the MSRKN method described by equation (2.1) to the equation $g'' = 0$, we obtain:

$$U_n = D\, U_{n-1},$$

where $U_n = \left(g_{n+1}, g_n, \ldots, g_{n-\ell+2}, h\, g'_{n+1}, h\, g'_n, \ldots, h\, g'_{n-\ell+2}\right)^T$ and

$$D = \begin{bmatrix} q^T & w^T \\ \tilde{I} & \mathbf{0} \\ \mathbf{0} & \nu^T \\ \mathbf{0} & \tilde{I} \end{bmatrix}_{2\jmath \times 2\jmath}, \quad \text{with } \tilde{I} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix}_{(\jmath-1)\times \jmath}. \tag{3.1}$$

According to [22], MSRKN method (2.1) is zero-stable iff all roots of the matrix D lie within or on the unit circle, and the multiplicity of the zeros on the unit circle does not exceed two.

To check whether a numerical method will give reasonable results at a given step size $h > 0$, we need a different concept of stability than zero stability. We apply the MSRKN method to the scalar test equation

$$g'' = -\omega^2 g, \quad \omega > 0. \tag{3.2}$$

Applying the MSRKN method (2.1) to test Problem (3.2), we obtain

$$\begin{aligned} G =& R\, g + h\, P\, g' - \Lambda^2\, A\, G, \\ g_{n+1} =& q^T\, g + h\, w^T\, g' - \Lambda^2\, b^T\, G, \\ h\, g'_{n+1} =& h\, \nu^T\, g' - \Lambda^2\, \bar{b}^T\, G, \end{aligned}$$

where $g = (g_n, \ldots, g_{n-\jmath+1})^T$, $g' = (g'_n, \ldots, g'_{n-\jmath+1})^T$, $\Lambda = h\,\omega$. The elimination of the vectors G leads to the recursion:

$$g_{n+1} = N_3\, g + h\, N_4\, g', \tag{3.3}$$

$$h\,g'_{n+1} = N_5\,g + h\,N_6\,g',$$

where

$$\begin{aligned}
N_2 =& I + \Lambda^2\,A,\\
N_3 =& q^T - \Lambda^2\,b^T\,N_2^{-1}\,R,\\
N_4 =& w^T - \Lambda^2\,b^T\,N_2^{-1}\,P,\\
N_5 =& -\Lambda^2\,\bar{b}^T\,N_2^{-1}\,R,\\
N_6 =& \nu^T - \Lambda^2\,\bar{b}^T\,N_2^{-1}\,P.
\end{aligned}$$

The recursion (3.3) may alternatively be written as

$$U_n = M(\Lambda)\,U_{n-1},$$

where $U_n = \Big(g_{n+1}, g_n, \ldots, g_{n-\ell+2}, h\,g'_{n+1}, h\,g'_n, \ldots, h\,g'_{n-\ell+2}\Big)^T$ and

$$M(\Lambda) = \begin{bmatrix} N_3 & N_4\\ \tilde{I} & \mathbf{0}\\ N_5 & N_6\\ \mathbf{0} & \tilde{I}\end{bmatrix}_{2\jmath\times 2\jmath}.$$

We call this matrix $M(\Lambda)$ the stability matrix.The behavior of the numerical solution is determined by the eigenvalues $\xi_i(\Lambda)$ of this matrix, where $i = 1, \ldots, 2\jmath$. The stability function of the MSRKN method is expressed as:

$$\Omega\Big(\xi, M(\Lambda)\Big) = \det\Big[\xi I - M(\Lambda)\Big]. \tag{3.4}$$

Its characteristic equation can be formulated as:

$$\Omega\Big(\xi, M(\Lambda)\Big) = P_0\,(\Lambda)\,\xi^4 + P_1\,(\Lambda)\,\xi^3 + P_2\,(\Lambda)\,\xi^2 + P_3\,(\Lambda)\,\xi + P_4\,(\Lambda).$$

Consequently, the TSRKN7 method has the following stability polynomial:

$$\begin{aligned}
\Omega\Big(\xi, M(\Lambda)\Big) =& \xi^4 + \bigg(-\frac{4091}{2050} + \frac{36422561927}{14391000000}\,\Lambda^2 - \frac{35192419947307}{1770093000000000}\,\Lambda^4\\
&+ \frac{2821766174038481}{663784875000000000}\,\Lambda^6 - \frac{1982654689939}{221261625000000000}\,\Lambda^8\bigg)\xi^3\\
&+ \bigg(\frac{2031}{2050} - \frac{32246974373}{10926500000}\,\Lambda^2 + \frac{409112052905437}{295015500000000}\,\Lambda^4\\
&- \frac{15070773070821947}{221261625000000000}\,\Lambda^6 + \frac{44477369815891}{136161000000000000}\,\Lambda^8\bigg)\xi^2\\
&+ \bigg(\frac{259228018421}{196677000000}\,\Lambda^2 + \frac{120077873211559}{590031000000000}\,\Lambda^4\\
&- \frac{4907923138532837}{221261625000000000}\,\Lambda^6 + \frac{11}{2050} + \frac{662798391779879}{885046500000000000}\,\Lambda^8\bigg)\xi
\end{aligned}$$

$$- \frac{138565399357}{5043000000000000}\Lambda^8 - \frac{87376020592813}{82973109375000000}\Lambda^6 - \frac{1870721530187}{110630812500000}\Lambda^4 + \frac{3916317617}{36876937500}\Lambda^2 - \frac{1}{2050}. \tag{3.5}$$

Definition 3.1. The interval $R_s = \{\Lambda > 0 : |\xi_i(\Lambda)| < 1, i = 1, \ldots, 2\jmath\}$ is called the MSRKN method's stability interval (see [22]).

According to Definition 3.1, the stability interval consists of the values of $\Lambda > 0$ for which all characteristic roots satisfy $|\xi_i(\Lambda)| < 1$. As an illustrative case, we first consider $\Lambda = 1$. Solving the stability polynomial (3.5) with respect to ξ gives the characteristic roots

$$\xi_1 = -1.54345185, \quad \xi_{2,3} = 0.54030496 \pm 0.84146565\,\mathrm{i}, \quad \xi_4 = -0.05683510.$$

Their corresponding moduli are

$$|\xi_1| = 1.54345185, \quad |\xi_{2,3}| \approx 0.99999694, \quad |\xi_4| = 0.05683510.$$

Since $|\xi_1| > 1$, the value $\Lambda = 1$ does not belong to the stability interval.

To determine the stability boundary, we consider the unit-circle crossing $\xi = -1$, which is the relevant crossing for the proposed TSRKN7 method. Substituting $\xi = -1$ into the stability polynomial gives

$$\Omega(-1, M(\Lambda)) = 0.$$

Solving this equation numerically gives the positive roots

$$\Lambda = 0.8200079525, \quad \Lambda = 3.117437302, \quad \Lambda = 3.179090575.$$

Thus, the first positive crossing associated with $\xi = -1$ occurs at

$$\Lambda_{\max} = 0.8200079525.$$

At $\Lambda = \Lambda_{\max}$, the characteristic roots are

$$\xi_{1,2} = 0.6822157828 \pm 0.7311500974\,\mathrm{i}, \quad \xi_3 = -0.06295102973, \quad \xi_4 = -1.0000000000,$$

with corresponding moduli

$$|\xi_{1,2}| = 0.9999994196, \quad |\xi_3| = 0.06295102973, \quad |\xi_4| = 1.0000000000.$$

Hence, $\Lambda = 0.8200079525$ represents the stability boundary.

Furthermore, at $\Lambda = 0.83$, one of the characteristic roots becomes

$$\xi = -1.027045596,$$

with $|\xi| = 1.027045596 > 1$, confirming the transition from stability to instability.

Therefore, the stability interval of the proposed TSRKN7 method is

$$R_s = (0,\, 0.8200079525) \approx (0,\, 0.82). \tag{3.6}$$

The stability analysis presented above is related to TSRKN7 when applied as an IVP integrator. In the linear shooting approach, the BVP is transformed into two

TABLE 3. Representative characteristic-root moduli near the stability boundary of the proposed TSRKN7 method.

Λ	$\max_i \lvert\xi_i(\Lambda)\rvert$	Status
0.82	0.9999994198	Stable
0.8200079525	1.0000000000	Stability boundary
0.83	1.0270455960	Unstable
1.00	1.5434518500	Unstable

associated IVPs, which are solved numerically using TSRKN7. Therefore, the stability of the method is important for controlling the numerical behavior of the solutions of these IVPs. However, this property alone is not sufficient to ensure the stability of the complete shooting procedure. The reconstruction of the BVP solution may also influence the propagation of numerical errors, particularly when the shooting problem is poorly conditioned. This issue is considered in Section 4 together with the linear shooting formulation.

4. Reduction to Two IVPs: Linear Shooting Method

The solution of the linear BVP can be obtained by expressing it as the sum of a particular solution of the nonhomogeneous equation and a scalar multiple of a solution of the corresponding homogeneous equation. The construction for each type of boundary condition is presented below.

(i) The case of Dirichlet boundary conditions.

Suppose $s(x)$ is the unique solution of the IVP

$$s'' = f_1(x,s), \quad f_1(x,s) = u(x)s + r(x), \quad s(a) = \alpha, \quad s'(a) = 0. \tag{4.1}$$

Furthermore, suppose $v(x)$ is the unique solution of the corresponding homogeneous IVP

$$v'' = f_2(x,v), \quad f_2(x,v) = u(x)v, \quad v(a) = 0, \quad v'(a) = 1. \tag{4.2}$$

Since $s(x)$ is a particular solution of the nonhomogeneous equation and $v(x)$ is a solution of the corresponding homogeneous equation, the solution can be written as

$$g(x) = s(x) + C_2 v(x). \tag{4.3}$$

The initial conditions in (4.1)–(4.2) ensure that the first Dirichlet boundary condition $g(a) = \alpha$ is satisfied for any value of C_2.

Applying the second Dirichlet boundary condition $g(b) = \beta$ gives

$$C_2 = \frac{\beta - s(b)}{v(b)}, \quad v(b) \neq 0.$$

Therefore,

$$g(x) = s(x) + \frac{\beta - s(b)}{v(b)} v(x). \tag{4.4}$$

(ii) The case of Neumann boundary conditions.

Suppose $s(x)$ is the unique solution of the IVP

$$s'' = f_1(x,s), \quad f_1(x,s) = u(x)s + r(x), \quad s(a) = 0, \quad s'(a) = \beta_1. \tag{4.5}$$

Furthermore, suppose $v(x)$ is the unique solution of the corresponding homogeneous IVP

$$v'' = f_2(x,v), \quad f_2(x,v) = u(x)v, \quad v(a) = 1, \quad v'(a) = 0. \tag{4.6}$$

The solution is expressed as

$$g(x) = s(x) + C_2 v(x). \tag{4.7}$$

The initial conditions in (4.5)–(4.6) ensure that the first Neumann boundary condition $g'(a) = \beta_1$ is satisfied for any value of C_2.

Applying the second Neumann boundary condition $g'(b) = \beta_2$ gives

$$C_2 = \frac{\beta_2 - s'(b)}{v'(b)}, \quad v'(b) \neq 0.$$

Therefore,

$$g(x) = s(x) + \frac{\beta_2 - s'(b)}{v'(b)} v(x). \tag{4.8}$$

(iii) The case of Robin boundary conditions.

Let $s(x)$ be the solution of the IVP

$$s'' = f_1(x,s), \quad f_1(x,s) = u(x)s + r(x), \quad s(a) = 0, \quad s'(a) = \alpha. \tag{4.9}$$

Let $v(x)$ be the solution of the corresponding homogeneous IVP

$$v'' = f_2(x,v), \quad f_2(x,v) = u(x)v, \quad v(a) = 1, \quad v'(a) = -D_1. \tag{4.10}$$

The initial conditions in (4.9)–(4.10) are chosen such that

$$s'(a) + D_1 s(a) = \alpha, \quad v'(a) + D_1 v(a) = 0.$$

Therefore,

$$g(x) = s(x) + C_2 v(x) \tag{4.11}$$

satisfies the first Robin boundary condition for any value of C_2.

Applying the second Robin boundary condition $g'(b) + D_2 g(b) = \beta$ gives $s'(b) + C_2 v'(b) + D_2 [s(b) + C_2 v(b)] = \beta$. Hence,

$$C_2 = \frac{\beta - s'(b) - D_2 s(b)}{v'(b) + D_2 v(b)},$$

provided that $v'(b) + D_2 v(b) \neq 0$. Consequently,

$$g(x) = s(x) + \frac{\beta - s'(b) - D_2 s(b)}{v'(b) + D_2 v(b)} v(x). \tag{4.12}$$

The stability of the shooting procedure also depends on the conditioning of the boundary reconstruction. Based on the formulas given above, the quantities in the

denominators of the shooting coefficients for the Dirichlet, Neumann, and Robin boundary conditions are, respectively,

$$v(b), \quad v'(b), \quad v'(b) + D_2 v(b).$$

If any of these quantities is close to zero, numerical errors in the auxiliary IVP solutions may be amplified during the reconstruction of the BVP solution. Therefore, the stability of TSRKN7 as an IVP integrator is important in the shooting procedure. However, it alone does not ensure the stability of the complete BVP solution procedure.

The convergence analysis in Section 3 concerns TSRKN7 as an IVP integrator. We now consider how the numerical errors in the auxiliary IVP solutions affect the reconstructed BVP solution. Let s_h and v_h denote the numerical approximations to s and v, respectively, obtained using TSRKN7. Under the usual smoothness and stability assumptions, the seventh-order TSRKN7 method gives

$$s_h(x) - s(x) = O(h^7), \qquad v_h(x) - v(x) = O(h^7).$$

Similarly, for the derivative approximations required in the Neumann and Robin cases,

$$s_h'(x) - s'(x) = O(h^7), \qquad v_h'(x) - v'(x) = O(h^7).$$

Consider first the Dirichlet case. The exact and numerical shooting coefficients are

$$C_2 = \frac{\beta - s(b)}{v(b)}, \qquad C_{2,h} = \frac{\beta - s_h(b)}{v_h(b)}.$$

Provided that $|v(b)| \geq \delta > 0$, the denominator remains bounded away from zero for sufficiently small h. Since $s_h(b) - s(b) = O(h^7)$ and $v_h(b) - v(b) = O(h^7)$, it follows that $C_{2,h} - C_2 = O(h^7)$. Therefore, using $g_h(x) = s_h(x) + C_{2,h} v_h(x)$, we obtain

$$g_h(x) - g(x) = s_h(x) - s(x) + (C_{2,h} - C_2) v_h(x) + C_2\Big(v_h(x) - v(x)\Big) = O(h^7).$$

The same argument applies to the Neumann and Robin cases, provided that $|v'(b)| \geq \delta > 0$ and $|v'(b) + D_2 v(b)| \geq \delta > 0$, respectively. Hence, provided that the corresponding shooting denominator is bounded away from zero, the boundary reconstruction preserves the seventh-order convergence of the TSRKN7 approximations. Consequently, the reconstructed numerical BVP solution satisfies

$$g_h(x) - g(x) = O(h^7).$$

4.1. **Algorithm for the Shooting Technique Using MSRKN Method.** To approximate the solution of the BVP (1.1) with Dirichlet (1.2), Neumann (1.3), or Robin (1.4) boundary conditions we use the following algorithm.

INPUT: α, β, β_1, β_2 boundary conditions; D_1, D_2 constants; a, b endpoints; N number of subintervals.

OUTPUT: approximations $\varphi_{1,i}$ to $g(x_i)$; $\varphi_{2,i}$ to $g'(x_i)$ for each $i = 0, 1, ..., N$.

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

(If Dirichlet boundary condition), set

$$\begin{aligned} s_{1,0} &= \alpha; \\ s_{2,0} &= 0; \\ v_{1,0} &= 0; \\ v_{2,0} &= 1. \end{aligned}$$

(If Neumann boundary condition), set

$$\begin{aligned} s_{1,0} &= 0; \\ s_{2,0} &= \beta_1; \\ v_{1,0} &= 1; \\ v_{2,0} &= 0. \end{aligned}$$

(If Robin boundary condition), set

$$\begin{aligned} s_{1,0} &= 0; \\ s_{2,0} &= \alpha; \\ v_{1,0} &= 1; \\ v_{2,0} &= -D_1. \end{aligned}$$

Hint: Since the proposed method is a two-step method, additional procedures are required to obtain the starting values $s_{1,1}$, $s_{2,1}$, $v_{1,1}$ and $v_{2,1}$. In this study, the seventh-order RKN method proposed in [32] is used to provide the starting values for the seventh-order TSRKN7 method.

Step 2: For $i = 0, \ldots, N-1$, proceed with Steps 3 and 4.

Step 3: Set $x_i = a + ih$.

Step 4: Apply the MSRKN method (2.1) to both IVPs to obtain $s_{1,i}$, $s_{2,i}$, $v_{1,i}$ and $v_{2,i}$.

Step 5: Determine the shooting coefficient C_2 according to the type of boundary condition:

$$C_2 = \begin{cases} \dfrac{\beta - s_{1,N}}{v_{1,N}}, & \text{Dirichlet boundary condition,} \\ \dfrac{\beta_2 - s_{2,N}}{v_{2,N}}, & \text{Neumann boundary condition,} \\ \dfrac{\beta - s_{2,N} - D_2 s_{1,N}}{v_{2,N} + D_2 v_{1,N}}, & \text{Robin boundary condition.} \end{cases}$$

Step 6: For $i = 0, \ldots, N$, compute

$$\begin{aligned} g(x_i) &= s_{1,i} + C_2 v_{1,i}, \\ g'(x_i) &= s_{2,i} + C_2 v_{2,i}. \end{aligned}$$

Step 7: Output

$$(x_i, g(x_i), g'(x_i)), \quad i = 0, 1, \ldots, N.$$

Stop. The process is complete.

5. Numerical Experiments

Next, we will conduct experiments on commonly tested problems to demonstrate the efficiency of the suggested methods. We provide numerical outcomes resulting from the implementation of the proposed method in comparison with other existing methods in the scientific literature. All the following problems are tested using Code::Blocks and the methods considered are outlined below:

- **TSRKN7:** six-stage seventh-order two-step RKN method with the FSAL property developed in this study,
- **RKNM6:** six-stage sixth-order RKN method without the FSAL property given in [33],
- **RKF6:** eight-stage sixth-order RK method without the FSAL property given in [34],
- **RKS7:** eight-stage seventh-order RK method with the FSAL property given in [35],
- **MaxError:** Max $(|g(x_n)-g_n|)$ represents the maximum absolute error between the true and numerical solutions,
- **CPU:** execution time in seconds,
- **NFE:** number of function calls.

Problem 1. *Consider the inhomogeneous linear two-point BVP*

$$g'' = \frac{1+(1-x)\,g}{(1+x)^2}, \quad x \in [0,1],$$

where $g(0) = 1$, $g(1) = 0.5$.

Exact solution: $g(x) = \frac{1}{1+x}$. *Boundary condition: Dirichlet. Source:* [21].

Problem 2. *Consider the inhomogeneous linear two-point BVP*

$$g'' = -xg + (x^3 - x^2 - x + 3)\sin x + 4x\cos(x), \quad x \in [0,1],$$

where $g'(0) = -1$, $g'(1) = 2\sin 1$.

Exact solution: $g(x) = (x^2-1)\sin x$. *Boundary condition: Neumann. Source:* [1].

Problem 3. *Consider the inhomogeneous linear two-point BVP*

$$x^2 g'' = 2g - x, \quad x \in [2,3],$$

where $g'(2) - g(2) = \frac{8}{38}$, $g'(3) + g(3) = -\frac{7}{38}$.

Exact solution: $g(x) = \frac{1}{38}\,(19x - 5x^2 - \frac{36}{x})$. *Boundary condition: Robin. Source:* [36].

Problem 4. *Consider the linear second-order two-point BVP*

$$x^2 g'' = 2g - x, \quad x \in [2,3],$$

where $g(2) = \frac{10}{19}$, $g(3) = \dfrac{45}{38}$.

Exact solution: $g(x) = \frac{19\,x - 36\,x^{-1}}{38}$. *Boundary condition: Dirichlet. Source:* [21].

Problem 5. *Consider the homogeneous linear two-point BVP*

$$-g'' = (2 - 4\,x^2)\,g, \quad x \in [0, 1],$$

where $g'(0) = 0$, $g'(1) = -\frac{2}{e}$.

Exact solution: $g(x) = e^{-x^2}$. *Boundary condition: Neumann. Source:* [11].

Table 4. Computational results for Problem 1

N	Methods	NFE	MaxError	CPU
2	TSRKN7	8	8.287862 (-6)	0.017
	RKS7	34	1.202647 (-4)	0.020
	RKNM6	12	7.438293 (-5)	0.018
	RKF6	32	1.520758 (-3)	0.031
4	TSRKN7	16	1.444547 (-7)	0.020
	RKS7	70	1.343387 (-5)	0.032
	RKNM6	24	2.513069 (-5)	0.028
	RKF6	64	9.023445 (-4)	0.047
8	TSRKN7	32	2.262665 (-9)	0.028
	RKS7	142	1.436012 (-6)	0.041
	RKNM6	48	6.312182 (-6)	0.052
	RKF6	128	5.074732 (-4)	0.056
16	TSRKN7	64	4.775844 (-10)	0.039
	RKS7	286	1.625709 (-7)	0.070
	RKNM6	96	1.581706 (-6)	0.072
	RKF6	256	2.603846 (-4)	0.079

6. Discussion and Conclusions

In this section, we evaluate the efficacy of the suggested TSRKN7 method in solving special second-order two-point BVPs. We examine five problems with different types of boundary conditions. For comparison, we utilize one-step RK and RKN methods from previous studies. The results achieved are presented in Tables 4–8 and Figures 1-5, to illustrate the accuracy and efficiency of the suggested method.

Tables 4–8 demonstrate the computational efficiency of the proposed two-step RKN method compared to classical one-step RK and RKN methods. This efficiency is reflected in both lower computational time and fewer function evaluations. The improved efficiency arises because of the two-step structure of the method, in which information from previous steps is used in constructing the new approximation. Consequently, fewer function evaluations are required at each step. Furthermore, the proposed TSRKN7 method possesses the FSAL property, which further enhances its efficiency by allowing the last function evaluation of one step to be reused at the

TABLE 5. Computational results for Problem 2

N	Methods	NFE	MaxError	CPU
2	TSRKN7	8	1.053180 (-6)	0.014
	RKS7	34	2.909512 (-3)	0.020
	RKNM6	12	1.456514 (-2)	0.018
	RKF6	32	1.739850 (-2)	0.018
4	TSRKN7	16	1.502772 (-8)	0.022
	RKS7	70	4.390857 (-4)	0.029
	RKNM6	24	3.130815 (-3)	0.028
	RKF6	64	8.372208 (-3)	0.029
12	TSRKN7	48	3.354451 (-9)	0.031
	RKS7	214	1.769191 (-5)	0.042
	RKNM6	72	3.105721 (-4)	0.037
	RKF6	192	3.437062 (-3)	0.046
22	TSRKN7	88	5.188809 (-10)	0.043
	RKS7	394	2.937997 (-6)	0.062
	RKNM6	132	8.990008 (-5)	0.051
	RKF6	352	1.915893 (-3)	0.068

TABLE 6. Computational results for Problem 3

N	Methods	NFE	MaxError	CPU
2	TSRKN7	8	1.290333 (-7)	0.016
	RKS7	34	8.747820 (-5)	0.019
	RKNM6	12	4.105733 (-4)	0.018
	RKF6	32	5.730744 (-3)	0.018
22	TSRKN7	88	2.237867 (-8)	0.023
	RKS7	394	1.560365 (-7)	0.037
	RKNM6	132	8.115887 (-6)	0.038
	RKF6	352	6.413485 (-4)	0.036
60	TSRKN7	240	4.062007 (-9)	0.047
	RKS7	1078	8.038821 (-9)	0.061
	RKNM6	360	1.145705 (-6)	0.056
	RKF6	960	2.383831 (-4)	0.063
80	TSRKN7	320	8.509574 (-9)	0.062
	RKS7	1438	3.410749 (-9)	0.109
	RKNM6	480	6.489947 (-7)	0.142
	RKF6	1280	1.791408 (-4)	0.282

TABLE 7. Computational results for Problem 4

N	Methods	NFE	MaxError	CPU
2	TSRKN7	8	2.919686 (-8)	0.016
	RKS7	34	4.991318 (-6)	0.022
	RKNM6	12	1.720803 (-5)	0.021
	RKF6	32	4.526634 (-4)	0.027
12	TSRKN7	48	1.775841 (-9)	0.021
	RKS7	214	1.709298 (-8)	0.872
	RKNM6	72	4.649645 (-7)	0.031
	RKF6	192	7.825927 (-5)	0.038
22	TSRKN7	88	4.777106 (-9)	0.032
	RKS7	394	2.664129 (-9)	1.830
	RKNM6	132	1.370944 (-7)	0.052
	RKF6	352	4.288163 (-5)	0.059
32	TSRKN7	128	8.754593 (-10)	0.041
	RKS7	574	8.526307 (-10)	3.940
	RKNM6	192	6.456424 (-8)	0.064
	RKF6	512	2.948767 (-5)	0.109

TABLE 8. Computational results for Problem 5

N	Methods	NFE	MaxError	CPU
2	TSRKN7	8	1.263254 (-5)	0.016
	RKS7	34	5.215252 (-3)	0.018
	RKNM6	12	4.254378 (-3)	0.018
	RKF6	32	1.416041 (-2)	0.023
4	TSRKN7	16	1.665940 (-7)	0.026
	RKS7	70	3.842694 (-4)	0.038
	RKNM6	24	1.493308 (-3)	0.031
	RKF6	64	3.801202 (-3)	0.038
66	TSRKN7	264	6.758204 (-9)	0.058
	RKS7	1186	3.271049 (-8)	0.079
	RKNM6	396	6.320597 (-6)	0.078
	RKF6	1056	1.872582 (-4)	0.083
100	TSRKN7	400	7.273437 (-11)	0.097
	RKS7	1798	9.194134 (-9)	0.172
	RKNM6	600	2.758011 (-6)	0.172
	RKF6	1600	1.288023 (-4)	0.189

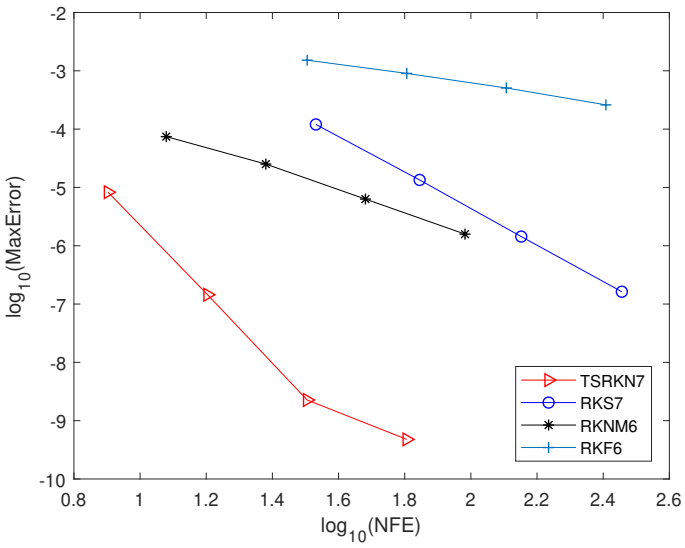


FIGURE 1. The efficiency curves for Problem 1

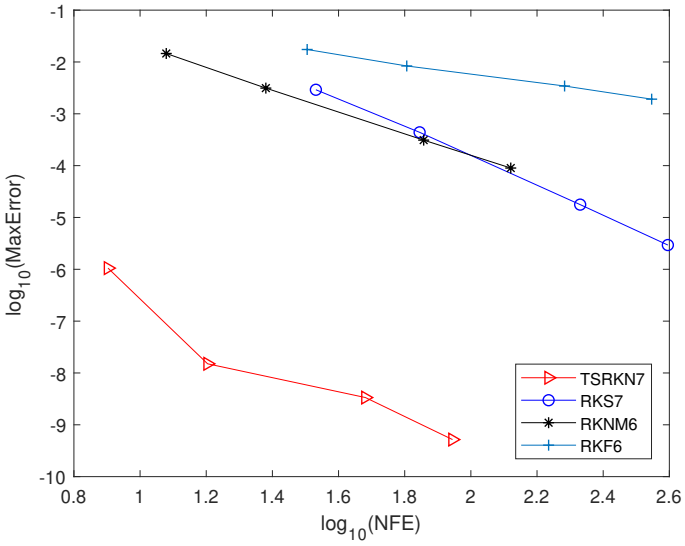


FIGURE 2. The efficiency curves for Problem 2

beginning of the subsequent step. As discussed in [22], a two-step RKN method with the FSAL property requires only $(\varsigma - \jmath)$ new function evaluations per step. Thus, the proposed TSRKN7 method requires only four new function evaluations per step. In contrast, employing classic RK methods (like RKS7 and RKF6) involves converting higher-order ordinary differential equations to systems of first-order equations, leading

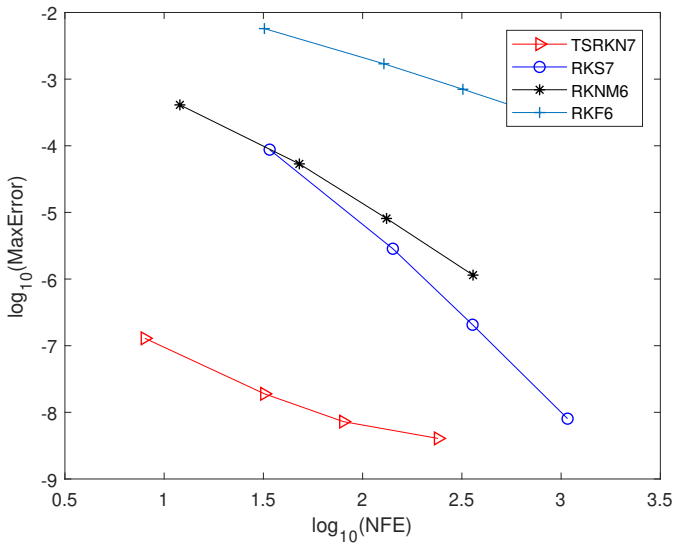


FIGURE 3. The efficiency curves for Problem 3

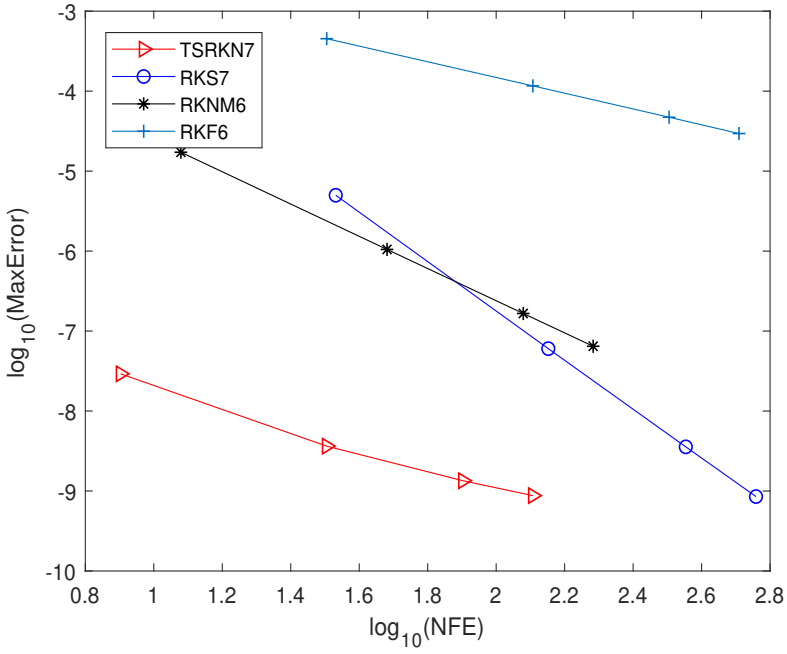


FIGURE 4. The efficiency curves for Problem 4

to a doubling of the problem's dimension and subsequently escalating the computational cost. In the case of the one-step RKN method, the RKNM6 method necessitates six function evaluations per step. For example, in Problem 1 with $N = 16$, TSRKN7

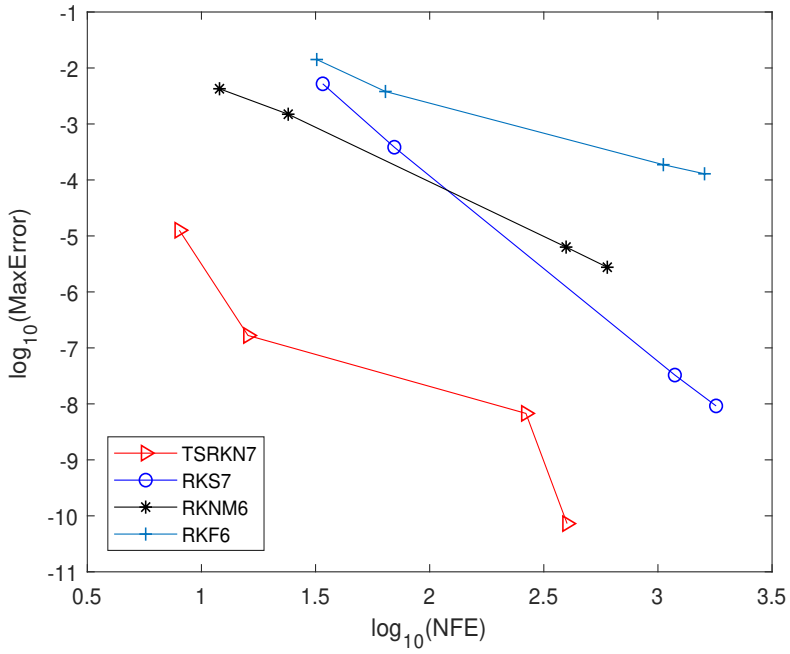


FIGURE 5. The efficiency curves for Problem 5

requires only 64 function evaluations, compared with 286, 96, and 256 evaluations for RKS7, RKNM6, and RKF6, respectively. Similarly, in Problem 2 with $N = 22$, TSRKN7 requires 88 function evaluations, whereas RKS7, RKNM6, and RKF6 require 394, 132, and 352 function evaluations, respectively. These results illustrate that the proposed method can achieve substantially fewer function evaluations. The reductions in NFE are also reflected in the computational times reported in the tables.

Regarding accuracy, the order of the methods should be taken into consideration. The proposed TSRKN7 method is seventh-order, whereas RKNM6 and RKF6 are sixth-order methods. Therefore, a direct comparison of their errors is not completely fair because TSRKN7 has a higher order. However, the comparison is still useful for evaluating the computational efficiency of the methods. The results show that TSRKN7 achieves smaller errors while requiring fewer function evaluations than the sixth-order methods.

For a more meaningful assessment of accuracy, the seventh-order RKS7 method provides the principal same-order comparison with the proposed TSRKN7 method. Both TSRKN7 and RKS7 are seventh-order methods, however, their numerical errors are not necessarily identical because their method formulations, coefficients, and associated error constants are different. The results in Tables 4, 5, and 8 show that TSRKN7 achieves smaller maximum errors than RKS7 for Problems 1, 2, and 5. For instance, in Problem 5 with $N = 100$, TSRKN7 produces a maximum error of 7.273437×10^{-11}, whereas RKS7 produces 9.194134×10^{-9}. In Problems 3 and 4, the

two seventh-order methods exhibit competitive accuracy. For example, in Problem 3 with $N = 60$, the maximum errors of TSRKN7 and RKS7 are 4.062007×10^{-9} and 8.038821×10^{-9}, respectively, while in Problem 4 with $N = 32$, the corresponding errors are 8.754593×10^{-10} and 8.526307×10^{-10}, respectively. Thus, the proposed TSRKN7 method does not uniformly outperform RKS7 in accuracy across all problems, but it provides competitive or superior accuracy across the test problems considered.

Figures 1–5 support the results presented in the tables. In particular, they show the maximum error versus the number of function evaluations for all the methods considered in solving the test problems. The figures show that TSRKN7 generally requires fewer function evaluations to achieve a given level of accuracy, demonstrating its favorable computational efficiency. The proposed multi-step RKN method therefore provides an effective alternative for the direct solution of special second-order linear two-point BVPs.

Acknowledgements. This research was funded by a grant from Universiti Putra Malaysia through Geran Putra Inisiatif (Project code: GPI/2023/9765400).

References

[1] A. Abdulsalam, N. Senu and Z. A. Majid, *Two-step RKN direct method for special second-order initial and boundary value problems*, IAENG Int. J. Appl. Math. **51**(3) (2021), 1–9.

[2] H. Ramos, M. A. Rufai and B. Carpentieri, *A pair of optimized Nyström methods with symmetric hybrid points for the numerical solution of second-order singular boundary value problems*, Symmetry **15**(9) (2023), 1–15. https://doi.org/10.3390/sym15091720

[3] K. Chompuvised and A. Dhamacharoen, *Solving boundary value problems of ordinary differential equations with non-separated boundary conditions*, Appl. Math. Comput. **217**(24) (2011), 10355–10360.

[4] A. Dhamacharen and K. Chompuvised, *An efficient method for solving multipoint equation boundary value problems*, World Academy of Science, Engineering and Technology **75**(2) (2013), 61–65.

[5] N. Mohd Nasir, Z. A. Abdul Majid, F. Ismail and N. Bachok, *Direct integration of boundary value problems using the block method via the shooting technique combined with Steffensen's strategy*, Math. **7**(11) (2019), 1–16.

[6] H. B. Keller, *Existence theory for two point boundary value problems*, Bull. Amer. Math. Soc. **72**(4) (1966), 728–731.

[7] Y. Tao and G. Gao, *Existence, uniqueness and approximation of classical solutions to nonlinear two-point boundary value problems*, Nonlinear Anal. **50**(7) (2002), 981–994. https://doi.org/10.1016/S0362-546X(01)00796-9

[8] H. B. Thompson and C. Tisdell, *Boundary value problems for systems of difference equations associated with systems of second-order ordinary differential equations*, Appl. Math. Lett. **15**(6) (2002), 761–766. https://doi.org/10.1016/S0893-9659(02)00039-3

[9] P. Amodio and I. Sgura, *High-order finite difference schemes for the solution of second-order BVPs*, J. Comput. Appl. Math. **176**(1) (2005), 59–76. https://doi.org/10.1016/j.cam.2004.07.008

[10] H. Ramos and M. A. Rufai, *A third-derivative two-step block Falkner-type method for solving general second-order boundary-value systems*, Math. Comput. Simul. **165** (2019), 139–155. https://doi.org/10.1016/j.matcom.2019.03.003

[11] Z. A. Majid, M. M. Hasni and N. Senu, *Solving second order linear Dirichlet and Neumann boundary value problems by block method*, IAENG Int. J. Appl. Math. **43**(2) (2013), 1–6.

[12] I. Aziz and B. Šarler, *The numerical solution of second-order boundary-value problems by collocation method with the Haar wavelets*, Math. Comput. Model. **52**(9–10) (2010), 1577–1590. https://doi.org/10.1016/j.mcm.2010.06.023

[13] F.-G. Lang and X.-P. Xu, *Quintic B-spline collocation method for second order mixed boundary value problem*, Comput. Phys. Commun. **183**(4) (2012), 913–921. https://doi.org/10.1016/j.cpc.2011.12.017

[14] M. Baccouch, *Analysis of optimal superconvergence of a local discontinuous Galerkin method for nonlinear second-order two-point boundary-value problems*, Appl. Numer. Math. **145**(4) (2019), 361–383. https://doi.org/10.1016/j.apnum.2019.05.003

[15] G. Chuluunbaatar, A. A. Gusev, V. L. Derbov, S. Vinitsky, O. Chuluunbaatar, L. L. Hai and V. Gerdt, *A Maple implementation of the finite element method for solving boundary-value problems for systems of second-order ordinary differential equations*, Commun. Comput. Inf. Sci. **1414** (2021), 152–166.

[16] I. M. Ross, R. J. Proulx and C. F. Borges, *A universal Birkhoff pseudospectral method for solving boundary value problems*, Appl. Math. Comput. **454** (2023), Article ID 128101. https://doi.org/10.1016/j.amc.2023.128101

[17] S. N. Ha, *A nonlinear shooting method for two-point boundary value problems*, Comput. Math. Appl. **42**(10–11) (2001), 1411–1420.

[18] S. Kheybari, M. T. Darvishi, F. Alizadeh, S. M. Boulaaras and K. Hosseini, *A modified scheme to the multiple shooting method for BVPs*, Alexandria Eng. J. **118** (2025), 649–663. https://doi.org/10.1016/j.aej.2025.01.070

[19] B. Jang, *Two-point boundary value problems by the extended Adomian decomposition method*, J. Comput. Appl. Math. **219**(1) (2008), 253–262. https://doi.org/10.1016/j.cam.2007.07.036

[20] S. N. Jator and J. Li, *Boundary value methods via a multistep method with variable coefficients for second order initial and boundary value problems*, Int. J. Pure Appl. Math. **50**(3) (2009), 403–420.

[21] H. Ramos and G. Singh, *Solving second order two-point boundary value problems accurately by a third derivative hybrid block integrator*, Appl. Math. Comput. **421**(3) (2022), 1–19. https://doi.org/10.1016/j.amc.2022.126960

[22] J. Li and X. Wang, *Multi-step Runge-Kutta-Nyström methods for special second-order initial value problems*, Appl. Numer. Math. **113** (2017), 54–70. https://doi.org/10.1016/j.apnum.2016.11.002

[23] J. Li, S. Deng and X. Wang, *Extended explicit pseudo two-step RKN methods for oscillatory systems* $y'' + My = f(y)$, Numer. Algorithms **78** (2018), 673–700. https://doi.org/10.1007/s11075-017-0395-9

[24] A. Abdulsalam, N. Senu, Z. A. Majid and N. M. A. Nik Long, *Higher-order multi-step Runge-Kutta-Nyström methods with frequency-dependent coefficients for second-order initial value problem* $u'' = f(x, u, u')$, Comput. Methods Differ. Equ. **12**(4) (2024), 808–826. https://doi.org/10.22034/cmde.2024.57684.2418

[25] J. Li and S. Deng, *Trigonometrically fitted multi-step RKN methods for second-order oscillatory initial value problems*, Appl. Math. Comput. **320** (2018), 740–753. https://doi.org/10.1016/j.amc.2017.10.036

[26] J. Li, S. Deng and X. Wang, *Multi-step Nyström methods for general second-order initial value problems* $y''(t) = f(t, y(t), y'(t))$, Int. J. Comput. Math. **96**(6) (2019), 1254–1277. https://doi.org/10.1080/00207160.2018.1464154

[27] A. Abdulsalam, N. Senu, Z. A. Majid and N. M. A. Nik Long, *Development of high-order adaptive multi-step Runge-Kutta-Nyström method for solving special second-order ODEs*, Math. Comput. Simul. **216** (2024), 104–125. https://doi.org/10.1016/j.matcom.2023.09.006

[28] T. S. Mohamed, N. Senu, Z. B. Ibrahim and N. M. A. Nik Long, *Efficient two-derivative Runge-Kutta-Nyström methods for solving general second-order ordinary differential equations* $y''(x) = f(x, y, y')$, Discrete Dyn. Nat. Soc. **2018** (2018), 1–10. https://doi.org/10.1155/2018/2393015

[29] K. Hussain, F. Ismail and N. Senu, *Two embedded pairs of Runge-Kutta type methods for direct solution of special fourth-order ordinary differential equations*, Math. Probl. Eng. **2015** (2015), 1–13. https://doi.org/10.1155/2015/196595

[30] J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, Wiley, Chichester, 1973.

[31] S. O. Fatunla, *Block methods for second order ODEs*, Int. J. Comput. Math. **41**(2) (1991), 55–63.

[32] T. E. Simos and C. Tsitouras, *On high order Runge-Kutta-Nyström pairs*, J. Comput. Appl. Math. **400**(2) (2022), 1–10. https://doi.org/10.1016/j.cam.2021.113753

[33] M. El-Mikkawy and E. D. Rahmo, *A new optimized non-FSAL embedded Runge-Kutta-Nyström algorithm of orders* 6 *and* 4 *in six stages*, Appl. Math. Comput. **145**(1) (2003), 33–43. https://doi.org/10.1016/S0096-3003(02)00436-8

[34] F. A. Fawzi, N. Senu, F. Ismail and Z. A. Majid, *An embedded* 6(5) *pair of explicit Runge-Kutta method for periodic IVPs*, Far East J. Math. Sci. **100**(11) (2016), 1841–1857. http://dx.doi.org/10.17654/MS100111841

[35] T. E. Simos and C. Tsitouras, *Evolutionary derivation of Runge-Kutta pairs for addressing inhomogeneous linear problems*, Numer. Algorithms **87**(2) (2021), 511–525.

[36] S. K. Bhatta and K. S. Sastri, *Symmetric spline procedures for boundary value problems with mixed boundary conditions*, J. Comput. Appl. Math. **45** (1993), 237–250. https://doi.org/10.1016/0377-0427(93)90043-B

[1]Department of Mathematics and Computer Applications,
College of Sciences,
Al-Nahrain University, Jadriya, Baghdad, Iraq
Email address: azraasalam55@gmail.com, athraa.a.s@nahrainuniv.edu.iq

[2]Institute for Mathematical Research,
Universiti Putra Malaysia,
UPM, Serdang, 43400, Selangor, Malaysia
Email address: norazak@upm.edu.my

[3]Department of Mathematics and Statistics,
Universiti Putra Malaysia,
UPM, Serdang, 43400, Selangor, Malaysia

[4]Department of Mechanical, Industrial, and Mechatronics Engineering,
Toronto Metropolitan University,
Toronto, Canada, M5B 2K3

[5]Center of Mathematical Sciences,
Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

[6]General Directorate of Wasit Education, Iraq