Applied Mathematics and Computation 421 (2022) 126960

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Solving second order two-point boundary value problems accurately by a third derivative hybrid block integrator

Higinio Ramos[a,b], Gurjinder Singh[c,*]

[a] Scientific Computing Group, Universidad de Salamanca, Plaza de la Merced 37008 Salamanca, Spain
[b] Spain Escuela Politécnica Superior de Zamora, Universidad de Salamanca, Campus Viriato, 49022 Zamora, Spain
[c] Department of Mathematical Sciences, I. K. Gujral Punjab Technical University, Jalandhar, Main Campus, Kapurthala 144603, Punjab, India

A R T I C L E I N F O

Article history:
Received 30 July 2019
Revised 29 December 2021
Accepted 17 January 2022

Keywords:
Ordinary differential equations
Boundary value problems
Block scheme
Convergence
Optimization strategy

A B S T R A C T

This article deals with the development of an optimized third-derivative hybrid block method for integrating general second order two-point boundary value problems (BVPs) subject to different types of boundary conditions (BCs) such as Dirichlet, Neumann or Robin. A purely interpolation and collocation approach has been used in order to develop the method. A constructive approach has been applied in the development of the method to consider two off-step optimal points among an infinite number of possible choices in a two-step block corresponding to a generic interval of the form $[x_n, x_{n+2}]$. The obtained method simultaneously produces an approximate solution over the entire integration interval. Some numerical experiments have been presented that show the good performance of the presented scheme.

1. Introduction

The field of numerical analysis of differential equations is continuously growing due to the gradual development of new models of real-world phenomena. Due to the unavailability of analytical solutions for most differential systems, it is necessary to obtain numerical approximations to the solutions. It is a well-known fact that existing approaches for solving differential equations are modified as the perspectives change or new techniques are developed to get approximate solutions more accurately and efficiently (see [1–54]).

Our goal in this article is to develop an efficient two-step block method in global sense (that produces approximate solutions simultaneously at all nodal points in an interval of interest) and show its good performance in solving second order two-point BVPs of ordinary differential equations (ODEs) of the form

$$u''(x) = f(x, u(x), u'(x)), \quad x \in [a, b], \tag{1}$$

with any one of the given possible types of BCs in Table 1:

Before proceeding, we assume that the equation in (1) together with the given boundary conditions satisfy the requirements that ensure the existence and uniqueness of the true solution (see [1–3]), namely, we assume that the function f is continuous in $[a, b] \times \mathbf{R}^2$ and verifies a Lipschitz condition in the variable $\mathbf{u} = (u, u')$, that is, it holds that for any $\mathbf{u}_1, \mathbf{u}_2 \in \mathbf{R}^2$

* Corresponding author.
E-mail addresses: higra@usal.es (H. Ramos), gurjinder@ptu.ac.in (G. Singh).

https://doi.org/10.1016/j.amc.2022.126960

there exist constants $L_j > 0,\ j = 0, 1,$ such that

$$|f(x, \mathbf{u}_1) - f(x, \mathbf{u}_2)| \le L_0|u_1 - u_2| + L_1|u_1' - u_2'|.$$

There are well-known approaches for solving a BVP numerically, as (i) the shooting approach [6], (ii) finite difference approaches [7,12–16], and (iii) approximate methods based on the idea of using linear relations of trial functions (e.g. collocation, Galerkin's technique and Rayleigh-Ritz scheme, etc.), [4,17–24]. A good collection of references on collocation methods can be found in [4]. In [8], the authors discussed a method based on a spline collocation approach for integrating mixed order systems of boundary value problems and in [9] a general purpose code COLSYS has been discussed for mixed order systems of BVPs in ODEs. In the shooting approach, a given BVP is transformed into a system consisting of first order initial value ODEs. Furthermore, the resulting system is solved with any available ODE solver, for example Runge-Kutta or linear multi-step methods. A major difficulty with the shooting approach is that sometimes a well-behaved BVP is transformed, requiring later the integration of an initial value problem which is unstable [25]. More precisely, the true solution of a BVP can be stable to some perturbations in the boundary conditions, but the solutions of the initial value problems arising in the shooting approach are unstable to perturbations of the initial values.

Some efficient codes have been developed for first order systems, for instance, Cash et al. [29] developed codes based on Mono Implicit Runge-Kutta and Lobatto schemes. Shampine et al. discussed in [30] a user friendly FORTRAN code for solving BVPs.

The approaches based on finite differences transform a given continuous ODE into a system of equations. After solving this system, one can get approximate solutions at all nodal points of interest at once. Nowadays, Computer Algebra Systems (CAS) usually incorporate routines for solving BVPs. For example, in MATLAB, some built-in solvers are [25]: bvp4c (fourth-order finite difference code) and bvp5c (fifth-order finite difference code), which are available to deal with a BVP numerically. These solvers first reformulate the given BVP into a system consisting of ODEs of first order and then attempt to solve it numerically. In [10], an efficient MATLAB code for solving two point boundary value problems using the codes twpbvp, twpbvpl and acdc is presented. A finite difference code for solving second order singular perturbation problems numerically has been proposed in [11] where a MATLAB code based on high order finite difference schemes approximates directly the original problem without reformulating it as a first order system (in fact this method is based also on the boundary value approach). On the website https://archimede.dm.uniba.it/~bvpsolvers/testsetbvpsolvers/ different codes in FORTRAN, R and the MATLAB environment are available for solving BVPs and other relevant details can be found in [26–28]. Currently, boundary value methods are also used for solving BVPs. For a good collection of references on these methods, one can consult Brugnano et al. [33,34] and references therein. Global methods based on boundary value methods generalized for solving BVPs can be found in [31,32]. Block methods, which were initially used for obtaining starting values for linear multi-step initial value solvers, can be extended to solve boundary value problems as well [35–42,48]. See et al. proposed in [49] a three-step block scheme of Adam's type for integrating nonlinear two-point BVPs of Dirichlet and Neumann-type. Biala [50] proposed a new class of linear multi-step methods using the theory of interpolation and collocation. Further, these linear multi-step methods were implemented as boundary value methods and block unification methods. Biala and Jator [51] also proposed a new family of boundary value methods with continuous coefficients and applied them via the block unification approach to solve BVPs. In [52–54] different approaches based on block methods have been used for solving higher order BVPs.

In this article, we shall be concerned firstly with the development of a third derivative hybrid block method for solving (1) in a global sense. The hybrid methods were developed for solving initial value problems in order to bypass the first Dahlquist barrier on linear multi-step initial value solvers. We have explored both the ideas of hybrid and block methods in this article in developing the numerical scheme. We have considered two off-step points in a two-step block interval $[x_n, x_{n+2}]$ of nodal points and impose the vanishing of the principal terms of the local truncation errors of the main formulas, as was done for initial value problems in the seminal paper by Ramos et al. [42]. In this way we get optimized values of these off-step points and get a hybrid block method in optimized version for solving the problem in (1) numerically (together with the given BCs).

The rest of the article is concerned with the analysis of the order of convergence of the formulas, and a detailed analysis of the convergence of the new scheme. Finally, some numerical experiments are presented in order to show the good performance of the proposed scheme. The article ends with some conclusions of the present work.

2. Derivation of an optimized hybrid block method

This section is concerned with the development of an optimized two-step block method for numerically solving (1) with any of the BC in Table 1. Firstly, we discretize the interval of interest taking $a = x_0 < x_1 < x_2 < \ldots < x_N = b$, where the nodal points are $x_j = a + jh;\ \ j = 0, 1, 2, 3, \ldots, N$, with N even, and $h = (b-a)/N$ the fixed step size. We are interested in approximating the solution at those points. To proceed further, consider a generic two-step interval of the nodal points of the form $[x_n, x_{n+2}]$. On this interval, we consider a polynomial $q(x)$ that approximates the true solution $u(x)$ of (1), that is,

$$u(x) \approx q(x) = \sum_{j=0}^{8} c_j \Phi_j(x), \tag{2}$$

Table 1
Different types of BCs.

BCs	*Type*
$u(a) = u_a, \quad u(b) = u_b$	Dirichlet
$u'(a) = u'_a, \quad u'(b) = u'_b$	Neumann
$\psi(u(a), u'(a)) = \psi_a, \quad \psi(u(b), u'(b)) = \psi_b$	Mixed

where the basis functions are $\Phi_j(x) = (x - x_n)^j$, and the c_j are unknown coefficients that will be determined by imposing the following interpolating and collocation conditions on $q(x)$:

$$\left.\begin{array}{ll} (i) & u_n = q(x_n) \\ (ii) & u'_n = q'(x_n) \\ (iii) & f_{n+i} = q''(x_{n+i}), \quad i = 0, r, 1, s, 2 \\ (iv) & g_{n+i} = q'''(x_{n+i}), \quad i = 0, 2 \end{array}\right\} \tag{3}$$

where the notations $u_n, u'_n, f_{n+i} = f(x_{n+i}, u_{n+i}, u'_{n+i}), g_{n+i} = g(x_{n+i}, u_{n+i}, u'_{n+i})$ are approximations of $u(x_n), u'(x_n), u''(x_{n+i})$ and $u'''(x_{n+i})$, respectively, with

$$g(x, u, u') = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial u}u' + \frac{\partial f}{\partial u'}f(x, u, u'),$$

and $u_{n+i} \approx u(x_{n+i}), u'_{n+i} \approx u'(x_{n+i})$. In the above expressions, x_{n+r} and x_{n+s} are the designations of two intermediate points in the block $[x_n, x_{n+2}]$ with the requirement $0 < r < 1 < s < 2$.

The conditions in (3) give a system of nine equations in nine unknowns, which can be easily solved by any Computer Algebra System (CAS) like *Matlab* or *Mathematica*, providing the values of the unknown coefficients $c_j,\ j = 0(1)8$. By substituting these values in (2), we get a continuous formula expressed in the form

$$\begin{array}{l} u(x) \approx q(x) = \alpha_0(x)u_n + \alpha_1(x)hu'_n + h^2 \sum_i \beta_i(x) f_{n+i} + h^3 \sum_j \gamma_j(x) g_{n+j}, \\ \quad i = 0, r, 1, s, 2; \;\; j = 0, 2. \end{array} \tag{4}$$

where $\alpha_0(x), \alpha_1(x),\ \beta_i(x), \gamma_j(x)$ are continuous coefficients.

2.1. Main formulas

In order to get the approximate values u_{n+2} and u'_{n+2}, we evaluate the expression of $q(x)$ given in (4) and its first derivative $q'(x)$ at $x = x_n + 2h$. Those approximations will be expressed in terms of the unknown parameters r and s. Now, in order to get appropriate values of r and s, we expand these formulas in (4) for u_{n+2} and u'_{n+2} using Taylor series about the point x_n. In this way, we get the local truncation errors of these formulas, which are given, respectively, by

$$LTE(u(x_{n+2}), h) = \frac{(2 - 3rs)u^{(9)}(x_n)h^9}{99225} + \mathcal{O}(h^{10}), \tag{5}$$

and

$$LTE(u'(x_{n+2}), h) = \frac{(2 - r - s)u^{(9)}(x_n)h^8}{33075} + \mathcal{O}(h^9). \tag{6}$$

Imposing that the principal terms in (5) and (6) vanish, we get a nonlinear system of equations given by

$$\begin{cases} 2 - 3rs & = 0 \\ 2 - r - s & = 0. \end{cases}$$

One can verify that the above system of equations has a unique solution with $0 < r < 1 < s < 2$. Solving this system, we obtain the optimized values of r and s as follows

$$r = 1 - \frac{\sqrt{3}}{3} \simeq 0.42265, \qquad s = 1 + \frac{\sqrt{3}}{3} \simeq 1.57735. \tag{7}$$

By inserting the above values of r and s in (5) and (6), we obtain

$$LTE(u(x_{n+2}), h) = \frac{-u^{(11)}(x_n)h^{11}}{58939650} + \mathcal{O}(h^{12}),$$
$$LTE(u'(x_{n+2}), h) = \frac{u^{(12)}(x_n)h^{11}}{589396500} + \mathcal{O}(h^{12}).$$

Using (7) we get from (4) the following two main formulas

$$u_{n+2} = u_n + 2hu'_n + \frac{h^2}{105}\Big(37f_n + (54+18\sqrt{3})f_{n+r} + 64f_{n+1} + (54-18\sqrt{3})f_{n+s} + f_{n+2} + 2hg_n\Big),$$

$$u'_{n+2} = u'_n + \frac{h}{105}\left(19f_n + 54f_{n+r} + 64f_{n+1} + 54f_{n+s} + 19f_{n+2} + h(g_n - g_{n+2})\right). \tag{8}$$

2.2. Additional formulas

Note that to obtain a discrete approximation of the true solution on the interval $[x_n, x_{n+2}]$ we have two formulas so far, but there are ten unknowns (the values of $u_n, u_{n+r}, u_{n+1}, u_{n+s}, u_{n+2}$ and those of $u'_n, u'_{n+r}, u'_{n+1}, u'_{n+s}, u'_{n+2}$). Thus, we consider additional formulas by evaluating $q(x)$ given in (4) and its first derivative $q'(x)$, at $x_{n+r}, x_{n+1}, x_{n+s}$.

In this way, one gets eight formulas to approximate the solution and the first derivative, that can be written using the following block formulations. For the approximate values of the solution we have

$$\begin{pmatrix} u_{n+r} \\ u_{n+1} \\ u_{n+s} \\ u_{n+2} \end{pmatrix} = \mathbf{e}u_n + \mathbf{c}hu'_n + h^2 F \begin{pmatrix} f_n \\ f_{n+r} \\ f_{n+1} \\ f_{n+s} \\ f_{n+2} \end{pmatrix} + h^3 G \begin{pmatrix} g_n \\ g_{n+2} \end{pmatrix}, \tag{9}$$

where $\mathbf{e} = (1,1,1,1)^T$, $\mathbf{c} = (r,1,s,2)^T$,

$$F = \begin{pmatrix} \frac{1925-683\sqrt{3}}{11340} & \frac{1}{36} & \frac{482-288\sqrt{3}}{2835} & \frac{379}{1260} - \frac{6\sqrt{3}}{35} & \frac{-19-\sqrt{3}}{11340} \\ \frac{1171}{6720} & \frac{9}{64} + \frac{3\sqrt{3}}{35} & \frac{1}{24} & \frac{9}{64} - \frac{3\sqrt{3}}{35} & \frac{19}{6720} \\ \frac{1925+683\sqrt{3}}{11340} & \frac{379}{1260} + \frac{6\sqrt{3}}{35} & \frac{2(241+144\sqrt{3})}{2835} & \frac{1}{36} & \frac{-19+\sqrt{3}}{11340} \\ \frac{37}{105} & \frac{6(3+\sqrt{3})}{35} & \frac{64}{105} & \frac{6(3-\sqrt{3})}{35} & \frac{1}{105} \end{pmatrix},$$

$$G = \begin{pmatrix} \frac{106-35\sqrt{3}}{11340} & \frac{2+\sqrt{3}}{11340} \\ \frac{67}{6720} & -\frac{1}{2240} \\ \frac{106+35\sqrt{3}}{11340} & \frac{2-\sqrt{3}}{11340} \\ \frac{2}{105} & 0 \end{pmatrix};$$

and for the approximate values of the derivative, it is

$$h\begin{pmatrix} u'_{n+r} \\ u'_{n+1} \\ u'_{n+s} \\ u'_{n+2} \end{pmatrix} = \mathbf{e}hu'_n + h^2 F' \begin{pmatrix} f_n \\ f_{n+r} \\ f_{n+1} \\ f_{n+s} \\ f_{n+2} \end{pmatrix} + h^3 G' \begin{pmatrix} g_n \\ g_{n+2} \end{pmatrix}, \tag{10}$$

where

$$F' = \begin{pmatrix} \frac{797+44\sqrt{3}}{3780} & \frac{81-8\sqrt{3}}{315} & \frac{8(36-23\sqrt{3})}{945} & \frac{81-43\sqrt{3}}{315} & \frac{-113+44\sqrt{3}}{3780} \\ \frac{257}{1680} & \frac{9}{35} + \frac{3\sqrt{3}}{16} & \frac{32}{105} & \frac{9}{35} - \frac{3\sqrt{3}}{16} & \frac{47}{1680} \\ \frac{797-44\sqrt{3}}{3780} & \frac{81+43\sqrt{3}}{315} & \frac{8(36+23\sqrt{3})}{945} & \frac{81+8\sqrt{3}}{315} & \frac{-113-44\sqrt{3}}{3780} \\ \frac{19}{105} & \frac{18}{35} & \frac{64}{105} & \frac{18}{35} & \frac{19}{105} \end{pmatrix},$$

$$G' = \begin{pmatrix} \frac{53+6\sqrt{3}}{3780} & \frac{17-6\sqrt{3}}{3780} \\ \frac{1}{210} & -\frac{1}{210} \\ \frac{53-6\sqrt{3}}{3780} & \frac{17+6\sqrt{3}}{3780} \\ \frac{1}{105} & -\frac{1}{105} \end{pmatrix}.$$

Note that the block method given by formulas (9)-(10) could be used for solving an initial value problem, provided that the initial values are given, but in this form it is not adequate for solving a BVP. In order to get a discrete solution of the BVP given in (1) with two BCs (any one of the types given in Table 1), we shall consider the formulas (9)-(10) for the values of $n = 0(2)N-2$, altogether with the two given boundary conditions. In this way, we get a global method consisting of a system of $4N+2$ equations in the $4N+2$ unknowns

$$\{u_0, u_r, u_1, u_s, u_2, u_{2+r}, u_3, u_{2+s}, u_4, \ldots, u_N\},$$

$$\left\{u'_0, u'_r, u'_1, u'_s, u'_2, u'_{2+r}, u'_3, u'_{2+s}, u'_4, \ldots, u'_N\right\}.$$

The resulting system of equations could be linear or nonlinear according to the nature of f in the Eq. (1) and the given boundary conditions. We shall consider two possibilities as follows

(i) **Linear system:** If the resulting system is of linear nature, then one can consider existing linear system solvers to deal with the situation.
(ii) **Nonlinear system:** If the resulting system is of nonlinear nature, one can consider Newton-Raphson's-type or other existing nonlinear system solvers to tackle the situation.

Note. In case of a nonlinear system, for implementing the iterative method considered it is necessary to have some good starting values in order to start the iterative process. In the implementation section, we shall discuss some criteria of choosing good initial approximations of the solution.

3. Theoretical analysis of the proposed scheme

In this section, we shall be concerned with some characteristics of the obtained formulas and the convergence analysis of the global method obtained from (9)-(10) altogether with the boundary conditions.

3.1. Time reversal symmetry

The time reversal symmetry (TRS) property is an important condition for a numerical method to maintain a symmetry feature of the solution of a given problem. This property has been studied extensively in the case of Hamiltonian problems [45,46]. Some well-known methods that preserve this property are the midpoint method, the Trapezoidal Rule or the Verlet method [47].

Let us take the problem in (1) with the first set of boundary conditions in Table 1

$$\begin{cases} u''(x) = f(x, u(x), u'(x)), \quad x \in [a,b], \\ u(a) = u_a, \quad u(b) = u_b\,. \end{cases} \tag{11}$$

Making the substitution $t = a + b - x$ we get

$$\begin{cases} z''(t) = k(t, z(t), z'(t)), \quad t \in [a,b], \\ z(a) = u_b, \quad z(b) = u_a, \end{cases} \tag{12}$$

where $z(t) \equiv u(a+b-t)$ and $k(t, z(t), z'(t)) \equiv f(a+b-t, z(t), -z'(t))$.

On the mesh $x_i = a + ih, i \in J = \bigcup_{l=0,l \text{ even}}^{N-2} (\{0, r, 1, s, 2\} + l)$ and on the corresponding mesh $\{t_i\}_{i \in J}$ with $t_i = a + b - x_i$, considering the two vectors

$$\hat{U} = \begin{pmatrix} u_a \\ u(x_r) \\ u(x_1) \\ u(x_s) \\ \vdots \\ u(x_{N-2}) \\ u(x_{N-2+r}) \\ u(x_{N-1}) \\ u(x_{N-2+s}) \\ u_b \end{pmatrix}, \qquad Z = \begin{pmatrix} u_b \\ z(t_{N-2+s}) \\ z(t_{N-1}) \\ z(t_{N-2+r}) \\ \vdots \\ z(t_s) \\ z(t_1) \\ z(t_r) \\ u_a \end{pmatrix},$$

it is straightforward to check that

$$Z = P_{2N+1} \hat{U}, \tag{13}$$

where P_{2N+1} is the anti-diagonal identity matrix of order $2N+1$

$$P_{2N+1} = \begin{pmatrix} & & 1 \\ & \cdot^{\cdot^{\cdot}} & \\ 1 & & \end{pmatrix}.$$

If the approximate solutions of problems (11) and (12) provided by the proposed numerical method verify a discrete analogous to (13), then it is said that the method satisfies the time reversal symmetry. What one usually wants in this situation is to preserve a discrete analog of time symmetry, namely, if the time discretization is applied to solve first forward and then backward in time, the boundary conditions are recovered. For the proposed method this is true if it is invariant under the following substitutions

$$h \to -h, \quad u_i \to u_{N-i}, i \in J\,.$$

In case a numerical method is formulated by means of symmetric formulas this is evident. For the formulas in (9)-(10) one could try to get a symmetric formulation, as for example the formulas in (8) may be expressed equivalently as

$$u_{n+2} = u_n + h(u'_n + u'_{n+2}) + \frac{6h^2}{35}\left(\sqrt{3}(f_{n+r} - f_{n+s}) + f_n - f_{n+2}\right) + \frac{h^3}{105}(g_n + g_{n+2})$$

$$u'_{n+2} = u'_n + \frac{h}{105}(54(f_{n+r} + f_{n+s}) + 64 f_{n+1} + 19(f_n + f_{n+2})) + \frac{h^2}{105}(g_n - g_{n+2})$$

With the other formulas in (9) one could try to look for a symmetric equivalent formulation, but in fact it is not necessary. Taking in mind the identity $2 - r = s$, the substitutions $h \to -h,\ u_{n+i} \to u_{n+2-i}, i = 0, r, 1, s, 2,$ give after some calculus the same formulas, thus the proposed method verifies the time reversal symmetry.

3.2. Order of convergence

The block method given by the formulas in (9)-(10) may be arranged in the following matrix form

$$\mathbf{\Lambda_1\ U_n = h\ \Lambda_2\ U'_n + h^2\ \Lambda_3\ F_n + h^3\ \Lambda_4\ G_n}, \tag{14}$$

where $\mathbf{\Lambda_1}, \mathbf{\Lambda_2}, \mathbf{\Lambda_3}$ and $\mathbf{\Lambda_4}$ are matrices of coefficients of dimensions 8×5, that can be easily obtained from the formulas, and

$$\begin{aligned}
\mathbf{U_n} &= (u_n, u_{n+r}, u_{n+1}, u_{n+s}, u_{n+2})^T,\\
\mathbf{U'_n} &= (u'_n, u'_{n+r}, u'_{n+1}, u'_{n+s}, u'_{n+2})^T,\\
\mathbf{F_n} &= (f_n, f_{n+r}, f_{n+1}, f_{n+s}, f_{n+2})^T,\\
\mathbf{G_n} &= (g_n, g_{n+r}, g_{n+1}, g_{n+s}, g_{n+2})^T.
\end{aligned}$$

Let $Z(x)$ be an analytical function. We consider the following difference operator associated with the formulas in (9)-(10)

$$\begin{aligned}&\bar{\mathcal{L}}[Z(x); h] = \textstyle\sum_j \bar{\alpha}_j Z(x + jh) - h\bar{\beta}_j Z'(x + jh) - h^2 \bar{\gamma}_j Z''(x + jh) - h^3 \bar{\delta}_j Z'''(x + jh),\\ &j = 0, r, 1, s, 2\end{aligned} \tag{15}$$

where $\bar{\alpha}_j, \bar{\beta}_j, \bar{\gamma}_j, \bar{\delta}_j$ are the corresponding columns of $\mathbf{\Lambda_1}, \mathbf{\Lambda_2}, \mathbf{\Lambda_3}$ and $\mathbf{\Lambda_4}$. Both the new proposed scheme and the operator (15) are said to have order p if using the Taylor series representation of $Z(x_n + jh), Z'(x_n + jh), Z''(x_n + jh)$ and $Z'''(x_n + jh)$ around the nodal point x_n, we get

$$\bar{\mathcal{L}}[Z(x_n); h] = \bar{\nu}_0 Z(x_n) + \bar{\nu}_1 h Z'(x_n) + \bar{\nu}_2 h^2 Z''(x_n) + \cdots + \bar{\nu}_q h^q Z^{(q)}(x_n) + \ldots$$

with $\bar{\nu}_0 = \bar{\nu}_1 = \bar{\nu}_2 = \cdots = \bar{\nu}_{p+1} = 0$ and $\bar{\nu}_{p+2} \neq 0$. Note that in the above expression, $\bar{\nu}_i$ are vectors and $\bar{\nu}_{p+2}$ stands for the vector of error constants. In the case of the proposed block method, we have $\bar{\nu}_0 = \bar{\nu}_1 = \cdots = \bar{\nu}_8 = 0$, and

$$\bar{\nu}_9 = \left(\frac{-1}{1837080\sqrt{3}}, 0, \frac{1}{1837080\sqrt{3}}, 0, \frac{-1}{612360}, \frac{1}{362880}, \frac{-1}{612360}, 0\right)^T.$$

In view of the components of vector $\bar{\nu}_9$ it is appropriate to point out here that the local truncation error for the formula that approximates $u(x_{n+1})$ is

$$LTE(u(x_{n+1}), h) = \frac{(3r(64s - 29) - 87s + 46)u^{(9)}(x_n)h^9}{12700800} + \mathcal{O}(h^{10}),$$

which for the optimized values of the parameters in (7) results in

$$LTE(u(x_{n+1}), h) = -\frac{u^{(10)}(x_n)h^{10}}{14515200} + \mathcal{O}(h^{11}).$$

This establishes that the third derivative block method given by the formulas in (9)-(10) has a seventh-order of convergence. In the next section we will see that this behaviour is maintained when the method is applied in the global form indicated above for solving a BVP.

3.3. Linear stability analysis

To analyze the practical performance of the proposed method concerning stability we consider the following test problem (see [21])

$$u'' + \mu u' = 0, \quad u(a) = u_a,\ u(b) = u_b,$$

whose exact solution is given by $u(x) = A + B\exp(-\mu x)$, where A and B are arbitrary constants that are determined through the boundary conditions.

If we apply the formulas in (9)-(10) to the above problem the very least that we expect of the finite difference solutions is that they behave monotonically decreasing as $\exp(-\mu x)$ for $\mu > 0$. After applying the method to the test problem and setting $z = \mu h$ it results that it may be arranged in vector form as

$$P\begin{pmatrix} u_{n+r} \\ u_{n+1} \\ u_{n+s} \\ u_{n+2} \\ hu'_{n+r} \\ hu'_{n+1} \\ hu'_{n+s} \\ hu'_{n+2} \end{pmatrix} = Q\begin{pmatrix} u_{n+r-2} \\ u_{n-1} \\ u_{n+s-2} \\ u_{n} \\ hu'_{n+r-2} \\ hu'_{n-1} \\ hu'_{n+s-2} \\ hu'_{n} \end{pmatrix}$$

where P is the following matrix

$$P = \begin{pmatrix}
1 & 0 & 0 & 0 & \frac{z}{36} & \frac{(482-288\sqrt{3})z}{2835} & \left(\frac{379}{1260} - \frac{6\sqrt{3}}{35}\right)z & -\frac{z\left(\left(2+\sqrt{3}\right)z+\sqrt{3}+19\right)}{11340} \\
0 & 1 & 0 & 0 & \left(\frac{9}{64} + \frac{3\sqrt{3}}{35}\right)z & \frac{z}{24} & \left(\frac{9}{64} - \frac{3\sqrt{3}}{35}\right)z & \frac{z(3z+19)}{6720} \\
0 & 0 & 1 & 0 & \frac{(379+216\sqrt{3})z}{1260} & \frac{2(241+144\sqrt{3})z}{2835} & \frac{z}{36} & \frac{z\left(\left(-2+\sqrt{3}\right)z+\sqrt{3}-19\right)}{11340} \\
0 & 0 & 0 & 1 & \frac{6}{35}\left(3+\sqrt{3}\right)z & \frac{64z}{105} & \frac{6\left(3-\sqrt{3}\right)}{35}z & \frac{z}{105} \\
0 & 0 & 0 & 0 & \frac{81-8\sqrt{3}}{315}z+1 & \frac{8\left(36-23\sqrt{3}\right)}{945}z & \frac{\left(81-43\sqrt{3}\right)}{315}z & \frac{z\left(\left(6\sqrt{3}-17\right)z+44\sqrt{3}-113\right)}{3780} \\
0 & 0 & 0 & 0 & \left(\frac{9}{35} + \frac{3\sqrt{3}}{16}\right)z & \frac{32z}{105}+1 & \left(\frac{9}{35} - \frac{3\sqrt{3}}{16}\right)z & \frac{z(8z+47)}{1680} \\
0 & 0 & 0 & 0 & \frac{81+43\sqrt{3}}{315}z & \frac{8\left(36+23\sqrt{3}\right)}{945}z & \frac{\left(81+8\sqrt{3}\right)}{315}z+1 & -\frac{z\left(\left(17+6\sqrt{3}\right)z+44\sqrt{3}+113\right)}{3780} \\
0 & 0 & 0 & 0 & \frac{18z}{35} & \frac{64z}{105} & \frac{18z}{35} & \frac{1}{105}z(z+19)+1
\end{pmatrix}$$

and the matrix Q is given by

$$Q = \begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & \frac{z\left(\left(106-35\sqrt{3}\right)z+683\sqrt{3}-1925\right)}{11340} - \frac{1}{\sqrt{3}} + 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & \frac{z(67z-1171)}{6720} + 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & \frac{z\left(\left(106+35\sqrt{3}\right)z-683\sqrt{3}-1925\right)}{11340} + \frac{1}{\sqrt{3}} + 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & \frac{1}{105}z(2z-37)+2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{z\left(\left(53+6\sqrt{3}\right)z-44\sqrt{3}-797\right)}{3780} + 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{z(8z-257)}{1680} + 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{z\left(\left(53-6\sqrt{3}\right)z+44\sqrt{3}-797\right)}{3780} + 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{105}(z-19)z+1
\end{pmatrix}.$$

This may be written finally as

$$\begin{pmatrix} u_{n+r} \\ u_{n+1} \\ u_{n+s} \\ u_{n+2} \\ hu'_{n+r} \\ hu'_{n+1} \\ hu'_{n+s} \\ hu'_{n+2} \end{pmatrix} = R(z)\begin{pmatrix} u_{n+r-2} \\ u_{n-1} \\ u_{n+s-2} \\ u_{n} \\ hu'_{n+r-2} \\ hu'_{n-1} \\ hu'_{n+s-2} \\ hu'_{n} \end{pmatrix}$$

where $R(z) = P^{-1}Q$ is the stability matrix.

The behavior of the numerical solution will depend on the eigenvalues of this matrix, which determine the stability properties of the method. The two eigenvalues of R different from zero are

$$\rho_1 = 1, \quad \rho_2 = \frac{-z^5 + 15z^4 - 105z^3 + 420z^2 - 945z + 945}{z^5 + 15z^4 + 105z^3 + 420z^2 + 945z + 945},$$

and the region of stability is the region in the complex plane where it is $|\rho_2| < 1$. This stability region turns out to be $\Re(z) > 0$.

4. Convergence analysis

From a practical point of view, the numerical approximations obtained by a numerical method for solving a given differential equation must have a convergent behavior. This subsection addresses the convergence theorem for the proposed global scheme for solving BVPs. Firstly, we shall state the definition of convergence of a numerical method for solving a BVP.

Definition 4.1. Let $u(x)$ be the true solution of a BVP given in (1) with any of the boundary conditions in Table 1, and $\{u_j\}_{j=0}^N$ the numerical approximations of $u(x)$ obtained by the proposed global method. The numerical method is said to have pth-order of convergence if for a sufficiently small step size h, there exists a constant K (independent of h) such that

$$\max_{0\le j\le N} \|u(x_j) - u_j\| \le Kh^p.$$

Note that from the above definition, we shall have

$$\max_{0\le j\le N} \|u(x_j) - u_j\| \to 0 \text{ as } h \to 0.$$

Theorem 4.1 (Convergence theorem)**.** *Let $u(x)$ be the true solution of the BVP in (1) with any of the boundary conditions in Table 1, and $\{u_j\}_{j=0}^N$ the discrete solution provided by the proposed global method. Then, assuming that $u(x)$ is sufficiently differentiable (up to order twelve) with bounded derivatives, the proposed method is seventh order convergent.*

Proof. For the sake of convenience, we shall consider the BVP given by the equation in (1) with boundary conditions of Dirichlet-type, which are very common in use. The theorem can be proved for other types of boundary conditions in a similar way by making appropriate changes.

Firstly, let us suppose that we have exact known values provided by the BCs, that is, $u_0 = u(x_0) = u_a$ and $u_N = u(x_N) = u_b$. Therefore, the unknowns in the global method are

$$\{u_r, u_1, u_s, u_2, u_{2+r}, u_3, u_{2+s}, u_4, \ldots, u_{N-1}, u_{N-2+s}\},$$

$$\left\{u'_0, u'_r, u'_1, u'_s, u'_2, u'_{2+r}, u'_3, u'_{2+s}, u'_4, \ldots, u'_{N-2+s}, u'_N\right\}.$$

The proof relies on the ability to organize the unknowns, so that the block form can be easily recognized in the matrices. As we will see, this is similar to the block form that has been used for the proof of the order of convergence and of the stability analysis.

In the following we will use the notation $O_{m,n}$ to denote the null matrix of dimension $m \times n$. Using the notation $n = 2(i-1)$, the formulas in (9)-(10) for $i = 1, 2\ldots, m = N/2$ can be arranged in the block form

$$z_i^T = S_i(z_{i-1})^T + hQ_i(f_{i-1})^T + hR_i(f_i)^T$$

where

$$z_i = (u_{2(i-1)+r}, u_{2i-1}, u_{2(i-1)+s}, u_{2i}, u'_{2(i-1)+r}, u'_{2i-1}, u'_{2(i-1)+s}, u'_{2i}),$$

$$f_i = (f_{2(i-1)+r}, f_{2i-1}, f_{2(i-1)+s}, f_{2i}, g_{2(i-1)+r}, g_{2i-1}, g_{2(i-1)+s}, g_{2i}),$$

and

$$S_1 = \begin{pmatrix} \mathbf{e} & h\mathbf{c} \\ O_{4,1} & \mathbf{e} \end{pmatrix}$$

is of size 8×2 with $\mathbf{e}$ and $\mathbf{c}$ as in (9) and

$$S_i = \begin{pmatrix} O_{4,3} & \mathbf{e} & O_{4,3} & h\mathbf{c} \\ O_{4,3} & O_{4,1} & O_{4,3} & \mathbf{e} \end{pmatrix}$$

for $i = 2, 3, \ldots, m$, is of size 8×8, and the matrices Q_i and R_i are defined accordingly taking into account the coefficients in the formulas (9)-(10).

With this notation the linear system to be solved is of size $4N + 2$ where we can include the boundary conditions in the first two rows. Let us define the $(4N + 2)$-vectors

$$\mathcal{U} = (u_0, u'_0, z_1, z_2, \ldots, z_m)^T, \tag{16}$$

$$\mathcal{F} = \left(f_0, g_0, f_1, f_2, \ldots, f_m\right)^T. \tag{17}$$

Then, the system that provides the approximate values can be written as

$$D\mathcal{U} + hX\mathcal{F} + C = 0, \tag{18}$$

where the $(4N + 2) \times (4N + 2)$ matrices D and X are given by

$$D = \begin{pmatrix} B_1 & O_{2,8} & \ldots & O_{2,8} & B_m \\ -S_1 & I & & & \\ O_{8,2} & -S_2 & \ddots & & \\ & & \ddots & I & O_{8,8} \\ O_{8,2} & O_{8,8} & \ldots & -S_m & I \end{pmatrix}$$

with

$$B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

and

$$B_m = \begin{pmatrix} O_{1,3} & 0 & O_{1,4} \\ O_{1,3} & 1 & O_{1,4} \end{pmatrix};$$

$$X = \begin{pmatrix} O_{2,2} & O_{2,8} & \dots & O_{2,8} & O_{2,8} \\ W_1 & V_1 & & & \\ O_{8,2} & W_2 & \ddots & & \\ & & \ddots & V_{m-1} & O_{8\times 8} \\ O_{8,2} & O_{8\times 8} & \dots & W_m & V_m \end{pmatrix},$$

with

$$W_1 = \begin{pmatrix} \frac{\left(-1925+683\sqrt{3}\right)h}{11340} & \frac{\left(-106+35\sqrt{3}\right)h^2}{11340} \\ -\frac{1171h}{6720} & -\frac{67h^2}{6720} \\ -\frac{\left(1925+683\sqrt{3}\right)h}{11340} & -\frac{\left(106+35\sqrt{3}\right)h^2}{11340} \\ -\frac{37h}{105} & -\frac{2h^2}{105} \\ \frac{-797-44\sqrt{3}}{3780} & -\frac{\left(53+6\sqrt{3}\right)h}{3780} \\ -\frac{257}{1680} & -\frac{h}{210} \\ \frac{-797+44\sqrt{3}}{3780} & \frac{\left(-53+6\sqrt{3}\right)h}{3780} \\ -\frac{19}{105} & -\frac{h}{105} \end{pmatrix},$$

$$W_i = \begin{pmatrix} 0 & 0 & 0 & \frac{\left(-1925+683\sqrt{3}\right)h}{11340} & 0 & 0 & 0 & \frac{\left(-106+35\sqrt{3}\right)h^2}{11340} \\ 0 & 0 & 0 & -\frac{1171h}{6720} & 0 & 0 & 0 & -\frac{67h^2}{6720} \\ 0 & 0 & 0 & -\frac{\left(1925+683\sqrt{3}\right)h}{11340} & 0 & 0 & 0 & -\frac{\left(106+35\sqrt{3}\right)h^2}{11340} \\ 0 & 0 & 0 & -\frac{37h}{105} & 0 & 0 & 0 & -\frac{2h^2}{105} \\ 0 & 0 & 0 & \frac{-797-44\sqrt{3}}{3780} & 0 & 0 & 0 & -\frac{\left(53+6\sqrt{3}\right)h}{3780} \\ 0 & 0 & 0 & -\frac{257}{1680} & 0 & 0 & 0 & -\frac{h}{210} \\ 0 & 0 & 0 & \frac{-797+44\sqrt{3}}{3780} & 0 & 0 & 0 & \frac{\left(-53+6\sqrt{3}\right)h}{3780} \\ 0 & 0 & 0 & -\frac{19}{105} & 0 & 0 & 0 & -\frac{h}{105} \end{pmatrix},$$

for $i = 2, \dots, m$, and

$$V_i = \begin{pmatrix} -\frac{h}{36} & \frac{2\left(-241+144\sqrt{3}\right)h}{2835} & \frac{\left(-379+216\sqrt{3}\right)h}{1260} & \frac{\left(19+\sqrt{3}\right)h}{11340} & 0 & 0 & 0 & -\frac{\left(2+\sqrt{3}\right)h^2}{11340} \\ -\frac{3\left(105+64\sqrt{3}\right)h}{2240} & -\frac{h}{24} & \frac{3\left(64\sqrt{3}-105\right)h}{2240} & -\frac{19h}{6720} & 0 & 0 & 0 & \frac{h^2}{2240} \\ -\frac{\left(379+216\sqrt{3}\right)h}{1260} & -\frac{2\left(241+144\sqrt{3}\right)h}{2835} & -\frac{h}{36} & \frac{\left(19-\sqrt{3}\right)h}{11340} & 0 & 0 & 0 & \frac{\left(-2+\sqrt{3}\right)h^2}{11340} \\ -\frac{6\left(3+\sqrt{3}\right)h}{35} & -\frac{64h}{105} & \frac{6\left(-3+\sqrt{3}\right)h}{35} & -\frac{h}{105} & 0 & 0 & 0 & 0 \\ \frac{-81+8\sqrt{3}}{315} & \frac{8\left(-36+23\sqrt{3}\right)}{945} & \frac{-81+43\sqrt{3}}{315} & \frac{113-44\sqrt{3}}{3780} & 0 & 0 & 0 & \frac{\left(-17+6\sqrt{3}\right)h}{3780} \\ -\frac{9}{35}-\frac{3\sqrt{3}}{16} & -\frac{32}{105} & -\frac{9}{35}+\frac{3\sqrt{3}}{16} & -\frac{47}{1680} & 0 & 0 & 0 & \frac{h}{210} \\ \frac{-81-43\sqrt{3}}{315} & -\frac{8\left(36+23\sqrt{3}\right)}{945} & \frac{-81-8\sqrt{3}}{315} & \frac{113+44\sqrt{3}}{3780} & 0 & 0 & 0 & -\frac{\left(17+6\sqrt{3}\right)h}{3780} \\ -\frac{18}{35} & -\frac{64}{105} & -\frac{18}{35} & -\frac{19}{105} & 0 & 0 & 0 & \frac{h}{105} \end{pmatrix}$$

for $i = 1, 2, \dots, m$.

The $(4N+2)$-vector C in (18) contains the known values, that is,

$$C = (u_a, u_b, 0, \dots, 0)^T.$$

Now, let $u(x)$ be the true solution of the considered BVP, and define the $(4N+2)$-vector $\bar{\mathcal{U}}$ as follows

$$\bar{\mathcal{U}} = (u(x_0), u'(x_0), \bar{z}_1, \bar{z}_2, \dots, \bar{z}_m)^T$$

and the $(4N+2)$-vector $\bar{\mathcal{F}}$ by

$$\bar{\mathcal{F}} = \left(f(x_0, u(x_0), u'(x_0)), g(x_0, u(x_0), u'(x_0)), \bar{f}_1, \bar{f}_2, \dots, \bar{f}_m\right)^T,$$

where the $\bar{z}_i$ and $\bar{f}_i$ are the vectors z_i and f_i with the approximate values changed by the exact ones.

Using the vector-matrix notation, the exact representation of the global system may be expressed as

$$D\bar{\mathcal{U}} + hX\bar{\mathcal{F}} + C = \mathcal{L}(h)\,, \tag{19}$$

where the $(4N+2)$-vector $\mathcal{L}(h)$ contains the local truncation errors of the formulas, given by

$$\mathcal{L}(h) = \begin{pmatrix} 0 \\ 0 \\ \frac{-1}{1837080\sqrt{3}}\,u^{(9)}(x_0)h^9 + \mathcal{O}(h^{10}) \\ \frac{1}{14515200}\,u^{(10)}(x_0)h^{10} + \mathcal{O}(h^{11}) \\ \frac{1}{1837080\sqrt{3}}\,u^{(9)}(x_0)h^9 + \mathcal{O}(h^{10}) \\ \frac{-1}{589396500}\,u^{(11)}(x_0)h^{11} + \mathcal{O}(h^{12}) \\ \frac{-1}{612360}\,u^{(9)}(x_0)h^8 + \mathcal{O}(h^9) \\ \frac{1}{362880}\,u^{(9)}(x_0)h^8 + \mathcal{O}(h^9) \\ \frac{-1}{612310}\,u^{(9)}(x_0)h^8 + \mathcal{O}(h^9) \\ \frac{1}{589396500}\,u^{(12)}(x_0)h^{11} + \mathcal{O}(h^{12}) \\ \vdots \\ \frac{1}{589396500}\,u^{(12)}(x_{N-2})h^{11} + \mathcal{O}(h^{12}) \end{pmatrix}.$$

By subtracting (18) from (19) we get

$$D\mathcal{E} + hX(\bar{\mathcal{F}} - \mathcal{F}) = \mathcal{L}(h), \tag{20}$$

where $\mathcal{E} = \bar{\mathcal{U}} - \mathcal{U} = (e_0, e_0', e_r, e_1, e_s, e_2, e_r', e_1', e_s', e_2', e_{2+r}, \ldots, e_N')^T$ consists of the errors at the off-step and nodal points. Note that the exact boundary conditions are known, and thus, $e_0 = u(x_0) - u_0 = 0$ and $e_N = u(x_N) - u_N = 0$.

Using the Mean Value Theorem, one can consider for $i = 0, r, 1, s, 2, 2+r, 3, 2+s, 4, \ldots, N$, the identities

$$\begin{aligned} f(x_i, u(x_i), u'(x_i)) - f(x_i, u_i, u_i') &= (u(x_i) - u_i)\frac{\partial f}{\partial u}(\xi_i) + (u'(x_i) - u_i')\frac{\partial f}{\partial u'}(\xi_i) \\ &= e_i\frac{\partial f}{\partial u}(\xi_i) + e_i'\frac{\partial f}{\partial u'}(\xi_i) \\ g(x_i, u(x_i), u'(x_i)) - g(x_i, u_i, u_i') &= (u(x_i) - u_i)\frac{\partial g}{\partial u}(\eta_i) + (u'(x_i) - u_i')\frac{\partial g}{\partial u'}(\eta_i) \\ &= e_i\frac{\partial g}{\partial u}(\eta_i) + e_i'\frac{\partial g}{\partial u'}(\eta_i)\,. \end{aligned}$$

In the above expressions ξ_i and η_i stand for intermediate points on the line segment joining $(x_i, u(x_i), u'(x_i))$ to (x_i, u_i, u_i'). Now, using the formulas in (21) we have that

$$\bar{\mathcal{F}} - \mathcal{F} = \mathcal{J}\mathcal{E}, \tag{21}$$

where $\mathcal{J}$ is the $(4N+2) \times (4N+2)$-matrix containing the partial derivatives,

$$\mathcal{J} = \begin{pmatrix} J_0 & & & \\ & J_1 & & \\ & & \ddots & \\ & & & J_m \end{pmatrix},$$

with

$$J_0 = \begin{pmatrix} \frac{\partial f}{\partial u}(\xi_0) & \frac{\partial f}{\partial u'}(\xi_0) \\ \frac{\partial g}{\partial u}(\eta_0) & \frac{\partial g}{\partial u'}(\eta_0) \end{pmatrix},$$

and

$$J_i = \begin{pmatrix} \frac{\partial f}{\partial u}(\xi_{2(i-1)+r}) & \frac{\partial f}{\partial u'}(\xi_{2(i-1)+r}) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial f}{\partial u}(\xi_{2i-1}) & \frac{\partial f}{\partial u'}(\xi_{2i-1}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial f}{\partial u}(\xi_{2(i-1)s}) & \frac{\partial f}{\partial u'}(\xi_{2(i-1)+s}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial f}{\partial u}(\xi_{2i}) & \frac{\partial f}{\partial u'}(\xi_{2i}) \\ \frac{\partial g}{\partial u}(\eta_{2(i-1)+r}) & \frac{\partial g}{\partial u'}(\eta_{2(i-1)+r}) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial g}{\partial u}(\eta_{2i-1}) & \frac{\partial g}{\partial u'}(\eta_{2i-1}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial g}{\partial u}(\eta_{2(i-1)s}) & \frac{\partial g}{\partial u'}(\eta_{2(i-1)s}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial g}{\partial u}(\eta_{2i}) & \frac{\partial g}{\partial u'}(\eta_{2i}) \end{pmatrix}$$

for $i = 1, 2, \ldots, m$.

Finally, from Eqs. (20) and (21) we have that

$$D\mathcal{E} + hX\mathcal{J}\mathcal{E} = \mathcal{L}(h),$$

and letting $\mathcal{M} = D + hX\mathcal{J}$, it can be written as

$$\mathcal{M}\mathcal{E} = \mathcal{L}(h). \tag{22}$$

The matrix $\mathcal{M}$ has the same block structure as D and X, and can be written as

$$\mathcal{M} = \begin{pmatrix} B_1 & O_{2,8} & \dots & O_{2,8} & B_m \\ -S_1 + hU_1 & I + hT_1 & & & \\ O_{8,2} & -S_2 + hU_2 & \ddots & & \\ & & \ddots & I + hT_{m-1} & O_{8,8} \\ O_{8,2} & O_{8,8} & & -S_m + hU_m & I + hT_m \end{pmatrix},$$

where $U_1 = W_1 J_0$, $U_i = W_i J_{i-1}$ for $i = 2, 3, \dots, m$, and $T_i = V_i J_i$ for $i = 1, 2, \dots, m$.

The diagonal blocks of size 8×8 of $\mathcal{M}$ are non singular for h sufficiently small. Note that $\mathcal{M}$ is a correction of rank 1 of the matrix obtained solving the problem giving only the initial conditions, that is, when $B_1 = I$ and $B_m = 0$. This matrix is non singular because it is a block lower triangular matrix with non singular diagonal blocks. Using the Sherman-Morrison formula it is easy to prove that the matrix $\mathcal{M}$ is also non singular.

Then, the equation in (22) may be rewritten as

$$\mathcal{E} = \mathcal{M}^{-1}\mathcal{L}(h)\,. \tag{23}$$

We consider the maximum norm in $\mathbb{R}^{4N+2}$, $\|\mathcal{V}\| = \max_{1\le i\le N+2}\{|\mathcal{V}_i|\}$ and the corresponding induced matrix norm in $\mathbb{R}^{(4N+2)\times(4N+2)}$. Then, expanding each term of $\mathcal{M}^{-1}$ in powers of h, it can be shown that after some tedious manipulations we have $\|\mathcal{M}^{-1}\| = \mathcal{O}(h^{-1})$. The proof relies on the structure of matrix $\mathcal{M}$. For $N = 2$ the determinant of $\mathcal{M}$ is a polynomial on h of degree 10. When we add the two blocks of dimensions 8×8 to form the matrix for $N = 4$ the degree of the determinant increases up to 17, and so on. For $N = 2j$ with $j = 1, 2, \dots,$ the degree of the determinant of $\mathcal{M}$ is $3 + 7j = 3 + 7N/2$. To get the inverse $\mathcal{M}^{-1}$ we use the classical approach considering the cofactors and the determinant. The highest possible degree of cofactors is $2 + 7N/2$ (some of the coefficient may vanish and thus those degrees will be lower). It can be shown that the determinant takes the form $h(a_0 + a_1 h + \dots + a_{2+7N/2} h^{2+7N/2})$, and thus in the worst case we have that some terms of $\mathcal{M}^{-1}$ verify

$$\frac{b_0 + b_1 h + \dots + b_{2+7N/2} h^{2+7N/2}}{h(a_0 + a_1 h + \dots + a_{2+7N/2} h^{2+7N/2})} = O(h^{-1}).$$

Finally, from Eq. (23) and assuming that $u(x)$ has bounded derivatives up to the necessary order, we get

$$\|\mathcal{E}\| \le \|\mathcal{M}^{-1}\|\,\|\mathcal{L}(h)\| = |\mathcal{O}(h^{-1})|\,|\mathcal{O}(h^8)| \le Kh^7\,.$$

This completes the proof. □

Remark 4.1. We have obtained that the global method exhibits a seventh order convergence at all the points considered. Nevertheless, in view of the form of the vector $\mathcal{L}(h)$ we see that, assuming sufficient smoothness, at the mesh points we obtain a superconvergence order (see [8]):

- $|e_{2j+1}| = |u(x_{2j+1}) - u_{2j+1}| \le |\mathcal{O}(h^{-1})|\,|\mathcal{O}(h^{10})| \le Kh^9, \quad j = 0, 1, \dots N/2 - 1,$
- $|e_{2j}| = |u(x_{2j}) - u_{2j}| \le |\mathcal{O}(h^{-1})|\,|\mathcal{O}(h^{11})| \le Kh^{10} \quad j = 1, 2 \dots N/2.$

This interesting behaviour will be shown in the numerical examples, where we have included in the tables the approximate order of convergence at the nodal points considered.

4.1. Existence and uniqueness of the discrete solution

The following result establishes the existence and uniqueness of the solution provided by the system of equations in (18). Note that the convergence analysis is concerned with the behaviour of the errors for sufficiently small values of h. In fact, we have that $\|\mathcal{E}\| \to 0$ as $h \to 0$. In view of this, to facilitate the analysis, in the following results we will assume that $h < 1$.

Theorem 4.2. *Assuming that $f(x, \mathbf{u})$ verifies a Lipschitz condition on the variable $\mathbf{u} = (u, u')$, it holds that the system in (18) has a unique solution whenever $h < h_0$ with h_0 the unique positive solution of the equation $h\,\bar{d}\,L\left(4\,\frac{b-a}{h} + 2\right)^{1/2} = 1$, where $L = \max_{i=0,1}\{L_i\}$, $\bar{d} = \max_{\substack{i=1,\dots,4N+2\\ j=1,\dots,4N+2}}\{|\bar{D}_{ij}|\}$, being $\bar{D} = D^{-1}X\big|_{h=1}$.*

Proof. Let us consider the function $H : \mathbb{R}^{4N+2} \longrightarrow \mathbb{R}^{4N+2}$ given by

$$H(\theta) = \left(-D^{-1}C - hD^{-1}X\mathcal{F}(\theta)\right),$$

where $\theta = (\theta_1, \ldots, \theta_{4N+2})^T$, and $\mathcal{F}(\theta)$ denotes the vector $\mathcal{F}$ in (17) after doing the substitution $\mathcal{U} \to \theta$ in the corresponding terms f_j, g_j, where $\mathcal{U}$ is the vector in (16).

Note that for $\theta = \mathcal{U}$ the system in (18) adopts the form $\theta = H(\theta)$, so that the existence and uniqueness of the solution of the system (18) is equivalent to that of the equation $\theta = H(\theta)$.

We consider in $\mathbb{R}^{4N+2}$ the maximum norm $\|\theta\| = \max_{1\le i\le 4N+2}\{|\theta_i|\}$. We have that

$$|(H(\theta))_i - (H(\theta^*))_i| = \left|h\left[D^{-1}X(\mathcal{F}(\theta) - \mathcal{F}(\theta^*))\right]_i\right| \le h\,\bar{d}\sum_{j=1}^{4N+2} L\left|\theta_j - \theta_j^*\right|,$$

where $L = \max_{i=0,1}\{L_i\}$, and $\bar{d} = \max_{\substack{i=1,\ldots,4N+2\\ j=1,\ldots,4N+2}}\{|\bar{D}_{ij}|\}$, being $\bar{D}_{ij}$ the elements of matrix $\bar{D} = D^{-1}X\big|_{h=1}$. Note that we can assume without loss of generality that $h < 1$. The choice of $h = 1$ here is used only to get a bound of the elements $|(D^{-1}X)_{ij}|$ since in view of the matrices D and X, the elements of $D^{-1}X$ are either zero, or of the forms k, kh, kh^2, where k represents the corresponding constants in each case. Then, for $h = 1$ we obtain the maximum value of the terms $|(D^{-1}X)_{ij}|$, and thus we can easily obtain a bound of all the elements of $D^{-1}X$.

Taking into account the above inequalities and using the Cauchy-Schwartz inequality we can put

$$\begin{aligned}\|H(\theta) - H(\theta^*)\| &= \max_{1\le i\le 4N+2}\{|(H(\theta))_i - (H(\theta^*))_i|\}\\ &\le h\,\bar{d}\,L\,(4N+2)^{1/2}\,\|\theta - \theta^*\|\\ &= h\,\bar{d}\,L\left(4\,\frac{b-a}{h} + 2\right)^{1/2}\|\theta - \theta^*\| = \kappa\,\|\theta - \theta^*\|\end{aligned}$$

with $\kappa = h\,\bar{d}\,L\left(4\,\dfrac{b-a}{h} + 2\right)^{1/2}$.

Since for $h < h_0$ it is $\kappa < 1$, we will have that H is a contraction. Hence, by Banach's Fixed-Point Theorem the proof is complete. □

5. Implementation details

As we have already remarked, for solving a two-point BVP by using the new scheme, we have to solve a system of $4N+2$ linear or nonlinear equations in $4N+2$ unknowns according to the type of right hand side of (1). If the resulting system is linear, one can use any available linear system solver. On the other hand, for solving a nonlinear system, usually the Newton-Raphson's-type iterative procedures are used. In order to implement these procedures some good starting initial approximations are required. We discuss the following possible cases

(i) **Dirichlet-type BC:** In case of Dirichlet type BC, that is, $u_0 = u_a,\ u_N = u_b$, the given system is further reduced to $4N$ equations in $4N$ unknowns. The following values can be used to provide the initial starting values (as has been done in the numerical examples)

$$\begin{aligned} u_i^{(0)} &= u_0 + \frac{u_N - u_0}{b-a}\,ih, \quad i = r, 1, s, 2, \ldots, N-2+s\\ u_i'^{(0)} &= \frac{u_N - u_0}{b-a}, \quad i = 0, r, 1, s, 2, \ldots, N-2+s, N.\end{aligned}$$

(ii) **Neumann of Robin BC:** In this case, one can adopt the same strategy as given in [38]. That is, in this case, we consider a class of nonlinear boundary value problems called P_j, $j = 0(1)m$, such that for $j = 0$ initially we have the problem P_0 that has only the trivial solution $u(x) = 0$. If we consider, $j = m$, we recover the original problem. Thus, one has a class of boundary value problems given by

$$P_j = \begin{cases} u'' = f(x, u, u') - f(x, 0, 0) + \dfrac{j}{m} f(x, 0, 0)\\ g_1(u(a), u'(a)) = \dfrac{j}{m} v_a\\ g_2(u(b), u'(b)) = \dfrac{j}{m} v_b\\ \text{for} \quad j = 0(1)m.\end{cases}$$

Note that each of these problem P_j, $j = 1(1)m$, is solved by the hybrid block numerical scheme proposed in this article where the starting values are taken after solving the problem P_{j-1}. Finally, by letting $j = m$ we get a system that corresponds to the original BVP, that can be solved by taking the starting values those obtained after solving the problem P_{m-1}.

Table 2
MaxErr for Problem **P-1.**

N	$7BTM$	$8BVM$	$8BUM$
64	6.1923×10^{-25}	3.0550×10^{-13}	2.3980×10^{-14}
128	6.1309×10^{-28} ROC: 9.980	9.8590×10^{-14}	5.0630×10^{-14}
256	6.0295×10^{-31} ROC: 9.989	4.5960×10^{-13}	6.6720×10^{-14}

The above strategies could be applied in order to implement Newton-Raphson's type iterative procedures for solving the resulting nonlinear systems. If one has some other criterion for getting appropriate initial approximations then it can be used. Even sometimes, it is enough to take the initial approximations as zero (that may be accomplished by considering $m = 1$, and has been used in the numerical examples). More details on this strategy can be obtained from [38] and references therein.

If the system to be solved with Newton's method is denoted by $\mathbf{F}(\mathbf{Y}) = 0$, the iteration step is given by

$$\mathbf{Y}^{i+1} = \mathbf{Y}^i - \left(\mathbf{J}^i\right)^{-1}\mathbf{F}^i,$$

where $\mathbf{J}$ denotes the jacobian matrix of $\mathbf{F}$. In practice, it is not necessary to calculate the inverse of $\mathbf{J}^i$ at each step, since solving the linear system $\mathbf{J}^i\mathbf{Z}^i = -\mathbf{F}^i$ the solution is obtained as $\mathbf{Y}^{i+1} = \mathbf{Y}^i + \mathbf{Z}^i$. The stopping criteria adopted for the Newton's method are $|\mathbf{Y}^{i+1} - \mathbf{Y}^i| < 10^{-10}$ and $|\mathbf{F}(\mathbf{Y}^i)| < 10^{-10}$ imposing that the number of iterations does not exceed 50.

Note that the computational cost required to calculate the third derivatives is higher than the cost required for classical methods that use only the values "of f. The number of function evaluations needed by the method can be easily calculated using the following formula

$$(N+1) + (N-2+2) + (N/2+1) = 5N/2 + 2,$$

where N is the number of nodal points.

All the methods have been implemented using *Mathematica* 11.3 on a personal computer with configuration i7-7500U, 1.80 GHz using double precision arithmetic in the numerical computations. When the errors were near the machine precision, to get errors with a higher precision, we used in the *Mathematica* code the option `WorkingPrecision->32` (this option specifies how many digits of precision should be maintained in internal computations.).

6. Numerical experiments

In this section, we have solved some test problems by using the new scheme, named as $7BTM$, and results are compared with some existing higher order schemes in the scientific literature. The methods considered for comparisons are of higher global orders than the proposed numerical scheme $7BTM$. We have not considered other standard methods applied to the associated first order systems because we have considered only the methods with the best performance in the cited articles.

In the following tables, the notation N stands for total number of nodal points and MaxErr is designated for the maximum absolute error along all the nodal points. We have included the estimation of the numerical order of convergence with the proposed method in the tables presented. This estimation has been obtained with the usual formula

$$ROC \simeq \log_2\left(\frac{MAE_{2h}}{MAE_h}\right),$$

where MAE_h denotes the maximum absolute error on the grid points of the integration interval taking step size h.

6.1. Comparison with some eighth-order methods by Biala [50]

Firstly, we shall compare the performance of the new scheme $7BTM$ with the eighth-order boundary value method $8BVM$ and eighth-order block unification method $8BUM$ given in [50]. For that purpose, we consider the following two problems.

A nonlinear BVP

As a first problem, let the nonlinear BVP with Robin-type BCs discussed in the scientific literature [50,51]

$$\textbf{P-1.}\begin{cases} u''(x) = \dfrac{(u'(x))^2 + u^2(x)}{2e^x}, \quad 0 \le x \le 1, \\ u(0) - u'(0) = 0, u(1) + u'(1) = 2e \\ \text{True solution: } u(x) = e^x. \end{cases}$$

This problem has been solved for different number of nodal points. Table 2 reveals the good performance of the proposed scheme $7BTM$.

A nonlinear system of BVPs

Table 3
MaxErr for Problem **P-2.**

N	$7BTM$	$8BVM$	$8BUM$
12	2.2676×10^{-16}	8.3900×10^{-13}	3.5630×10^{-12}
24	2.7160×10^{-19}	1.2420×10^{-13}	1.4600×10^{-14}
	ROC: 9.705		
48	2.8265×10^{-22}	3.9990×10^{-11}	1.6290×10^{-14}
	ROC: 9.908		

Table 4
MaxErr for Problem **P-3.** and **P-4.**

Problem→	P-3.		P-4.	
h	$7BTM$	$8HOM$	$7BTM$	$8HOM$
$\frac{1}{2}$	1.0653×10^{-8}	6.770×10^{-8}	5.4979×10^{-11}	3.030×10^{-10}
$\frac{1}{4}$	3.2933×10^{-11}	3.580×10^{-10}	9.3038×10^{-14}	1.200×10^{-12}
	ROC: 8.337		ROC: 9.206	
$\frac{1}{8}$	5.8488×10^{-14}	1.530×10^{-12}	1.1035×10^{-16}	5.340×10^{-15}
	ROC: 9.137		ROC: 9.719	
$\frac{1}{16}$	7.7367×10^{-17}	5.950×10^{-15}	1.1681×10^{-19}	3.790×10^{-15}
	ROC: 9.562		ROC: 9.883	

As a next test problem, let us consider the system of BVPs of nonlinear type [50]

$$\textbf{P-2.} \qquad \begin{cases} u''(x) + 20u'(x) + 4\cos(x)u(x) + \sin(u(x), v(x)) & = f_1(x), \\ v''(x) + 5e^x v'(x) + 6\sinh(x)v(x) + \cos(v(x)) & = f_2(x), \end{cases}$$

where $0 \le x \le 1$ and

$$\begin{cases} f_1(x) & = 21e^x + 4e^x\cos(x) + \sin(e^x\sinh(x)), \\ f_2(x) & = \cos(\sinh(x)) + 5e^x\cosh(x) + \sinh(x) + 6\sinh^2(x). \end{cases}$$

The true solution of the system is $u(x) = e^x,\ v(x) = \sinh(x)$. The problem is solved subject to the boundary conditions

$$\begin{cases} u(0) = 1,\ u(1) = e, \\ v(0) = 0,\ v(1) = \sinh(1), \end{cases}$$

as in [50].

This nonlinear system is solved for different number of nodal points. One can observe from Table 3, the good performance of the new scheme.

6.2. Comparison with an eighth-order method by Usmani [16]

$$\textbf{P-3.} \begin{cases} x^2u''(x) = 2u(x) - x,\ 2 \le x \le 3, \\ u(2) = \dfrac{10}{19},\ u(3) = \dfrac{45}{38}, \\ \text{True solution: } u(x) = \dfrac{19x - 36x^{-1}}{38} \end{cases}$$

$$\textbf{P-4.} \qquad \begin{cases} u''(x) = u(x) + x^2 - 2,\ 0 \le x \le 1 \\ u(0) = 0,\ u(1) = 1, \\ \text{True solution: } u(x) = \dfrac{e^2x^2 - x^2 + 2e^{1-x} - 2e^{x+1}}{1 - e^2} \end{cases}$$

Both of the above problems have been solved for different step-sizes in order to illustrate performance of the new scheme in comparison with the high order numerical scheme given in [16]. The numerical results in Table 4 demonstrate a good performance of the scheme $7BTM$.

6.3. Comparison with an eighth-order tri-diagonal finite difference method by Chawla [13]

Now we compare the performance of our proposed scheme with an 8th order tri-diagonal finite difference scheme in [13], named as $8TFD$. For that purpose, we consider the following BVPs that were used in [13] in order to illustrate the

Table 5
MaxErr for Problem **P-5.** and **P-6.**

Problem→ N	P-5. *7BTM*	*8TFD*	P-6. *7BTM*	*8TFD*
4	3.0371×10^{-9}	5.5000×10^{-9}	2.5258×10^{-8}	2.9000×10^{-8}
8	7.9762×10^{-12}	2.6000×10^{-11}	7.2060×10^{-11}	1.4000×10^{-10}
	ROC: 8.572		8.453	
16	1.3170×10^{-14}	1.1000×10^{-13}	1.2483×10^{-13}	5.9000×10^{-13}
	ROC: 9.242		9.173	

Table 6
MaxErr for Problem **P-7.**

N	ϵ	*7BTM*	COLSYS
68	10^{-2}	9.8×10^{-11}	2.2×10^{-9}
140	10^{-4}	7.1×10^{-5}	1.6×10^{-8}

Table 7
Data for Problem **P-7.**

N	MaxErr	ROC
For $\epsilon = 10^{-4}$		
512	1.2749×10^{-9}	
1024	1.5709×10^{-12}	9.664
For $\epsilon = 10^{-5}$		
512	3.6430×10^{-5}	
1024	9.1995×10^{-8}	8.629

performance of the schemes

$$\textbf{P-5.}\begin{cases} u''(x) = \dfrac{(2-x)e^{2u(x)} + \frac{1}{1+x}}{3}, \ 0 \le x \le 1, \\ u(0) = 1, \ u(1) = -\log(2), \\ \text{True solution: } u(x) = \log\left(\dfrac{1}{1+x}\right) \end{cases}$$

and

$$\textbf{P-6.} \quad \begin{cases} u''(x) = \dfrac{(1-x)u(x)+1}{(1+x)^2}, \ 0 \le x \le 1, \\ u(0) = 1, \ u(1) = 0.5, \\ \text{True solution: } u(x) = \dfrac{1}{1+x}. \end{cases}$$

Both of the above problems have been solved for different values of $N = 4, 8, 16$. The problems have also been solved by an 8th-order tri-diagonal finite difference method given in [13]. The numerical results provided in Table 5 indicate a good performance of the 7*BTM*.

6.4. Comparison with code COLSYS given in [8]

A stiff singularly perturbed BVP

Consider the following BVP with Dirichlet-type BCs discussed in [4,8]

$$\textbf{P-7.} \quad \begin{cases} \epsilon u''(x) + xu'(x) = -\epsilon\pi^2\cos(\pi x) - (\pi x)\sin(\pi x), -1 \le x \le 1, \\ u(-1) = -2, u(1) = 0\,, \\ \text{True solution: } u(x) = \cos(\pi x) + \text{erf}(x/\sqrt{2\epsilon})/\text{erf}(1/\sqrt{2\epsilon}). \end{cases}$$

This problem has been solved for different values of $\epsilon = 10^{-2}$ and $\epsilon = 10^{-4}$. The numerical results have been presented in Table 6 by applying the scheme 7*BTM* and the code COLSYS given in [8]. Note that COLSYS uses a mesh variation strategy whereas the scheme 7*BTM* uses constant mesh size. For the numerical data given in Table 6, the code COLSYS uses tolerance 10^{-6} for $\epsilon = 10^{-2}$ and $\epsilon = 10^{-4}$. It can be observed from the Table 6 that the scheme 7*BTM* performs better in terms of accuracy for $\epsilon = 10^{-2}$ whereas the code COLSYS performs better in terms of accuracy for $\epsilon = 10^{-4}$. This shows that the scheme 7*BTM* should be enhanced with a mesh variation strategy in order to make the proposed scheme competitive with the COLSYS. In Fig. 1, a plot of errors is given whereas Fig. 2 is concerned with the exact and numerical solution of the problem for $\epsilon = 10^{-2}$ and $N = 68$. Further, in Table 7, we have presented the data by applying the proposed code 7*BTM* for small values of ϵ, that is, $\epsilon = 10^{-4}$ and $\epsilon = 10^{-5}$ with very small mesh sizes, that is, for $N = 512$ and $N = 1024$.This indicates

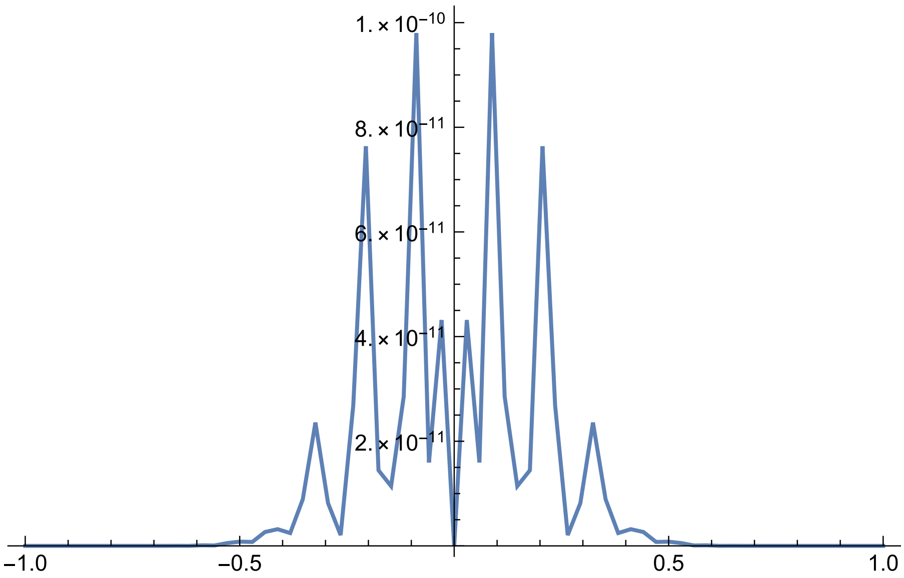

Fig. 1. Plot of errors for Problem **P-7.**

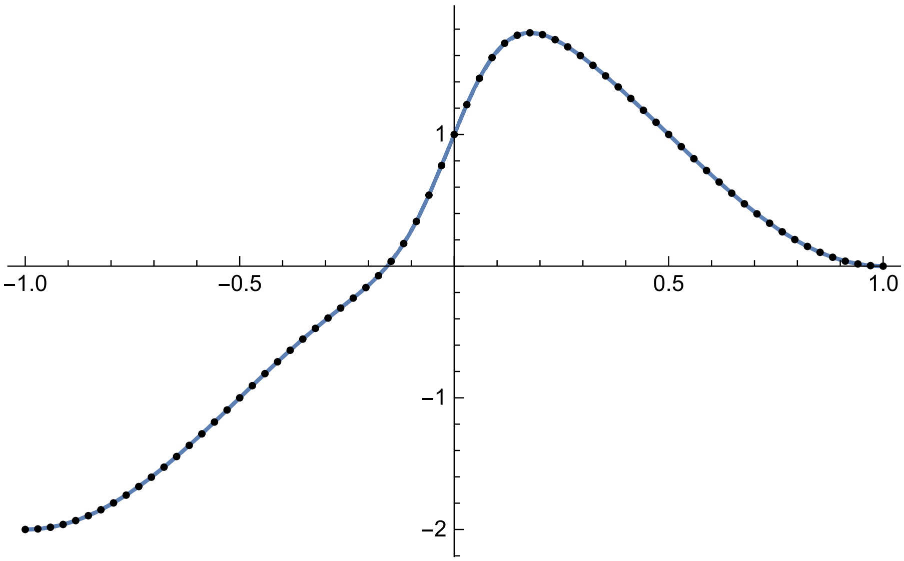

Fig. 2. Exact and numerical solution for **P-7.**

that the method also produces acceptable results even for small values of the perturbation parameters and in this case only need to consider more grid points.

6.5. Comparison with code TWPBVPC given in [5]

A singularly perturbed BVP

Consider the following BVP with Dirichlet-type BCs discussed in [2,5]

$$\textbf{P-8.}\begin{cases}\epsilon u''(x) - xu'(x) - u(x) = -(1+\epsilon\pi^2)\cos(\pi x) + (\pi x)\sin(\pi x), -1 \le x \le 1,\\ u(-1) = -1, u(1) = -1\,,\\ \text{True solution: } u(x) = \cos(\pi x).\end{cases}$$

This problem has been solved for $\epsilon = 10^{-2}$ taking $N = 34, 62$. The numerical results presented in Table 8 have been obtained by applying the scheme $7BTM$ and the code TWPBVPC given in [5]. Note that the code TWPBVPC requires a tolerance (*tol*) which was taken as *tol*=10^{-4} for $N = 34$ and *tol*=10^{-6} for $N = 62$. The data given in Table 8 reveal the good performance of

Table 8
Mixed relative error for Problem **P-8.**

N	$7BTM$	TWPBVPC
34	8.993×10^{-14}	2.09×10^{-6}
62	3.130×10^{-16}	4.12×10^{-9}

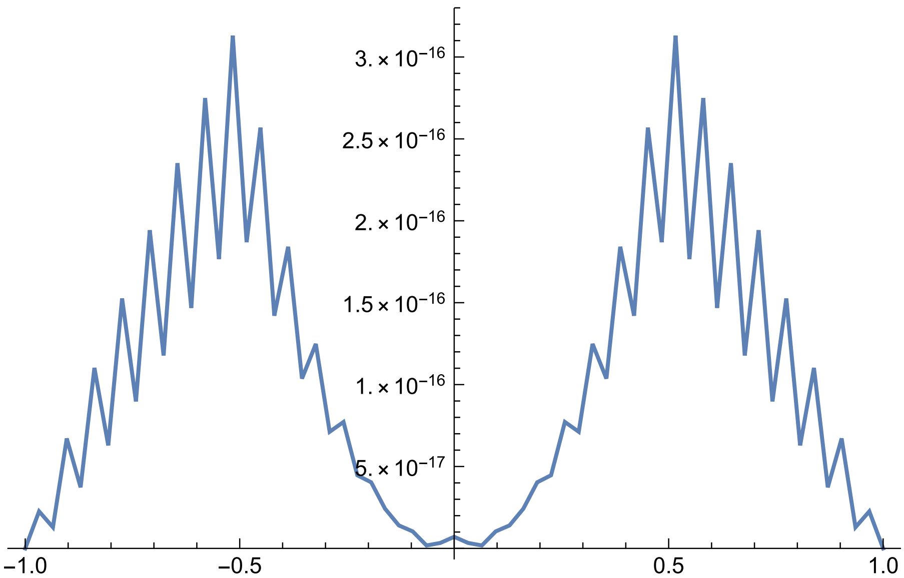

Fig. 3. Plot of errors for Problem **P-8.**

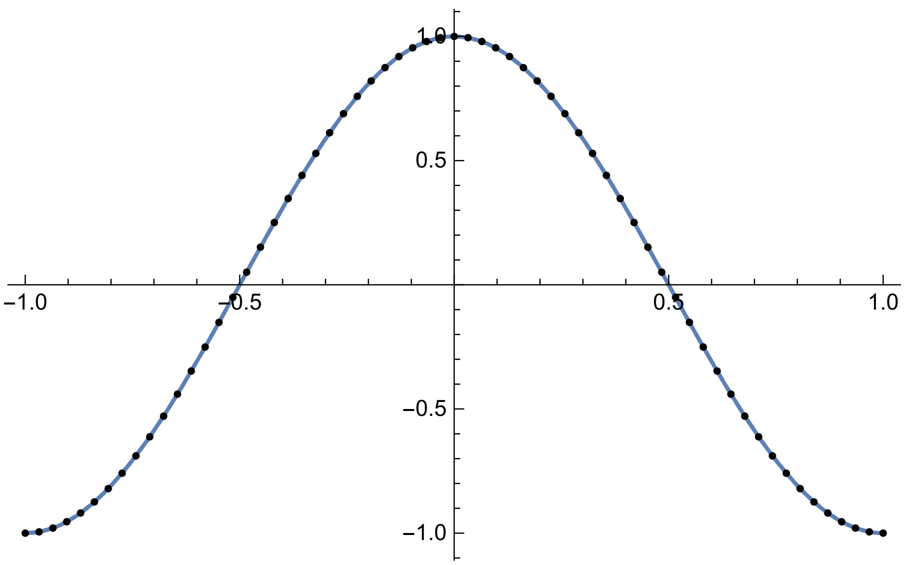

Fig. 4. Exact and numerical solution for Problem **P-8.**

the scheme $7BTM$. In this table we computed the errors using the formula

$$\max_{0\le i\le N}\left\{\frac{|u(x_i)-u_i|}{1+|u(x_i)|}\right\}.$$

In Fig. 3, a plot of mixed relative errors is given whereas Fig. 4 is concerned with the exact and numerical solutions of the problem for $\epsilon = 10^{-2}$ and $N = 62$. Further, we have presented the data by applying the proposed code for small values of the perturbation parameter, that is, $\epsilon = 10^{-5}$ and $\epsilon = 10^{-6}$. The numerical data given in Table 9 show the good performance of the proposed code.

Table 9
Data for Problem **P-8.**

N	MaxErr	ROC
For $\epsilon = 10^{-5}$		
64	5.5670×10^{-12}	
128	6.7690×10^{-16}	13.005
For $\epsilon = 10^{-6}$		
64	5.5909×10^{-9}	
128	6.8277×10^{-13}	12.999

7. Conclusions

In this article, we have developed an optimized version of a third derivative hybrid block method for solving general second order two-point boundary value problems numerically. A constructive approach has been applied in the development of the method in order to justify the optimal values of the two-off step points of the method. A theoretical analysis of the new scheme has been carried out, proving its seventh order global convergence. Some test problems are solved in order to show the good performance of the new scheme in comparison with some higher order methods existing in the scientific literature.

Acknowledgments

The authors would like to thank the anonymous reviewers for their work and constructive comments that greatly contributed to improve the manuscript.

References

[1] E.A. Coddington, N. Levinson, Theory of ordinary differential equations, International Series in Pure and Applied Mathematics, McGraw Hill, New York, 1955.
[2] K. Soetaert, J. Cash, F. Mazzia, Solving Differential Equations in R, Springer, 2012.
[3] H.B. Keller, Numerical Methods for Two-Point Boundary Value Problems, Ginn-Blaisdell, Waltham, Massachusetts, 1968.
[4] U.M. Ascher, R.M.M. Mattheij, R.D. Russel, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, Prentice Hall, Englewood Cliffs, New York, 1988.
[5] Test set for BVP solvers, 2020. https://archimede.dm.uniba.it/~bvpsolvers/testsetbvpsolvers/.
[6] S.N. Ha, A nonlinear shooting method for two-point boundary value problems, Comput. Math. Applics. 42 (2001) 1411–1420.
[7] P. Amodio, I. Segura, High-order finite difference schemes for the solution of second-order BVPs, J. Comput. Appl. Math. 176 (2005) 59–76.
[8] U. Ascher, J. Christiansen, R.D. Russell, A collocation solver for mixed order systems of boundary value problems, Math. Comp. 33 (146) (1979) 659–679.
[9] U. Ascher, J. Christiansen, R.D. Russell, Collocation software for boundary-value ODEs, ACM Trans. Math. Software 7 (2) (1981) 209–222.
[10] J.R. Cash, D. Hollevoet, F. Mazzia, A.M. Nagy, The MATLAB code bvptwp.m for the numerical solution of two point boundary value problems, ACM Trans. Math. Softw. 39 (2) (2013) 15.1–15.12.
[11] P. Amodio, G. Settanni, A finite differences MATLAB code for the numerical solution of second order singular perturbation problems, J. Comp. Appl. Math. 236 (16) (2012) 3869–3879.
[12] H.B. Thompson, 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.
[13] M.M. Chawla, An eighth order tridiagonal finite difference method for nonlinear two-point boundary value problems, BIT 17 (1977) 281–285.
[14] M.M. Chawla, High-accuracy tri-diagonal finite difference approximations for non-linear two-point boundary value problems, J. Inst. Maths. Appl. 22 (1978) 203–209.
[15] M.M. Chawla, A sixth-order tri-diagonal finite difference method for non-linear two-point boundary value problems, BIT 17 (2) (1977) 128–133.
[16] R.A. Usmani, A method of high order accuracy for the numerical integration of boundary value problems, BIT 13 (1973) 458–469.
[17] F.-G. Lang, X.P. Xu, Quintic b-spline collocation method for second order mixed boundary value problem, Comp. Phy. Comm. 183 (2012) 913–921.
[18] N. Caglar, H. Caglar, B-spline method for solving linear system of second-order boundary value problems, Comput. Math. Appl. 57 (2009) 757–762.
[19] M. Dehghan, A. Nikpour, Numerical solution of the system of second-order boundary value problems using the local radial basis functions based differential quadrature collocation method, Appl. Math. Model. 37 (2013) 8578–8599.
[20] S. Ul-Islam, I. Aziz, B. Sarler, The numerical solution of second-order boundary-value problems by collocation method with the haar wavelets, Math. Comp. Model. 52 (2010) 1577–1590.
[21] M.K. Jain, T. Aziz, Cubic spline solution of two-point boundary value problems with significant first derivatives, Comput. Methods Appl. Mech. Eng. 39 (1983) 83–91.
[22] M. Lakestani, M. Dehghan, The solution of a second-order nonlinear differential equation with Neumann boundary conditions using semi-orthogonal b-spline wavelets, Int. J. Comput. Math. 83 (2006) 685–694.
[23] J. Lu, Variational iteration method for solving a nonlinear system of second-order boundary value problems, Comput. Math. Appl. 54 (2007) 1133–1138.
[24] S.N. Jator, Z. Sinkala, A high order b-spline collocation method for linear boundary value problems, Appl. Math. Comp. 191 (2007) 100–116.
[25] L.F. Shampine, I. Gladwell, S. Thompson, Solving ODEs with MATLAB, Cambridge, 2003.
[26] F. Mazzia, J.R. Cash, K. Soetaert, Solving boundary value problems in the open source software R: package bvpSolve, Opuscula Math. 34 (2) (2014) 387–403.
[27] F. Mazzia, J.R. Cash, A fortran test set for boundary value problem solvers, in: AIP Conference Proceedings, 2015, p. 1648.
[28] J.R. Cash, F. Mazzia, Efficient global methods for the numerical solution of nonlinear systems of two point boundary value problems, Recent Adv. Comput. Appl.Math. (2011) 23–39.
[29] J.R. Cash, D. Hollevoet, F. Mazzia, A.M. Nagy, Algorithm 927: The MATLAB code bvptwp.m for the numerical solution of two point boundary value problems, ACM Trans. Math. Softw. 39 (2) (2013).
[30] L.F. Shampine, P.H. Muir, H. Xu, A user-friendly fortran BVP solver, JNAIAM J. Numer. Anal. Ind. Appl. Math. 1 (2) (2006) 201–217.
[31] C. Manni, F. Mazzia, A. Sestini, H. Speleers, BS2 methods for semi-linear second order boundary value problems, Appl. Math. Comp. 255 (2015) 147–156.

[32] F. Mazzia, A. Sestini, D. Trigiante, The continuous extension of the b-spline linear multistep methods for BVPs on non-uniform meshes, Appl. Numer. Math. 59 (3–4) (2009) 723–738.
[33] L. Brugnano, D. Trigiante, Solving Differential Problems by Multi-Step Initial and Boundary Value Methods, Gordan and Breach Science Publishers, 1998.
[34] L. Brugnano, D. Trigiante, Boundary value methods: the third way between linear multi-step and Runge-Kutta methods, Comput. Math. Appl. 36 (1998) 269–284.
[35] W.E. Milne, Numerical Solution of Differential Equations, John Wiley & Sons, 1953.
[36] L.F. Shampine, H.A. Watts, Block implicit one-step methods, Math. Comp. 23 (1969) 731–740.
[37] S.O. Fatunla, Block methods for second order odes, Int. J. Comput. Math. 41 (1991) 55–63.
[38] H. Ramos, M. Rufai, A third derivative two-step block Falkner-type method for solving general second order boundary-value problems, Math. Comput. Simul. 165 (2019) 139–155, doi:10.1016/j.matcom.2019.03.003.
[39] H. Ramos, G. Singh, A tenth-order a-stable two-step hybrid block method for solving initial value problems of ODEs, Appl. Math. Comp. 310 (2017) 75–88.
[40] H. Ramos, G. Singh, A note on variable step-size formulation of a Simpson's-type second derivative block method for solving stiff systems, Appl. Math. Lett. 64 (2017) 101–107.
[41] H. Ramos, S. Mehta, J. Vigo-Aguiar, A unified approach for the development of k-step block Falkner-type methods for solving general second-order initial-value problems in ODEs, J. Comp. Appl. Math. 318 (2017) 550–564, doi:10.1016/j.cam.2015.12.018.
[42] H. Ramos, Z. Kalogiratou, T. Monovasilis, T.E. Simos, An optimized two-step hybrid block method for solving general second order initial-value problems, Numer. Algor. 72 (4) (2016) 1089–1102, doi:10.1007/s11075-015-0081-8.
[43] H. Ramos, M. Rufai, Third derivative modification of k-step block Falkner methods for the numerical solution of second order initial-value problems, Appl. Math. Comp. 333 (2018) 231–245.
[44] H. Ramos, M. Rufai, Numerical solution of boundary value problems by using an optimized two-step block method, Numer. Algor. 84 (1) (2020) 229–251.
[45] L. Aceto, D. Trigiante, Symmetric schemes, time reversal symmetry and conservative methods for hamiltonian systems, J. Comput. Appl. Math. 107 (1999) 257–274.
[46] P. Amodio, F. Iavernaro, D. Trigiante, Conservation properties of symmetric BVMs applied to linear Hamiltonian problems, in: P.M.A. Sloot, A.G. Hoekstra, C.J.K. Tan, J.J. Dongarra (Eds.), Computational Science - ICCS 2002. ICCS 2002. Lecture Notes in Computer Science, vol. 2331, Springer, Berlin, Heidelberg, 2002.
[47] E.H.C. Lubich, G. Wanner, Geometric numerical integration structure-preserving algorithms for ordinary differential equations, Springer Series in Computational Mathematics, vol. 31, 2002.
[48] L. Brugnano, D. Trigiante, High-order multi-step methods for boundary value problems, Appl. Numer. Math. 18 (1995) 79–94.
[49] P.P. See, Z.A. Majid, M. Suleiman, Three-step block method for solving nonlinear boundary value problems, Abst. Appl. Anal. (2014), doi:10.1155/2014/379829.
[50] T.A. Biala, A computational study of the boundary value methods and the block unification methods for $y'' = f(x, y, y')$,, Abst. Appl. Anal. (2016) 1–15, doi:10.1155/2016/8465103.
[51] T.A. Biala, S.N. Jator, A family of boundary value methods for systems of second-order boundary value problems, Int. J. Diff. Eqs. (2017) 1–12, doi:10.1155/2017/2464759.
[52] M.I. Modebei, S.N. Jator, H. Ramos, Block hybrid method for the numerical solution of fourth order boundary value problems, J. Comp. Appl. Math. 337 (2020) 112876, doi:10.1016/j.cam.2020.112876.
[53] H. Ramos, M.A. Rufai, A two-step hybrid block method with fourth derivatives for solving third-order boundary value problems, J. Comp. Appl. Math. 404 (2022) 113419, doi:10.1016/j.cam.2021.113419.
[54] H. Ramos, A.L. Momoh, Development and implementation of a tenth-order hybrid block method for solving fifth-order boundary value problems, Math. Model. Anal. 26 (2021) 267–286, doi:10.3846/mma.2021.12940.