Iranian Journal of Numerical Analysis and Optimization

Volume 11 , Number 1

Winter 2021

Serial Number: 19

Ferdowsi University of Mashhad, Iran

In the Name of God

Iranian Journal of Numerical Analysis and Optimization (IJNAO)

This journal is authorized under the registration No. 174/853 dated 1386/2/26 (2007/05/16), by the Ministry of Culture and Islamic Guidance.

Volume 11, Number 1, Winter 2021

ISSN-Print: 2423-6977, **ISSN-Online:** 2423-6969

Publisher: Faculty of Mathematical Sciences, Ferdowsi University of Mashhad

Published by: Ferdowsi University of Mashhad Press

Printing Method: Electronic

Address: Iranian Journal of Numerical Analysis and Optimization

Faculty of Mathematical Sciences, Ferdowsi University of Mashhad

P.O. Box 1159, Mashhad 91775, Iran.

Tel. : +98-51-38806222 , **Fax:** +98-51-38807358

E-mail: ijnao@um.ac.ir

Website: http://ijnao.um.ac.ir

This journal is indexed by:

- Scopus database of Elsevier (from 31/07/2020)
- Zentralblatt
- ISC
- SID
- Civilica
- Magiran
- DOAJ
- OAJI
- CiteFactor
- AcademicKeys

• The Journal granted the Scientific-Research degree by the Iranian Ministry of Science, Research, and Technology.

Iranian Journal of Numerical Analysis and Optimization

Volume 11, Number 1, Winter 2021

Ferdowsi University of Mashhad - Iran

Iranian Journal of Numerical Analysis and Optimization

This journal is published under the auspices of Ferdowsi University of Mashhad

We would like to acknowledge the help of Miss Narjes khatoon Zohorian in the preparation of this issue.

Letter from the Editor–in–Chief

I would like to welcome you to the Iranian Journal of Numerical Analysis and Optimization (IJNAO). This journal is published two issues per year and supported by the Faculty of Mathematical Sciences at the Ferdowsi University of Mashhad. Faculty of Mathematical Sciences with three centers of excellence and three research centers is well-known in mathematical communities in Iran.

The main aim of the journal is to facilitate discussions and collaborations between specialists in applied mathematics, especially in the fields of numerical analysis and optimization, in the region and worldwide.

Our vision is that scholars from different applied mathematical research disciplines, pool their insight, knowledge and efforts by communicating via this international journal.

In order to assure high quality of the journal, each article is reviewed by subject-qualified referees.

Our expectations for IJNAO are as high as any well-known applied mathematical journal in the world. We trust that by publishing quality research and creative work, the possibility of more collaborations between researchers would be provided. We invite all applied mathematicians especially in the fields of numerical analysis and optimization to join us by submitting their original work to the Iranian Journal of Numerical Analysis and Optimization.

Ali R. Soheili

Contents

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 1–31
DOI:10.22067/ijnao.2021.11304.0

Research Article

Solving fuzzy multiobjective linear bilevel programming problems based on the extension principle

A. Abbasi Molai

Abstract

Fuzzy multiobjective linear bilevel programming (FMOLBP) problems are studied in this paper. The existing methods replace one or some deterministic model(s) instead of the problem and solve the model(s). Doing this work, we lose much information about the compromise decision, and it does not make sense for the uncertain conditions. To overcome the difficulties, Zadeh's extension principle is applied to solve the FMOLBP problems. Two crisp multiobjective linear three-level programming problems are proposed to find the lower and upper bound of its objective values in different levels. The problems are reduced to some linear optimization problems using one of the scalarization approaches, called the weighting method, the dual theory, and the vertex enumeration method. The lower and upper bounds are estimated by the resolution of the corresponding linear optimization problems. Hence, the membership functions of compromise objective values are produced, which is the main contribution of this paper. This technique is applied for the problem for the first time. This method applies all information of a fuzzy number and does not estimate it by a crisp number. Hence, the compromise decision resulted from the proposed method is consistent with reality. This point can minimize the gap between theory and practice. The results are compared with the results of existing approaches. It shows the efficiency of the proposed approach.

AMS subject classifications (2020): 90C70, 90C29, 90C46.

Keywords: Fuzzy mathematical programming; Fuzzy number; Fuzzy multiobjective bilevel programming; Extension principle; Vertex points; Weighting method.

Received 7 January 2020; revised 16 April 2020; accepted 3 September 2020
Ali Abbasi Molai
School of Mathematics and Computer Sciences, Damghan University, Damghan, P.O.Box 36715-364, Iran. e-mail: a_abbasimolai@du.ac.ir

1 Introduction

A bilevel multiobjective programming problem (BL-MOPP) is applied to model decentralized decision problems that contain the objectives at the upper level, called leader, and the objectives at the lower level, called the follower. The decisions are made in a hierarchical order from the upper level to the lower level. The decision-makers in each level try to optimize their objective functions. However, the decision in a level may be affected by the objective values in another level. The objectives in both levels may have confliction objectives, which should be optimized simultaneously. The coefficients in the objective functions and constraints are determined by experts; see [3, 46]. In the most real-world situations, the possible values of the coefficients are usually estimated imprecisely or vaguely by experts. These values cannot be presented in terms of crisp values because we miss much information. The most appropriate form to express the values is fuzzy sets.

Many researchers have been studied bilevel programming problems since it was introduced by Von Stackelberg [36]. A class of algorithms and approaches was well designed to solve the problem like the Kuhn–Tucker method [7, 8, 9, 16], the vertex enumeration method [9], the K-best approach [9, 2], and the branch and bound procedure [14]. The BL-MOPPs were studied to formulate the problems of the real-world, for example, the location planning problem for stone industrial parks; see [5]. Their coefficients are expressed by crisp numbers. On the other hand, the coefficients cannot usually be expressed in terms of a real number in the uncertain conditions. Also, multilevel programming problems (MLPPs) in [15, 35, 34] were discussed for hierarchical decentralized planning problems. An overview of bilevel programming problem was provided by Colson, Marcotte, and Savard [13]. The fuzzy approach for MLPP in [34] was extended to solve bilevel and three-level nonlinear multiobjective programming problems by Abo-Sinna and Baky [2]. Interactive fuzzy programming was developed to solve fuzzy multilevel linear programming problem in [29]. The balance space method was proposed for the nonlinear multiobjective bilevel programming problem in [2]. The fuzzy goal programming (FGP) algorithm was used to solve BL-MOPP in [6]. An algorithm was designed based on FGP to solve MLPP in [9]. An FGP model was proposed to solve BL-MOPP by Pramanik and Dey [4]. This model does not follow the hierarchical structure of bilevel programming and does not consider the upper level decisions, and it is as a single-level multiobjective programming problem. In bilevel programming problems, the decisions of leader should be dominant over that of followers. Mohamed [5] well developed an efficient method to solve the multiobjective programming problems using FGP approach. Moitra and Pal [15] then extended the concept of FGP approach to finding a satisfactory solution for bilevel programming problems. Moreover, FGP approach was extended to solve a deterministic decentralized BL-MOPP in [6] and MLPP in [4]. Due to the existence of uncertainty in real-world situations, some or all parameters in objective functions and/or in constraints of the leader and/or the follower can be described by fuzzy numbers. Budnitzkia [5] designed a solution procedure to solve the linear fuzzy bilevel programming using α-cuts of the fuzzy polytopes. Zhang and his colleagues [17, 35, 37] utilized the extended solution concept and theorems of bilevel programming and presented procedures to solve the fuzzy linear bilevel programming (FLBP) in special cases of membership functions such as the triangular and the general form [42, 43, 44, 45]. When the leader, the follower, or both of them have multiple objectives, the fuzzy set technique was applied to investigate FLBP problems by Zhang, Lu, and Dillon [46]. They also extended their previous research to the fuzzy multiobjective linear bilevel programming (FMOLBP) problem and developed an approximation branch-and-bound algorithm to solve the FMOLBP problem; see [46]. A new definition proposed in [32, 35] implies that the upper level constraints containing the second variable move to the lower level. This point changes the nature of the problem; see [1, 8]. On the other hand, the method given in [46] solves several crisp multiobjective nonlinear programming problems along with many crisp objectives and constraints by the approximation branch and bound method until the termination criterion holds. Hence, it will have high computational complexity and we have to check the complementary slackness conditions in each of the iterations, which is time-consuming.

Toksari and Bilim [13] focused on decentralized bilevel multiobjective fractional programming problems (DBL-MOFPPs) with a single decision-maker at the first level and multiple decision-makers at the second level, where the coefficients are crisp numbers. They presented an FGP based on Jacobian matrix for DBL-MOFPP. In the approach, membership functions of fuzzy goals are constructed for all objectives at two levels, and they are linearized by using the Jacobian matrix. Then the FGP approach is applied to obtain the highest degree of each of the membership goals by finding the most satisfactory solution for all decision-makers. This algorithm was applied for the crisp problems based on FGP. Peric, Babic, and Omerovi [15] studied a similar problem with crisp coefficients and presented a method based on the FGP. The approaches fail to solve the proposed model of this paper due to its fuzzy coefficients. Liu and Yang [18] proposed an interactive programming method to obtain the compromise optimal solution of the multilevel multiobjective linear programming problem with crisp parameters. The resolution process is done in two stages: analysis and decision-making. If the objective values were not satisfactory, then the decision-maker improves the values by giving a concession to the unsatisfied values. When the satisfactory degree in upper levels is mat, the problem of lower levels will be solved. The proposed method in [18] was designed for the case that the coefficients are crisp numbers. The proposed method cannot be applied to the proposed model with fuzzy coefficients. Kamal et al. [19] applied crisp bilevel multiobjective programming for production planning problems. Then an FGP algorithm was designed to solve the problem. This approach cannot also be applied to solve FMOLBP due to fuzziness of its coefficients. Abdelaziz and Mejri

[23] formulated a shared inventory model as a bilevel programming problem. Both emergency and backorders are considered in the model. The interaction between decision-makers is done in an uncertain environment of probability type. They designed a decentralized bilevel programming problem, where the leader has multiple objectives. The numerical study was focused on the simulation. We cannot use the approach to solve the proposed model with fuzzy coefficients because the approach in [23] discusses with the uncertainty of the probability type. Baky Eid, and El Sayed [3] extended the FGP to solve BL-MOPP with fuzzy demands that are given as triangular fuzzy numbers. The FGP algorithm is applied to attain the highest degree of each of the membership goals by minimizing their deviational variables and does not present the membership functions of the objectives of leader and follower, completely. In this paper, we will answer the following questions about the proposed method in [3]:

1- How can we find the membership functions of objectives of leader and follower instead of obtaining the highest degree of each of the membership goals?

2- Finding the required goals is not always easy, especially, when the dimension of the problem increases. How can we overcome these difficulties?

The answers to the above questions are the motivations of this paper. In this paper, we will consider the FMOLBP problem in a general case, where all coefficients are fuzzy numbers. Zadeh's extension principle [6, 40, 41] is applied to solve the problem. In this approach, a pair of crisp multiobjective linear three-level mathematical programming problem is formulated to compute the lower and upper bounds of the α-level of the objective values. Then, one of the scalarization methods, called weighting method, dual theory, and vertex enumeration approach are applied to solve the crisp multiobjective linear three-level programming problems. Finally, the membership function of the fuzzy compromise objective value is derived numerically by enumerating different values of α. This is the first contribution of the paper. In this approach, the KKT conditions are not used because of its nonlinearity nature. When the constraint functions at the upper level have an arbitrary linear form, the K-best algorithm cannot always find the optimal solution; see [31]. Hence, the most suitable method will be the vertex enumeration approach along with the weighting method and dual theory. These techniques preserve the linearity property of the problem. In addition, the results of the proposed approach are compared with the results of existing approaches. It shows the efficiency of the proposed approach with respect to the existing methods. This point is the second contribution of the paper.

The organization of this paper is as follows: Section 2 is formed of three subsections. The notations and properties of fuzzy numbers are reminded. Also, the weighting method is reviewed to solve multiobjective linear programming problem in the second subsection. In the third subsection, the multiobjective linear bilevel programming problem is formulated and an approach is presented to solve it based on the weighting method, dual theory, and vertex enumeration approach. Section 3 is divided into two subsections. The first subsection formulates the FMOLBP problem. The second subsection designs a solution procedure to solve the problem based on Zadeh's extension principle. Section 4 designs an efficient algorithm to solve the problem of FMOLBP based on the results of Section 3. A numerical example is presented to illustrate the algorithm in Section 5. Section 6 presents a comparison among the proposed algorithm and the existing methods to show the efficiency of the proposed algorithm. Conclusions are presented in Section 7.

2 Preliminaries

This section provides the required preliminaries in terms of three subsections. The preliminaries are about fuzzy numbers, weighting method in multiobjective linear programming problem, and multiobjective linear bilevel programming.

2.1 Fuzzy numbers

Let $\mathbb{R}$ be the set of all real numbers, let $\mathbb{R}^n$ be n-dimensional Euclidean space, and let $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ be two vectors, where $x_i, y_i \in \mathbb{R}$, $i = 1, \ldots, n$. For any two vectors $x, y \in \mathbb{R}^n$, we write $x \geq y$ if and only if $x_i \geq y_i$, for all $i = 1, \ldots, n$. The notation "o" displays the zero vector. Its dimension is determined in the context.

Assume $X \subseteq \mathbb{R}$. In this paper, it is supposed that the set X is equal to $\mathbb{R}$ or $[0, 1]$. If $\tilde{a}$ is a fuzzy subset of X with membership function $\mu_{\tilde{a}} : X \to [0, 1]$, then the α-level set of $\tilde{a}$ is as $\tilde{a}_\alpha = \{x \in X \mid \mu_{\tilde{a}}(x) \geq \alpha\}$ for all $\alpha \in (0, 1]$ and $\tilde{a}_0 = cl(\{x \in X \mid \mu_{\tilde{a}}(x) > 0\})$.

Definition 1. (a) Assume that $FN(X)$ is the set of all fuzzy subsets $\tilde{a}$ of X with $\mu_{\tilde{a}}$ satisfying the following conditions:
(i) There exists $x \in X$ such that $\mu_{\tilde{a}}(x) = 1$;
(ii) $\mu_{\tilde{a}}(\lambda.x + (1 - \lambda).y) \geq \min\{\mu_{\tilde{a}}(x), \mu_{\tilde{a}}(y)\}$ for all $x, y \in X$ and $\lambda \in [0, 1]$;
(iii) $\{x \in X \mid \mu_{\tilde{a}}(x) \geq \alpha\} = \tilde{a}_\alpha$ is a closed subset for each $\alpha \in (0, 1]$;
(iv) The 0-level set $\tilde{a}_0$ is a compact subset of X.
(b) A triangular fuzzy number is a fuzzy number that is presented by $\tilde{a} = (a_1, a_2, a_3)$, where $a_1 \leq a_2 \leq a_3$ and its membership function is as follows:

$$\mu_{\tilde{a}}(x) = \begin{cases} \frac{x-a_1}{a_2-a_1} & \text{if } a_1 \leq x \leq a_2, \\ \frac{a_3-x}{a_3-a_2} & \text{if } a_2 \leq x \leq a_3, \\ 0 & otherwise. \end{cases}$$

With regard to Definition 1, the α-level set $\tilde{a}$ can be written as $\tilde{a}_\alpha = [\tilde{a}^L_\alpha, \tilde{a}^U_\alpha]$; see [40]. In the triangular case, the α-level set $\tilde{a}$ is as $\tilde{a}_\alpha = [a_1 + (a_2 - a_1).\alpha, a_3 - (a_3 - a_2).\alpha]$. If $\tilde{a}$ is a crisp number with the value m, then its membership function is as follows:

$$\mu_{\tilde{a}}(x) = \begin{cases} 1 & \text{if } x = m, \\ 0 & otherwise. \end{cases} \tag{1}$$

The set of all finite fuzzy numbers on $\mathbb{R}$ is denoted by $FN^*(\mathbb{R})$.

Definition 2. Let $\tilde{a}_i \in FN(\mathbb{R})$ for $i = 1, \ldots, n$. We define $\tilde{a} = (\tilde{a}_1, \ldots, \tilde{a}_n)$ and $\mu_{\tilde{a}} : \mathbb{R}^n \to [0, 1]$ by $\mu_{\tilde{a}}(x) = \min_{i=1,\ldots,n} \mu_{\tilde{a}_i}(x_i)$, where $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, and $\tilde{a}$ is called an n-dimensional fuzzy number on $\mathbb{R}^n$. If $\tilde{a}_i \in FN^*(\mathbb{R})$, $i = 1, \ldots, n$, then $\tilde{a}$ is called an n-dimensional finite fuzzy number on $\mathbb{R}^n$. Let $FN(\mathbb{R}^n)$ and $FN^*(\mathbb{R}^n)$ be the set of all n-dimensional fuzzy numbers and the set of all n-dimensional finite fuzzy numbers on $\mathbb{R}^n$, respectively.

2.2 Weighting method in multiobjective linear programming (MOLP) problem

Consider a multiobjective linear programming (MOLP) problem

$$\min \quad z(x) = (z_1(x), \ldots, z_k(x)), \tag{2}$$
$$\text{s.t.} \quad x \in X \subseteq \mathbb{R}^n, \tag{3}$$

where $x \in \mathbb{R}^n$ and X denote the decision-making variables and feasible set, respectively. Moreover, $Z : \mathbb{R}^n \longrightarrow \mathbb{R}^k$ and $z_i(x)$, $i = 1, \ldots, k$, are real-valued functions over X.

Definition 3 ([17])**.** The vector x^* is said to be a Pareto optimal solution (or compromise solution) if and only if there does not exist another $x \in X$ such that $z_i(x) \leq z_i(x^*)$ for all i and $z_j(x) \neq z_j(x^*)$ for at least one j.

The weighting method for obtaining a Pareto optimal solution is to solve the weighting problem formulated by taking the weighted sum of all of the objective functions of the original MOLP (1)–(3). Thus, the weighting problem is defined by

$$\min \quad W.z(x) = \sum_{i=1}^{k} w_i.z_i(x), \tag{4}$$

$$s.t. \qquad x \in X, \quad \sum_{i=1}^{k} w_i = 1, \quad w_i \geq 0, \quad for \quad i = 1, \ldots, k, \tag{5}$$

where $W = (w_1, \ldots, w_k)$ is the vector of weighting coefficients assigned to the objective functions, and it is assumed that $W = (w_1, \ldots, w_k) \geq o$. The following theorem presents a relationship between the Pareto optimal solution of problem (1)–(3) and the optimal solution of problem (2)–(5).

Theorem 1 ([17]). If $x^* \in X$ is an optimal solution of the weighting problem (2)–(5) for some $W > o$, then x^* is a Pareto optimal solution of the MOLP (1)–(3). Conversely, if $x^* \in X$ is a Pareto optimal solution of the MOLP (1)–(3), then x^* is an optimal solution of the weighting problem (2)–(5) for some $W = (w_1, \ldots, w_k) \geq o$.

2.3 Multiobjective linear bilevel programming(MOLBP)

First of all, throughout this paper, we assume that $X = \{x \in \mathbb{R}^n | x \geq o\}$ and $Y = \{y \in \mathbb{R}^m | y \geq o\}$. The MOLBP problem is formulated as follows:
For $x \in X \subseteq \mathbb{R}^n$, $y \in Y \subseteq \mathbb{R}^m$, $F : X \times Y \longrightarrow \mathbb{R}^s$, and $f : X \times Y \longrightarrow \mathbb{R}^t$, we define

$$\begin{aligned}
& Z(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2) \\
& = \min_{x \in X} F(x, y) = (c_1.x + d_1.y, \ldots, c_s.x + d_s.y)^T, \qquad (6) \\
& s.t. \quad A_1.x + B_1.y \leq k_1, \qquad (7) \\
& \qquad \min_{y \in Y} f(x, y) = (c'_1.x + d'_1.y, \ldots, c'_t.x + d'_t.y)^T, \qquad (8) \\
& \qquad s.t. \quad A_2.x + B_2.y \leq k_2, \qquad (9)
\end{aligned}$$

where

$$c_i = (c_{1i}, \ldots, c_{ni}),\ c'_j = (c'_{1j}, \ldots, c'_{nj}) \in \mathbb{R}^n, \quad i = 1, \ldots, s, j = 1, \ldots, t,$$

$$d_i = (d_{1i}, \ldots, d_{mi}),\ d'_j = (d'_{1j}, \ldots, d'_{mj}) \in \mathbb{R}^m, \quad i = 1, \ldots, s,\ j = 1, \ldots, t,$$

$k_1 = (k_{11}, \ldots, k_{p1}) \in \mathbb{R}^p$, $k_2 = (k_{12}, \ldots, k_{q2}) \in \mathbb{R}^q$, $A_1 = [a_{ij}] \in \mathbb{R}^{p \times n}$, $B_1 = [b_{ij}] \in \mathbb{R}^{p \times m}$, $A_2 = [e_{ij}] \in \mathbb{R}^{q \times n}$, and $B_2 = [s_{ij}] \in \mathbb{R}^{q \times m}$.
Let $S = \{(x, y) \in X \times Y | A_1.x + B_1.y \leq k_1, A_2.x + B_2.y \leq k_2\}$ and $S(x) = \{y \in Y \mid B_2.y \leq k_2 - A_2.x\}$. Denote by $R(x)$ the set of the Pareto optimal solutions of the lower level problem for any fixed x. Then we have

the following definitions.

Definition 4 ([8])**.** For a fixed $(x, y) \in S$, if y is a Pareto optimal solution to the lower level problem, then (x, y) is a feasible solution to problem (6)–(6).

Definition 5 ([8])**.** If (x, y) is a feasible solution to problem (6)–(6) and x is a Pareto optimal solution to the upper level problem for fixed y, then (x, y) is a Pareto optimal solution to problem (6)–(6).

We now apply the weighting method and rewrite problem (6)–(6) as follows:

$$\overline{Z}(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2)$$

$$= \min_{x \in X} \quad W.F(x, y) = \sum_{i=1}^{s} w_i.(c_i.x + d_i.y), \tag{10}$$

$$s.t. \quad A_1.x + B_1.y \leq k_1, \sum_{i=1}^{s} w_i = 1, w_i \geq 0, \quad for \quad i = 1, \ldots, s, \tag{11}$$

$$\min_{y \in Y} \quad W'.f(x, y) = \sum_{i=1}^{t} w'_i.(c'_i.x + d'_i.y), \tag{12}$$

$$s.t. \quad A_2.x + B_2.y \leq k_2, \sum_{i=1}^{t} w'_i = 1, w'_i \geq 0, \quad for \quad i = 1, \ldots, t,. \tag{13}$$

For some fixed $x \geq o$, the compromise solution of the lower programming can be found by solving the following linear programming with a given weight $W' \geq o$.

$$\min_{y} \quad \sum_{i=1}^{t} w'_i.(c'_i.x + d'_i.y), \tag{14}$$

$$s.t. \quad B_2.y \leq k_2 - A_2.x, \tag{15}$$

$$\sum_{i=1}^{t} w'_i = 1, w'_i \geq 0, \quad for \quad i = 1, \ldots, t, y \geq o. \tag{16}$$

Since $c'_i.x$, for $i = 1, \ldots, t$, are constant, we can ignore the terms. Hence, a dual multiplier u_i is considered for each constraint (15), the primal variables are in terms of the vector of y, and other expressions are considered as constant. Thus, we can get the dual of problem (14)–(16) according to the dual definition for linear programming problems in [40] as follows:

$$\max_{u} \quad (k_2 - A_2.x)^T.u, \tag{17}$$

$$s.t. \quad B_2^T.u \geq \sum_{i=1}^{t} w_i'.d_i', \tag{18}$$

$$\sum_{i=1}^{t} w_i' = 1, \; w_i' \geq 0, \quad for \quad i = 1, \ldots, t, \; u \geq o, \tag{19}$$

where $u \in \mathbb{R}^q$ is the dual multipliers. The following theorem converts MOLBP problem (6)–(6) to a one-level programming problem with one objective function.

Theorem 2. The pair (x^*, y^*) is the compromise solution of problem (6)–(6) if and only if there exists u^* such that (x^*, y^*, u^*) is the solution of the following program:

$$\min \quad F(x,y) = \sum_{i=1}^{s} w_i.(c_i.x + d_i.y), \tag{20}$$

$$s.t. \quad A_1.x + B_1.y \leq k_1, \tag{21}$$

$$B_2^T.u \geq \sum_{i=1}^{t} w_i'.d_i', \tag{22}$$

$$\sum_{i=1}^{t} w_i'.(d_i'.y) - (k_2 - A_2.x)^T.u = 0, \tag{23}$$

$$\sum_{i=1}^{t} w_i' = 1, w_i' \geq 0, \quad for \quad i = 1, \ldots, t, \quad x, y, u \geq o. \tag{24}$$

Proof. The proof is similar to the proof of Theorem 1 in [33]. □

Let $U = \{u | B_2^T.u \geq \sum_{i=1}^{t} w_i'.d_i', u \geq o\}$ denote the feasible region to linear programming (17)–(19). If $U \neq \emptyset$, then the set U has at least one vertex and at most finite vertices. Moreover, if there is an optimal solution to the linear programming (17)–(19), then it must be one of the vertex points of set U (see [28]). With attention to the above results, we can convert problem (20)–(24) to a series of the following linear programming problems by obtaining all vertices of U, denoted by $U^v = \{u^1, u^2, \ldots, u^r\}$, according to the method in linear programming problem from [28].

$$LP(u^p): \min_{x,y} \quad F(x,y) = \sum_{i=1}^{s} w_i.(c_i.x + d_i.y), \tag{25}$$

$$s.t. \quad A_1.x + B_1.y \leq k_1, \tag{26}$$

$$\sum_{i=1}^{t} w'_i.(d'_i.y) - (k_2 - A_2.x)^T.u^p = 0, \tag{27}$$

$$\sum_{i=1}^{t} w'_i = 1, w'_i \geq 0, \quad for \quad i = 1,\ldots,t, \quad x, y \geq o. \tag{28}$$

We can solve the above $LP(u^p)$ for $p = 1,\ldots,r$. Let $I \subseteq \{1,\ldots,r\}$ denote the Indices of problems that $LP(u^i)$, for $i \in I$, has an optimal solution. For $i \in I$, let (x^i, y^i) be the optimal solution for problem $LP(u^i)$ and $F(x^k, y^k) = \min\{F(x^i, y^i)|i \in I\}$. Therefore, we have the following theorem.

Theorem 3. The pair (x^k, y^k) is an optimal solution for problem (20)–(24) and a Pareto optimal solution for problem (6)–(6).

We are now ready to formulate and solve fuzzy multiobjective linear bilevel programming.

3 Fuzzy multiobjective linear bilevel programming

In this section, the FMOLBP problem is formulated and a procedure is designed to solve the problem based on Zadeh's extension principle [6, 40, 41].

3.1 The formulation of FMOLBP

For $x \in X \subseteq \mathbb{R}^n, y \in Y \subseteq \mathbb{R}^m$, $\tilde{F} : X \times Y \longrightarrow FN^*(\mathbb{R}^s)$, and $\tilde{f} : X \times Y \longrightarrow FN^*(\mathbb{R}^t)$, the problem of FMOLBP is formulated by

$$\tilde{Z} = \min_{x \in X} \quad \tilde{F}(x,y) = (\tilde{c}_1.x + \tilde{d}_1.y, \ldots, \tilde{c}_s.x + \tilde{d}_s.y)^T, \tag{29}$$

$$s.t. \quad \tilde{A}_1.x + \tilde{B}_1.y \leq \tilde{k}_1, \tag{30}$$

$$\min_{y \in Y} \quad \tilde{f}(x,y) = (\tilde{c}'_1.x + \tilde{d}'_1.y, \ldots, \tilde{c}'_t.x + \tilde{d}'_t.y)^T, \tag{31}$$

$$s.t. \quad \tilde{A}_2.x + \tilde{B}_2.y \leq \tilde{k}_2, \tag{32}$$

$$x \geq o, \quad y \geq o, \tag{33}$$

where

$$\tilde{c}_i = (\tilde{c}_{1i}, \ldots, \tilde{c}_{ni}), \tilde{c}'_j = (\tilde{c}'_{1j}, \ldots, \tilde{c}'_{nj}) \in FN^*(\mathbb{R}^n), \; i = 1,\ldots,s, \; j = 1,\ldots,t,$$

$\tilde{d}_i = (\tilde{d}_{1i}, \ldots, \tilde{d}_{mi}), \tilde{d}'_j = (\tilde{d}'_{1j}, \ldots, \tilde{d}'_{mj}) \in FN^*(\mathbb{R}^m),\ i = 1, \ldots, s,\ j = 1, \ldots, t,$

$\tilde{k}_1 = (\tilde{k}_{11}, \ldots, \tilde{k}_{p1}) \in FN^*(\mathbb{R}^p),\ \tilde{k}_2 = (\tilde{k}_{12}, \ldots, \tilde{k}_{q2}) \in FN^*(\mathbb{R}^q),$
$\tilde{A}_1 = [\tilde{a}_{ij}]_{p\times n},\ \tilde{a}_{ij} \in FN^*(\mathbb{R}),\ \tilde{B}_1 = [\tilde{b}_{ij}]_{p\times m},\ \tilde{b}_{ij} \in FN^*(\mathbb{R}),$
$\tilde{A}_2 = [\tilde{e}_{ij}]_{q\times n},\ \tilde{e}_{ij} \in FN^*(\mathbb{R})$, and $\tilde{B}_2 = [\tilde{s}_{ij}]_{q\times m},\ \tilde{s}_{ij} \in FN^*(\mathbb{R})$.
The feasible solution set of FMOLBP is defined as follows:

$$S = \{(x, y) \in X \times Y | \tilde{A}_1.x + \tilde{B}_1.y \le \tilde{k}_1, \tilde{A}_2.x + \tilde{B}_2.y \le \tilde{k}_2\}. \tag{34}$$

In an FMOLBP problem, for each $(x, y) \in S$, the value of the objective functions $\tilde{F}(x, y) = (\tilde{F}_1(x, y), \ldots, \tilde{F}_s(x, y))$ and $\tilde{f}(x, y) = (\tilde{f}_1(x, y), \ldots, \tilde{f}_t(x, y))$ of leader and follower is s-dimension and t-dimension fuzzy vectors, respectively.

3.2 The solution procedure for the problem (29)–(33)

We will now consider problem (29)–(33). Let $\mu_{\tilde{c}_{1i}}, \ldots, \mu_{\tilde{c}_{ni}}, \mu_{\tilde{c}'_{1j}}, \ldots, \mu_{\tilde{c}'_{nj}}, \mu_{\tilde{d}_{1i}}, \ldots, \mu_{\tilde{d}_{mi}}, \mu_{\tilde{d}'_{1j}}, \ldots, \mu_{\tilde{d}'_{mj}}, i = 1, \ldots, s,\ j = 1, \ldots, t,\ \mu_{\tilde{k}_{i1}}, \mu_{\tilde{k}_{j2}}, i = 1, \ldots, p,\ j = 1, \ldots, q,\ \mu_{\tilde{a}_{ij}}, i = 1, \ldots, p, j = 1, \ldots, n$, $\mu_{\tilde{b}_{ij}}, i = 1, \ldots, p, j = 1, \ldots, m,\ \mu_{\tilde{e}_{ij}}, i = 1, \ldots, q, j = 1, \ldots, n$, and $\mu_{\tilde{s}_{ij}}$, $i = 1, \ldots, q, j = 1, \ldots, m$, denote the membership functions $\tilde{c}_{1i}, \ldots, \tilde{c}_{ni}$, $\tilde{c}'_{1j}, \ldots, \tilde{c}'_{nj},\ \tilde{d}_{1i}, \ldots, \tilde{d}_{mi},\ \tilde{d}'_{1j}, \ldots, \tilde{d}'_{mj},\ i = 1, \ldots, s, j = 1, \ldots, t,\ \tilde{k}_{i1},\ \tilde{k}_{j2}$, $i = 1, \ldots, p, j = 1, \ldots, q,\ \tilde{a}_{ij},\ i = 1, \ldots, p, j = 1, \ldots, n,\ \tilde{b}_{ij},\ i = 1, \ldots, p, j = 1, \ldots, m,\ \tilde{e}_{ij},\ i = 1, \ldots, q, j = 1, \ldots, n$, and $\tilde{s}_{ij},\ i = 1, \ldots, q, j = 1, \ldots, m$, respectively.

We have $\tilde{v} = \{(v, \mu_{\tilde{v}}(v)) \mid v \in S(\tilde{v})\}$, where $\tilde{v}$ is one of the parameters $\tilde{c}_{1i}, \ldots, \tilde{c}_{ni},\ \tilde{c}'_{1j}, \ldots, \tilde{c}'_{nj},\ \tilde{d}_{1i}, \ldots, \tilde{d}_{mi},\ \tilde{d}'_{1j}, \ldots, \tilde{d}'_{mj},\ i = 1, \ldots, s, j = 1, \ldots, t$, $\tilde{k}_{i1},\ \tilde{k}_{j2},\ i = 1, \ldots, p, j = 1, \ldots, q,\ \tilde{a}_{ij},\ i = 1, \ldots, p, j = 1, \ldots, n,\ \tilde{b}_{ij},\ i = 1, \ldots, p, j = 1, \ldots, m,\ \tilde{e}_{ij},\ i = 1, \ldots, q, j = 1, \ldots, n,\ \tilde{s}_{ij},\ i = 1, \ldots, q, j = 1, \ldots, m$, and $S(\tilde{v})$ is the support of $\tilde{v}$. Denote the α-cuts of $\tilde{v}$ by

$$(\tilde{v})_\alpha = [(\tilde{v})^L_\alpha, (\tilde{v})^U_\alpha] = [\min_v \{v \in S(\tilde{v}) \mid \mu_{\tilde{v}}(v) \ge \alpha\},$$
$$\max_v \{v \in S(\tilde{v}) \mid \mu_{\tilde{v}}(v) \ge \alpha\}], \tag{35}$$

where $\tilde{v}$ is one of the above fuzzy parameters. These intervals show that the coefficients of objective function and constraints in problem (29)–(33) lie at the α-level set. Here, we intend to obtain the membership function $\tilde{Z}$. To do this, Zadeh's extension principle [6, 40, 41] is used. Based on the extension principle, $\mu_{\tilde{Z}}$ can be written by

$$\mu_{\tilde{Z}}(z) = \sup_{\substack{c_i,\, c'_j,\, d_i,\, d'_j,\\ k_1,\, k_2,\, A_1,\, B_1,\\ A_2,\, B_2.}} \min\{\mu_{\tilde{c}_{1i}}(c_{1i}), \ldots, \mu_{\tilde{c}_{ni}}(c_{ni}), \mu_{\tilde{c}'_{1j}}(c'_{1j}), \ldots, \mu_{\tilde{c}'_{nj}}(c'_{nj}),$$
$$\mu_{\tilde{d}_{1i}}(d_{1i}), \ldots, \mu_{\tilde{d}_{mi}}(d_{mi}), \mu_{\tilde{d}'_{1j}}(d'_{1j}), \ldots, \mu_{\tilde{d}'_{mj}}(d'_{mj}), \mu_{\tilde{k}_{i1}}(k_{i1}),$$
$$\mu_{\tilde{k}_{j2}}(k_{j2}), \mu_{\tilde{a}_{ij}}(a_{ij}), \mu_{\tilde{b}_{ij}}(b_{ij}), \mu_{\tilde{e}_{ij}}(e_{ij}), \mu_{\tilde{s}_{ij}}(s_{ij}), \text{ for all } i,j \mid$$
$$z = Z(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2)\}, \quad (36)$$

where matrices $c_i = [c_{li}]_{1\times n}$, $c'_j = [c'_{lj}]_{1\times n}$, $d_i = [d_{li}]_{1\times m}$, $d'_j = [d'_{lj}]_{1\times m}$, $i = 1, \ldots, s, j = 1, \ldots, t$, $k_1 = [k_{i1}]_{p\times 1}$, $k_2 = [k_{j2}]_{q\times 1}$, $A_1 = [a_{ij}]_{p\times n}$, $B_1 = [b_{ij}]_{p\times m}$, $A_2 = [e_{ij}]_{q\times n}$, and $B_2 = [s_{ij}]_{q\times m}$, are the supports of $\tilde{c}_i, \tilde{c}'_j, \tilde{d}_i, \tilde{d}'_j$, $i = 1, \ldots, s, j = 1, \ldots, t$, $\tilde{k}_1, \tilde{k}_2$, $\tilde{A}_1, \tilde{B}_1$, $\tilde{A}_2$, and $\tilde{B}_2$, respectively. Also $Z(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2)$ is defined as follows:

$$\begin{aligned}
&Z(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2)\\
&= \min_{x\in X} F(x,y) = (c_1.x + d_1.y, \ldots, c_s.x + d_s.y)^T,\\
&s.t. \quad A_1.x + B_1.y \le k_1,\\
&\min_{y\in Y} f(x,y) = (c'_1.x + d'_1.y, \ldots, c'_t.x + d'_t.y)^T,\\
&s.t. \quad A_2.x + B_2.y \le k_2.
\end{aligned} \quad (37)$$

If the α-cuts of $\tilde{Z}$ at all α values are converted to the same point, then the objective functions are crisp numbers. Otherwise, they are fuzzy numbers. According to (36), $\mu_{\tilde{Z}}$ is the minimum of $\mu_{\tilde{v}}(v)$ for $v = c_i, c'_j, d_i, d'_j, k_{i1}, k_{j2}, a_{ij}, b_{ij}, e_{ij}, s_{ij},$ for all i,j. We need $\mu_{\tilde{v}}(v) \ge \alpha$, for $v = c_i, c'_j, d_i, d'_j, k_{i1}, k_{j2}, a_{ij}, b_{ij}, e_{ij}, s_{ij},$ for all i,j, and at least one $\mu_{\tilde{v}}(v) \ge \alpha$, for $v = c_i, c'_j, d_i, d'_j, k_{i1}, k_{j2}, a_{ij}, b_{ij}, e_{ij}, s_{ij},$ for all i,j, equal to α such that $z = Z(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2)$ to satisfy $\mu_{\tilde{Z}}(z) = \alpha$. To compute the membership function $\mu_{\tilde{Z}}$, we intend to compute the left and right shape function of $\mu_{\tilde{Z}}$. It is equivalent to finding $\tilde{Z}^L_\alpha$ and $\tilde{Z}^U_\alpha$ of the α-cuts of $\tilde{Z}$. Since $\tilde{Z}^L_\alpha$ is the minimum of $Z(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2)$ and $\tilde{Z}^U_\alpha$ is the maximum of $Z(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2)$, we have

$$\begin{aligned}
\tilde{Z}^L_\alpha = \min\{&Z(c_1, \ldots, c_s, d_1, \ldots, d_s, c'_1, \ldots, c'_t, d'_1, \ldots, d'_t, A_1, B_1, A_2, B_2, k_1, k_2)\\
&\mid (\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha, where\ v = c_{1i}, \ldots, c_{ni}, d_{1i}, \ldots, d_{mi}, c'_{1j}, \ldots, c'_{nj},\\
&d'_{1j}, \ldots, d'_{mj}, a_{ij}, b_{ij}, e_{ij}, s_{ij}, k_{i1}, k_{j2}, \text{ for all } i,j\}
\end{aligned} \quad (38)$$

and

$$\begin{aligned}\tilde{Z}^U_\alpha = \max\{ & Z(c_1,\ldots,c_s,d_1,\ldots,d_s,c'_1,\ldots,c'_t,d'_1,\ldots,d'_t,A_1,B_1,A_2,B_2,k_1,k_2) \\ & \mid \tilde{v}^L_\alpha \le v \le \tilde{v}^U_\alpha, where \;\; v = c_{1i},\ldots,c_{ni},d_{1i},\ldots,d_{mi},c'_{1j},\ldots,c'_{nj}, \\ & d'_{1j},\ldots,d'_{mj},a_{ij},b_{ij},e_{ij},s_{ij},k_{i1},k_{j2}, \text{ for all } i,j\}. \end{aligned} \tag{39}$$

They can equivalently be rewritten as two problems below:

$$\tilde{Z}^L_\alpha = \min_{\substack{(\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha, where \\ v = c_{1i},\ldots,c_{ni},d_{1i},\ldots,d_{mi}, \\ c'_{1j},\ldots,c'_{nj},d'_{1j},\ldots,d'_{mj}, \\ a_{ij},b_{ij},e_{ij},s_{ij},k_{i1},k_{j2}, \text{ for all } i,j.}} \begin{cases} \min\limits_{x\in X}(c_1x+d_1y,\ldots,c_sx+d_sy)^T, \\ \quad s.t. \;\; A_1x+B_1y \le k_1, \\ \qquad \min\limits_{y\in Y}(c'_1x+d'_1y,\ldots,c'_tx+d'_ty)^T, \\ \qquad\quad s.t. \;\; A_2x+B_2y \le k_2, \end{cases} \tag{40}$$

and

$$\tilde{Z}^U_\alpha = \max_{\substack{(\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha, where \\ v = c_{1i},\ldots,c_{ni},d_{1i},\ldots,d_{mi}, \\ c'_{1j},\ldots,c'_{nj},d'_{1j},\ldots,d'_{mj}, \\ a_{ij},b_{ij},e_{ij},s_{ij},k_{i1},k_{j2}, \text{ for all } i,j.}} \begin{cases} \min\limits_{x\in X}(c_1x+d_1y,\ldots,c_sx+d_sy)^T, \\ \quad s.t. \;\; A_1x+B_1y \le k_1, \\ \qquad \min\limits_{y\in Y}(c'_1x+d'_1y,\ldots,c'_tx+d'_ty)^T, \\ \qquad\quad s.t. \;\; A_2x+B_2y \le k_2. \end{cases} \tag{41}$$

We can use the weighting method for the problem to find the compromise solutions of problem (40). Hence, we can rewrite problem (40) as follows:

$$\overline{\tilde{Z}}^L_\alpha = \max_{\substack{(\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha, where \\ v = c_{1i},\ldots,c_{ni},d_{1i},\ldots,d_{mi}, \\ c'_{1j},\ldots,c'_{nj},d'_{1j},\ldots,d'_{mj}, \\ a_{ij},b_{ij},e_{ij},s_{ij},k_{i1},k_{j2}, \text{ for all } i,j.}} \begin{cases} \min\limits_{x\in X} \sum\limits_{i=1}^{s} w_i(c_i.x) + \sum\limits_{i=1}^{s} w_i(d_i.y), \\ \quad s.t. \;\; A_1x+B_1y \le k_1, \\ \qquad \sum\limits_{i=1}^{s} w_i = 1, w_i \ge 0, for \;\; i=1,\ldots,s, \\ \qquad \min\limits_{y\in Y} \sum\limits_{i=1}^{t} w'_i(c'_i.x) + \sum\limits_{i=1}^{t} w'_i(d'_i.y), \\ \qquad s.t. \;\; A_2x+B_2y \le k_2, \\ \qquad \sum\limits_{i=1}^{t} w'_i = 1, w'_i \ge 0, for \;\; i=1,\ldots,t. \end{cases} \tag{42}$$

Since model (42) finds the minimum of all minimum objective values, we should consider the objective function of leader in the best case, that is, we set c_i and d_i to their lower bound, in other words, $(\tilde{c}_i)^L_\alpha$ and $(\tilde{d}_i)^L_\alpha$, for each i, respectively. Therefore, we can rewrite problem (42) as follows:

$$
\overline{\tilde{Z}}^L_\alpha = \max_{\substack{(\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha,\, where \\ v = c'_{1j},\ldots,c'_{nj},d'_{1j},\ldots,d'_{mj}, \\ a_{ij},b_{ij},e_{ij},s_{ij},k_{i1},k_{j2},\ \text{for all } i,j.}} \begin{cases} \min\limits_{x\in X} \sum\limits_{i=1}^{s} w_i((\tilde{c}_i)^L_\alpha.x) + \sum\limits_{i=1}^{s} w_i((\tilde{d}_i)^L_\alpha.y), \\ \quad s.t. \ \ A_1x + B_1y \le k_1, \\ \qquad \sum\limits_{i=1}^{s} w_i = 1, w_i \ge 0, for\ i = 1,\ldots,s, \\ \qquad \min\limits_{y\in Y} \sum\limits_{i=1}^{t} w'_i(c'_i.x) + \sum\limits_{i=1}^{t} w'_i(d'_i.y), \\ \qquad s.t. \ \ A_2x + B_2y \le k_2, \\ \qquad \sum\limits_{i=1}^{t} w'_i = 1, w'_i \ge 0, for\ i = 1,\ldots,t. \end{cases} \tag{43}
$$

Notation 1. Assume that $\tilde{A} = [\tilde{a}_{ij}]_{m\times n}$ is a fuzzy matrix with fuzzy components. Then $(\tilde{A})^L_\alpha = [(\tilde{a}_{ij})^L_\alpha]_{m\times n}$ and $(\tilde{A})^U_\alpha = [(\tilde{a}_{ij})^U_\alpha]_{m\times n}$.

To obtain the minimum objective value, we should consider the values of A_1, B_1, A_2, B_2, k_1, and k_2 in $(\tilde{A}_1)^U_\alpha, (\tilde{B}_1)^U_\alpha, (\tilde{A}_2)^U_\alpha, (\tilde{B}_2)^U_\alpha, (\tilde{k}_1)^L_\alpha$, and $(\tilde{k}_2)^L_\alpha$, respectively, because we can obtain the largest feasible solution set, in this case. Hence, the least value is found for the objective function. To this end, problem (43) is converted to the following problem.

$$
\overline{\tilde{Z}}^L_\alpha = \min_{\substack{(\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha,\, where \\ v = c'_{1j},\ldots,c'_{nj},d'_{1j},\ldots,d'_{mj}, \\ \text{for all } j.}} \begin{cases} \min\limits_{x\in X} \sum\limits_{i=1}^{s} w_i((\tilde{c}_i)^L_\alpha.x) + \sum\limits_{i=1}^{s} w_i((\tilde{d}_i)^L_\alpha.y), \\ \quad s.t. \ \ (\tilde{A}_1)^U_\alpha x + (\tilde{B}_1)^U_\alpha y \le (\tilde{k}_1)^L_\alpha, \\ \qquad \sum\limits_{i=1}^{s} w_i = 1, w_i \ge 0, for\ i = 1,\ldots,s, \\ \qquad \min\limits_{y\in Y} \sum\limits_{i=1}^{t} w'_i(c'_i.x) + \sum\limits_{i=1}^{t} w'_i(d'_i.y), \\ \qquad s.t. \ \ (\tilde{A}_2)^U_\alpha x + (\tilde{B}_2)^U_\alpha y \le (\tilde{k}_2)^L_\alpha, \\ \qquad \sum\limits_{i=1}^{t} w'_i = 1, w'_i \ge 0, for\ i = 1,\ldots,t. \end{cases} \tag{44}
$$

With regard to Theorem 2, we can write the inner problem as follows:

$$
\overline{\tilde{Z}}^{L}_{\alpha} = \min_{\substack{(\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha, \\ where \;\; v = c'_{1j}, \dots, c'_{nj}, \\ d'_{1j}, \dots, d'_{mj}, \;\; \text{for all } j.}} \begin{cases} \min \sum_{i=1}^{s} w_i((\tilde{c}_i)^L_\alpha . x) + \sum_{i=1}^{s} w_i((\tilde{d}_i)^L_\alpha . y), \\ s.t. \;\; (\tilde{A}_1)^U_\alpha x + (\tilde{B}_1)^U_\alpha y \le (\tilde{k}_1)^L_\alpha, \\ \quad \sum_{i=1}^{s} w_i = 1, w_i \ge 0, for \; i = 1, \dots, s, \\ \quad \sum_{i=1}^{t} w'_i (d'_i . y) - ((\tilde{k}_2)^L_\alpha - (\tilde{A}_2)^U_\alpha x)^T . u = 0, \\ \quad ((\tilde{B}_2)^U_\alpha)^T . u \ge \sum_{i=1}^{t} w'_i . d'_i, \\ \quad \sum_{i=1}^{t} w'_i = 1, w'_i \ge 0, for \; i = 1, \dots, t. \end{cases} \tag{45}
$$

Or equivalently, we can rewrite problem (45) as follows:

$$
\overline{\tilde{Z}}^{L}_{\alpha} = \min \sum_{i=1}^{s} w_i((\tilde{c}_i)^L_\alpha . x) + \sum_{i=1}^{s} w_i((\tilde{d}_i)^L_\alpha . y), \tag{46}
$$

$$
s.t. \;\; (\tilde{A}_1)^U_\alpha x + (\tilde{B}_1)^U_\alpha y \le (\tilde{k}_1)^L_\alpha, \tag{47}
$$

$$
\sum_{i=1}^{t} w'_i (d'_i . y) - ((\tilde{k}_2)^L_\alpha - (\tilde{A}_2)^U_\alpha x)^T . u = 0, \tag{48}
$$

$$
((\tilde{B}_2)^U_\alpha)^T . u \ge \sum_{i=1}^{t} w'_i . d'_i, \tag{49}
$$

$$
(\tilde{d}'_i)^L_\alpha \le d'_i \le (\tilde{d}'_i)^U_\alpha, \quad \text{for all } i = 1, \dots, t, \tag{50}
$$

$$
\sum_{i=1}^{s} w_i = 1, \sum_{i=1}^{t} w'_i = 1, w_i, w'_j \ge 0, for \;\; i = 1, \dots, s, \; j = 1, \dots, t,
$$

$$
x, y, u \ge o. \tag{51}
$$

We can now remove the constraint (50) and equivalently rewrite problem (46)–(51) as follows:

$$\overline{\tilde{Z}}^L_\alpha = \min \sum_{i=1}^{s} w_i((\tilde{c}_i)^L_\alpha.x) + \sum_{i=1}^{s} w_i((\tilde{d}_i)^L_\alpha.y), \tag{52}$$

$$s.t.\ \ (\tilde{A}_1)^U_\alpha x + (\tilde{B}_1)^U_\alpha y \leq (\tilde{k}_1)^L_\alpha, \tag{53}$$

$$\sum_{i=1}^{t} w'_i((\tilde{d'}_i)^L_\alpha.y) \leq ((\tilde{k}_2)^L_\alpha - (\tilde{A}_2)^U_\alpha x)^T.u \leq \sum_{i=1}^{t} w'_i((\tilde{d'}_i)^U_\alpha.y), \tag{54}$$

$$((\tilde{B}_2)^U_\alpha)^T.u \geq \sum_{i=1}^{t} w'_i.(\tilde{d}'_i)^U_\alpha, \tag{55}$$

$$\sum_{i=1}^{s} w_i = 1, \sum_{i=1}^{t} w'_i = 1, w_i, w'_j \geq 0, for\ \ i = 1,\ldots,s,\ j = 1,\ldots,t,$$
$$x, y, u \geq o. \tag{56}$$

Let $U = \{u \geq o \mid ((\tilde{B}_2)^U_\alpha)^T.u \geq \sum_{i=1}^{t} w'_i.(\tilde{d}'_i)^U_\alpha\}$. Assume that the vertex points of U are denoted by the set as $U^V = \{u^1,\ldots,u^r\}$. Hence, we should solve at most r linear programming problems as follows:

$$LP(u^k) : \overline{\tilde{Z}}^L_{\alpha,k} = \min \sum_{i=1}^{s} w_i((\tilde{c}_i)^L_\alpha.x) + \sum_{i=1}^{s} w_i((\tilde{d}_i)^L_\alpha.y), \tag{57}$$

$$s.t.\ \ (\tilde{A}_1)^U_\alpha x + (\tilde{B}_1)^U_\alpha y \leq (\tilde{k}_1)^L_\alpha, \tag{58}$$

$$\sum_{i=1}^{t} w'_i((\tilde{d'}_i)^L_\alpha.y) \leq ((\tilde{k}_2)^L_\alpha - (\tilde{A}_2)^U_\alpha x)^T.u^k \leq \sum_{i=1}^{t} w'_i((\tilde{d'}_i)^U_\alpha.y), \tag{59}$$

$$\sum_{i=1}^{s} w_i = 1, \sum_{i=1}^{t} w'_i = 1, w_i, w'_j \geq 0,$$
$$for\ \ i = 1,\ldots,s,\ = 1,\ldots,t,\ x, y \geq o, \tag{60}$$

where $k = 1,\ldots,r$. Solving r linear programming problems $LP(u^k)$, we can find the optimal solution of the problem for each $\alpha \in [0,1]$ as it was mentioned in Subsection 2.3. Finding the optimal values of variables x and y for each $\alpha \in [0,1]$, we can estimate values $(\tilde{c}_i)^L_\alpha.x + (\tilde{d}_i)^L_\alpha.y$ for each $\alpha \in [0,1]$ and $i = 1,\ldots,s$. Hence, we can obtain $\tilde{Z}^L_\alpha$.
We now focus on obtaining the $\tilde{Z}^U_\alpha$. To do this, the weighting method is applied to the problem. Thus, problem (41) can be rewritten as follows:

$$\overline{\tilde{Z}}^U_\alpha = \max_{\substack{(\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha,\, where \\ v = c_{1i},\ldots,c_{ni}, d_{1i},\ldots,d_{mi}, \\ c'_{1j},\ldots,c'_{nj}, d'_{1j},\ldots,d'_{mj}, \\ a_{ij}, b_{ij}, e_{ij}, s_{ij}, k_{i1}, k_{j2},\ \text{for all } i,j.}} \begin{cases} \min\limits_{x\in X} \sum\limits_{i=1}^{s} w_i(c_i.x) + \sum\limits_{i=1}^{s} w_i(d_i.y), \\ \quad s.t.\ \ A_1x + B_1y \le k_1, \\ \qquad \sum\limits_{i=1}^{s} w_i = 1, w_i \ge 0, for\ i = 1,\ldots,s, \\ \qquad \min\limits_{y\in Y} \sum\limits_{i=1}^{t} w'_i(c'_i.x) + \sum\limits_{i=1}^{t} w'_i(d'_i.y), \\ \qquad s.t.\ \ A_2x + B_2y \le k_2, \\ \qquad \sum\limits_{i=1}^{t} w'_i = 1, w'_i \ge 0, for\ i = 1,\ldots,t. \end{cases} \tag{61}$$

Since model (61) finds the maximum of all minimum objective values, we should consider the objective function of leader in the worst case, that is, we set c_i and d_i to their upper bound, in other words, $(\tilde{c}_i)^U_\alpha$ and $(\tilde{d}_i)^U_\alpha$, respectively.

$$\overline{\tilde{Z}}^U_\alpha = \max_{\substack{(\tilde{v})^L_\alpha \le v \le (\tilde{v})^U_\alpha,\, where \\ v = c'_{1j},\ldots,c'_{nj}, d'_{1j},\ldots,d'_{mj}, \\ a_{ij}, b_{ij}, e_{ij}, s_{ij}, k_{i1}, k_{j2},\ \text{for all } i,j.}} \begin{cases} \min\limits_{x\in X} \sum\limits_{i=1}^{s} w_i((\tilde{c}_i)^U_\alpha.x) + \sum\limits_{i=1}^{s} w_i((\tilde{d}_i)^U_\alpha.y), \\ \quad s.t.\ \ A_1x + B_1y \le k_1, \\ \qquad \sum\limits_{i=1}^{s} w_i = 1, w_i \ge 0, for\ i = 1,\ldots,s, \\ \qquad \min\limits_{y\in Y} \sum\limits_{i=1}^{t} w'_i(c'_i.x) + \sum\limits_{i=1}^{t} w'_i(d'_i.y), \\ \qquad s.t.\ \ A_2x + B_2y \le k_2, \\ \qquad \sum\limits_{i=1}^{t} w'_i = 1, w'_i \ge 0, for\ i = 1,\ldots,t. \end{cases} \tag{62}$$

To obtain an upper bound (or a maximum) for the minimum objective value, we should consider the values of components of the matrices A_1, B_1, A_2, B_2, k_1, and k_2 in $(\tilde{A}_1)^L_\alpha, (\tilde{B}_1)^L_\alpha, (\tilde{A}_2)^L_\alpha, (\tilde{B}_2)^L_\alpha, (\tilde{k}_1)^U_\alpha$, and $(\tilde{k}_2)^U_\alpha$, respectively, because we can find the smallest feasible region, in this case. Hence, the upper bound is obtained for the objective function. Doing this work, problem (19) is converted to the following problem.

$$\overline{\tilde{Z}}_{\alpha}^{U} = \max_{\substack{(\tilde{v})_{\alpha}^{L} \leq v \leq (\tilde{v})_{\alpha}^{U}, where \\ v = c'_{1j}, \ldots, c'_{nj}, d'_{1j}, \ldots, d'_{mj}, \\ \text{for all } j.}} \begin{cases} \min_{x \in X} \sum_{i=1}^{s} w_i((\tilde{c}_i)_{\alpha}^{U}.x) + \sum_{i=1}^{s} w_i((\tilde{d}_i)_{\alpha}^{U}.y), \\ \quad s.t. \ \ (\tilde{A}_1)_{\alpha}^{L}.x + (\tilde{B}_1)_{\alpha}^{L}.y \leq (\tilde{k}_1)_{\alpha}^{U}, \\ \quad \sum_{i=1}^{s} w_i = 1, w_i \geq 0, for \ i = 1, \ldots, s, \\ \quad \min_{y \in Y} \sum_{i=1}^{t} w'_i(c'_i.x) + \sum_{i=1}^{t} w'_i(d'_i.y), \\ \quad s.t. \ \ (\tilde{A}_2)_{\alpha}^{L}.x + (\tilde{B}_2)_{\alpha}^{L}.y \leq (\tilde{k}_2)_{\alpha}^{U}, \\ \quad \sum_{i=1}^{t} w'_i = 1, w'_i \geq 0, for \ i = 1, \ldots, t. \end{cases} \tag{63}$$

We apply Theorem 2 for problem (3) and rewrite the problem as follows:

$$\overline{\tilde{Z}}_{\alpha}^{U} = \max_{\substack{(\tilde{v})_{\alpha}^{L} \leq v \leq (\tilde{v})_{\alpha}^{U}, \\ where \ v = c'_{1j}, \ldots, c'_{nj}, \\ d'_{1j}, \ldots, d'_{mj}, \quad \text{for all } j.}} \begin{cases} \min \sum_{i=1}^{s} w_i((\tilde{c}_i)_{\alpha}^{U}.x) + \sum_{i=1}^{s} w_i((\tilde{d}_i)_{\alpha}^{U}.y), \\ \quad s.t. \ \ (\tilde{A}_1)_{\alpha}^{L}.x + (\tilde{B}_1)_{\alpha}^{L}.y \leq (\tilde{k}_1)_{\alpha}^{U}, \\ \quad \sum_{i=1}^{t} w'_i(d'_i.y) - ((\tilde{k}_2)_{\alpha}^{U} - (\tilde{A}_2)_{\alpha}^{L}.x)^T.u = 0, \\ \quad ((\tilde{B}_2)_{\alpha}^{L})^T.u \geq \sum_{i=1}^{t} w'_i.d'_i, \\ \quad \sum_{i=1}^{s} w_i = 1, \sum_{i=1}^{t} w'_i = 1, \\ \quad w_i, w'_j \geq 0, for \ i = 1, \ldots, s, \ j = 1, \ldots, t, \\ \quad\quad x, y, u \geq o. \end{cases} \tag{64}$$

Let $U = \{u \geq o \mid ((\tilde{B}_2)_{\alpha}^{L})^T.u \geq \sum_{i=1}^{t} w'_i.(\tilde{d}'_i)_{\alpha}^{U}\}$. Suppose that the vertex points of set U are denoted by a set $U^V = \{u^1, \ldots, u^{r'}\}$. Hence, we should solve at most r' linear programming problems as follows:

$$LP(u^k): \overline{\tilde{Z}}^{U}_{\alpha,k} = \min \sum_{i=1}^{s} w_i((\tilde{c}_i)^U_\alpha .x) + \sum_{i=1}^{s} w_i((\tilde{d}_i)^U_\alpha .y), \tag{65}$$

$$s.t.\ \ (\tilde{A}_1)^L_\alpha .x + (\tilde{B}_1)^L_\alpha .y \leq (\tilde{k}_1)^U_\alpha, \tag{66}$$

$$\sum_{i=1}^{t} w'_i(\tilde{d}'_i)^L_\alpha .y \leq ((\tilde{k}_2)^U_\alpha - (\tilde{A}_2)^L_\alpha .x)^T .u^k \leq \sum_{i=1}^{t} w'_i(\tilde{d}'_i)^U_\alpha .y, \tag{67}$$

$$\sum_{i=1}^{s} w_i = 1, \sum_{i=1}^{t} w'_i = 1, w_i, w'_j \geq 0, for\ i = 1,\dots,s,\ = 1,\dots,t,$$

$$x, y \geq o, \tag{68}$$

where $k = 1,\dots,r'$. Finding the optimal values of variables x and y for each $\alpha \in [0,1]$, we can estimate values $(\tilde{c}_i)^U_\alpha .x + (\tilde{d}_i)^U_\alpha .y$ for each $\alpha \in [0,1]$ and $i = 1,\dots,s$. Hence, we can obtain $\tilde{Z}^U_\alpha$. Having $\tilde{Z}^L_\alpha$ and $\tilde{Z}^U_\alpha$, we can easily compute the compromise vector $\tilde{Z}$. We will illustrate the process of obtaining $\tilde{Z}^L_\alpha$, $\tilde{Z}^U_\alpha$, and $\tilde{Z}$ by a numerical example, in Section 5.

Remark 1. As it was considered the formulation of FMOLBP in Subsection 3.1 and the solution procedure for problem (29)–(33) in Subsection 3.2, the fuzzy numbers can be arbitrary according to Definition 1(a) and there is no restriction on the fuzzy number type. The solution procedure does not depend on the form or type of fuzzy numbers.

In the next section, we design an efficient algorithm to solve the problem of FMOLBP using the points in this section.

4 An algorithm for resolution of the problem of FMOLBP as (29)–(33)

An efficient algorithm is designed to solve problem (29)–(33). This algorithm produces the fuzzy compromise objective values of the problem as fuzzy compromise objective values in the leader and follower and fuzzy optimal weighted objective value. The step-by-step algorithm is as follows:

Step 1. Determine the weight vectors $W = (w_1,\dots,w_s)$ and $W' = (w'_1,\dots,w'_t)$ such that $\sum_{i=1}^{s} w_i = 1$, $\sum_{i=1}^{t} w'_i = 1$, and $w_i \geq 0, w'_j \geq 0$, for $i = 1,\dots,s,\ j = 1,\dots,t$.

Step 2. Determine the fuzzy parameters $\tilde{v} = \{(v, \mu_{\tilde{v}}(v)) \mid v \in S(\tilde{v})\}$ of problem (29)–(33), where $\tilde{v}$ is one of the parameters $\tilde{c}_{1i},\dots,\tilde{c}_{ni}$, $\tilde{c}'_{1j},\dots,\tilde{c}'_{nj}$, $\tilde{d}_{1i},\dots,\tilde{d}_{mi}$, $\tilde{d}'_{1j},\dots,\tilde{d}'_{mj}$, $i = 1,\dots,s, j = 1,\dots,t$, $\tilde{a}_{ij}$, $i = 1,\dots,p, j = 1,\dots,n$, $\tilde{b}_{ij}$, $i = 1,\dots,p, j = 1,\dots,m$, $\tilde{e}_{ij}$, $i = 1,\dots,q, j = 1,\dots,n$, $\tilde{s}_{ij}$,

$i = 1, \ldots, q, j = 1, \ldots, m$, $\tilde{k}_{i1}$, $\tilde{k}_{j2}$, $i = 1, \ldots, p, j = 1, \ldots, q$, and $S(\tilde{v})$ is the support of $\tilde{v}$.

Step 3. Create α-cuts of the fuzzy parameters $\tilde{v}$ as follows:

$$(\tilde{v})_\alpha = [(\tilde{v})^L_\alpha, (\tilde{v})^U_\alpha] = [\min_v\{v \in S(\tilde{v}) \mid \mu_{\tilde{v}}(v) \geq \alpha\},$$
$$\max_v\{v \in S(\tilde{v}) \mid \mu_{\tilde{v}}(v) \geq \alpha\}].$$

Step 4. Create the following system:

$$\begin{cases} ((\tilde{B}_2)^U_\alpha)^T.u \geq \sum_{i=1}^t w'_i.(\tilde{d}'_i)^U_\alpha, \\ u \geq o. \end{cases}$$

Step 5. Select some discrete points for α from $[0,1]$ as $\alpha_t = 0.1 \times t$ for $t = 0, 1, \ldots, 10$ or $\alpha_t = 0.25 \times t$ for $t = 0, 1, 2, 3, 4$. The selection depends on the decision-maker or the required accuracy (it can be selected one of them or another partition from $[0,1]$ for choosing the values of α).

Step 6. Compute the vertex points of the system of Step 4 for values α_t of Step 5. Denote the vertex points by $U^v_{\alpha_t} = \{u^{1,\alpha_t}, \ldots, u^{r,\alpha_t}\}$. If the system of Step 4 is empty for each $\alpha \in [0,1]$, then problem (29)–(33) has no compromise solutions. The feasible solution set is empty. End.

Step 7. Create the linear programming problem (57)–(60), for $U^v_{\alpha_t} = \{u^{1,\alpha_t}, \ldots, u^{r,\alpha_t}\}$ and value α_t. Solve r linear programming problems and select the minimum objective value among them.

Step 8. Find the lower bound of α-cut of the fuzzy compromise objective values of leader and follower or the fuzzy optimal weighted objective value using Step 7, for values α_t's of Step 5.

Step 9. Create the following system:

$$\begin{cases} ((\tilde{B}_2)^L_\alpha)^T.u \geq \sum_{i=1}^t w'_i.(\tilde{d}'_i)^U_\alpha, \\ u \geq o. \end{cases}$$

Step 10. Compute the vertex points of the system of Step 9 for values α_t of Step 5. Denote the vertex points by $(U')^v_{\alpha_t} = \{(u')^{1,\alpha_t}, \ldots, (u')^{r',\alpha_t}\}$. If the system of Step 9 is empty for each $\alpha \in [0,1]$, then problem (29)–(33) has no compromise solutions. The feasible solution set is empty. End.

Step 11. Create the linear programming (65)–(68), for $(U')^v_{\alpha_t} = \{(u')^{1,\alpha_t}, \ldots, (u')^{r',\alpha_t}\}$ and value α_t. Solve r' linear programming problems and select the maximum objective value among them.

Step 12. Find the upper bound of α-cut of the fuzzy compromise objective values of leader and follower or the fuzzy optimal weighted objective value using Step 11, for values α_t's of Step 5.

Step 13. Create the fuzzy compromise objective values of leader and follower

or the optimal weighted objective value by two Steps 8 and 12.
Step 14. End.

Some numerical examples are presented to illustrate the algorithm and compare it with other existing approaches, in Sections 5 and 6. The comparison shows the efficiency of the proposed algorithm.

5 Numerical example

In this section, we illustrate the proposed algorithm in Section 5 to solve FMOLBP by a numerical example.

Example 1. Consider the following problem:

$$\begin{aligned}
\tilde{Z} = &\min_{x\in X} \tilde{F}(x,y) = (\tilde{c}_1.x + \tilde{d}_1.y, \tilde{c}_2.x + \tilde{d}_2.y)^T,\\
&s.t. \quad \tilde{A}_1.x + \tilde{B}_1.y \le \tilde{k}_1,\\
&\qquad \min_{y\in Y} \tilde{f}(x,y) = (\tilde{c}'_1.x + \tilde{d}'_1.y, \tilde{c}'_2.x + \tilde{d}'_2.y)^T,\\
&\qquad\quad s.t. \quad \tilde{A}_2.x + \tilde{B}_2.y \le \tilde{k}_2,
\end{aligned}$$

where $\tilde{c}_1 = [\tilde{c}_{11}] = [(1,2,3)]$, $\tilde{d}_1 = [\tilde{d}_{11}] = [(-1,1,2)]$, $\tilde{c}_2 = [\tilde{c}_{12}] = [(2,3,5)]$, $\tilde{d}_2 = [\tilde{d}_{12}] = [(4,5,6)]$, $\tilde{c}'_1 = [\tilde{c}'_{11}] = [(2,3,4)]$, $\tilde{d}'_1 = [\tilde{d}'_{11}] = [(3,4,5)]$, $\tilde{c}'_2 = [\tilde{c}'_{12}] = [(-1,0,1)]$, $\tilde{d}'_2 = [\tilde{d}'_{12}] = [(1.5,2.3,3.5)]$, $\tilde{A}_1 = \begin{pmatrix}\tilde{a}_{11}\\ \tilde{a}_{21}\end{pmatrix} = \begin{pmatrix}(-3,-2,-1)\\ (-2,-1,0)\end{pmatrix}$, $\tilde{B}_1 = \begin{pmatrix}\tilde{b}_{11}\\ \tilde{b}_{21}\end{pmatrix} = \begin{pmatrix}(-1,0,1)\\ (1.5,2.5,3.5)\end{pmatrix}$, $\tilde{k}_1 = \begin{pmatrix}\tilde{k}_{11}\\ \tilde{k}_{21}\end{pmatrix} = \begin{pmatrix}(5,6,7)\\ (6,7,9)\end{pmatrix}$, $\tilde{A}_2 = [\tilde{e}_{11}] = [(1,1,1)]$, $\tilde{B}_2 = [\tilde{s}_{11}] = [(2.2,3.4,4.6)]$, and $\tilde{k}_2 = [\tilde{k}_{12}] = [(5,6,8)]$.
Step 1. The weight vectors are $w_1 = w_2 = w'_1 = w'_2 = 0.5$.
Step 2. The fuzzy parameters $\tilde{v} = \{(v, \mu_{\tilde{v}}(v)) \mid v \in S(\tilde{v})\}$ of problem (29)–(33) for this example are given above.
Step 3. The α-cuts of the fuzzy parameters of $\tilde{v} = (v_1, v_2, v_3)$ can be computed the following formulas: $(\tilde{v})_\alpha = [(\tilde{v})^L_\alpha, (\tilde{v})^U_\alpha]$, where $(\tilde{v})^L_\alpha = v_1 + (v_2 - v_1)\alpha$ and $(\tilde{v})^U_\alpha = v_3 - (v_3 - v_2)\alpha$.
Step 4. The system corresponding to this step for this example is
$U = \{u \ge o \mid (4.6 - 1.2\alpha).u \ge w'_1(5-\alpha) + w'_2(3.5 - 1.2\alpha)\}$.
Step 5. The selected values for α are as 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.
Step 6. The vertex points of set U^v_α are easily obtained for each $\alpha = 0, 0.1, \ldots, 1$ as $u_\alpha = \frac{0.5(5-\alpha)+0.5(3.5-1.2\alpha)}{(4.6-1.2\alpha)}$. Moreover U^v_α is singleton for each α.
Step 7. The linear programming problem (57)–(60), for U^v_α, and the value α is created and the problem related to $\overline{\tilde{Z}}^L_{\alpha,k}$ is as follows:

$$\begin{aligned} LP(u^k):&\overline{\tilde{Z}}^L_{\alpha,k} = \min(w_1(1+\alpha)+w_2(2+\alpha))x+(w_1(-1+2\alpha)+w_2(4+\alpha))y,\\ &s.t.\quad \begin{pmatrix}(-1-\alpha)\\(-\alpha)\end{pmatrix}.x+\begin{pmatrix}(-\alpha+1)\\(-\alpha+3.5)\end{pmatrix}.y\leq\begin{pmatrix}(5+\alpha)\\(6+\alpha)\end{pmatrix},\\ &(w'_1(3+\alpha)+w'_2(1.5+0.8\alpha))y\leq((5+\alpha)-x)u_\alpha\\ &\qquad\leq(w'_1(5-\alpha)+w'_2(3.5-1.2\alpha))y,\\ &\qquad x,y\geq o.\end{aligned}$$

Step 8. The lower bound of α-cut of the fuzzy compromise objective values of leader using Step 7, for values α_t's of Step 5, is solved. Its results display in Table 1.

Step 9. The system corresponding to this step for this example is $U'=\{u\geq o \mid (2.2+1.2\alpha).u\geq w'_1(5-\alpha)+w'_2(3.5-1.2\alpha)\}$.

Step 10. The vertex points of set $(U')^v_\alpha$ are easily obtained for each $\alpha = 0, 0.1,\ldots,1$ as $u'_\alpha = \frac{0.5(5-\alpha)+0.5(3.5-1.2\alpha)}{(2.2+1.2\alpha)}$. Moreover $(U')^v_\alpha$ is singleton for each α.

Step 11. The linear programming problem (65)–(68), for $(U')^v_\alpha$, and the value α is created, and the problem related to $\overline{\tilde{Z}}^U_{\alpha,k}$ is as follows:

$$\begin{aligned} &LP(u^k):\\ &\quad\overline{\tilde{Z}}^U_{\alpha,k} = \min(w_1.(3-\alpha)+w_2.(5-2\alpha)).x+(w_1.(2-\alpha)+w_2.(6-\alpha)).y,\\ &\quad s.t.\quad \begin{pmatrix}(-3+\alpha)\\(-2+\alpha)\end{pmatrix}.x+\begin{pmatrix}(-1+\alpha)\\(1.5+\alpha)\end{pmatrix}.y\leq\begin{pmatrix}(7-\alpha)\\(9-2\alpha)\end{pmatrix},\\ &\quad(w'_1(3+\alpha)+w'_2(1.5+0.8\alpha))y\leq((8-2\alpha)-x).u'_\alpha\\ &\qquad\leq(w'_1(5-\alpha)+w'_2(3.5-1.2\alpha))y,\\ &\qquad x,y\geq o.\end{aligned}$$

Step 12. The upper bound of α-cut of the fuzzy compromise objective values of leader using Step 11, for values α_t's of Step 5, is solved. Its results display in Table 1.

Step 13. The fuzzy compromise objective values of leader by two Steps 8 and 12 are created in Table 1. End.

Figure 1 displays the fuzzy objective values of leader using the data of Table 1. Using the data of columns 1, 2, and 3, the fuzzy objective values of follower can be easily computed.

In the next section, the fuzzy compromise objectives of leader and follower, for an example from [3] as a benchmark, are computed and compared with other existing methods. It shows the efficiency of the proposed algorithm in this paper.

Table 1: Estimation of compromise vector $\tilde{Z} = \left(\tilde{Z}_1, \tilde{Z}_2\right)$ using $\left(\tilde{Z}_i\right)^L_\alpha$ and $\left(\tilde{Z}_i\right)^U_\alpha$, for $i = 1, 2$, with different α-level values.

α	(x^*, y^*) **optimal of problem related to** $\overline{\overline{Z}}^L_\alpha$	(x^*, y^*) **optimal of problem related to** $\overline{\overline{Z}}^U_\alpha$	$\tilde{Z}_1$		$\tilde{Z}_2$	
			$\left(\tilde{Z}_1\right)^L_\alpha = (\tilde{c}_1)^L_\alpha x + \left(\tilde{d}_1\right)^L_\alpha y$	$\left(\tilde{Z}_1\right)^U_\alpha = (\tilde{c}_1)^U_\alpha x + \left(\tilde{d}_1\right)^U_\alpha y$	$\left(\tilde{Z}_2\right)^L_\alpha = (\tilde{c}_2)^L_\alpha x + \left(\tilde{d}_2\right)^L_\alpha y$	$\left(\tilde{Z}_2\right)^U_\alpha = (\tilde{c}_2)^U_\alpha x + \left(\tilde{d}_2\right)^U_\alpha y$
0	(0,1.087)	(0,3.636)	-1.087	7.272	4.348	21.816
0.1	(0,1.138)	(0,3.362)	-0.9104	6.3878	4.6658	19.8358
0.2	(0,1.193)	(0,3.115)	-0.7158	5.607	5.0106	18.067
0.3	(0,1.25)	(0,2.891)	-0.5	4.9147	5.375	16.4787
0.4	(0,1.311)	(0,2.687)	-0.2622	4.2992	5.7684	15.0472
0.5	(0,1.375)	(0,2.5)	0	3.75	6.1875	13.75
0.6	(0,1.443)	(0,2.329)	0.2886	3.2606	6.6378	12.5766
0.7	(0,1.516)	(0,2.171)	0.6064	2.8223	7.1252	11.5063
0.8	(0,1.593)	(0,2.025)	0.9558	2.43	7.6464	10.53
0.9	(0,1.676)	(0,1.89)	1.3408	2.079	8.2124	9.639
1.0	(0,1.765)	(0,1.765)	1.765	1.765	8.825	8.825

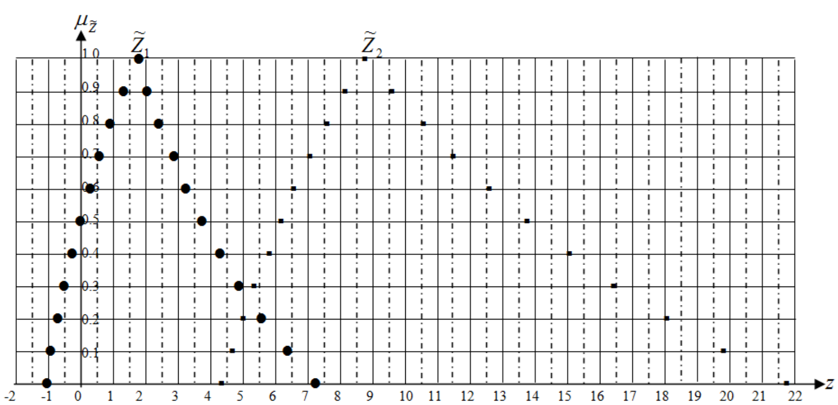


Figure 1: Membership function of compromise vector $\tilde{Z} = \left(\tilde{Z}_1, \tilde{Z}_2\right)$, where $\tilde{Z}_1$ and $\tilde{Z}_2$ are computed by $\left(\left(\tilde{Z}_1\right)^L_\alpha, \left(\tilde{Z}_1\right)^U_\alpha\right)$ and $\left(\left(\tilde{Z}_2\right)^L_\alpha, \left(\tilde{Z}_2\right)^U_\alpha\right)$, respectively.

6 Comparison with other existing methods

In this section, we present an example from [3, 4] as a benchmark to show the efficiency of the proposed method in this paper with the existing methods.

Example 2. Consider the following problem:

$$\begin{aligned}
&\min_{x_1,x_2} \begin{pmatrix} \tilde{f}_{11}(x) = (x_1 + \tilde{3}x_2 + \tilde{2}x_3 + \tilde{3}x_4) \\ \tilde{f}_{12}(x) = (\tilde{2}x_1 + \tilde{9}x_2 + \tilde{3}x_3 + \tilde{5}x_4) \\ \tilde{f}_{13}(x) = (\tilde{3}x_1 + \tilde{9}x_2 + \tilde{9}x_3 + x_4) \end{pmatrix} \\
&s.t. \quad \min_{x_3,x_4} \begin{pmatrix} \tilde{f}_{21}(x) = (\tilde{6}x_1 + \tilde{3}x_2 + \tilde{2}x_3 + \tilde{2}x_4) \\ \tilde{f}_{22}(x) = (\tilde{5}x_1 + \tilde{9}x_2 - \tilde{9}x_3 + \tilde{6}x_4) \end{pmatrix} \\
&\qquad s.t. \quad \tilde{3}x_1 - x_2 + x_3 + \tilde{3}x_4 \le \tilde{48}, \\
&\qquad\qquad \tilde{2}x_1 + \tilde{4}x_2 + \tilde{2}x_3 - \tilde{2}x_4 \le \tilde{35}, \\
&\qquad\qquad x_1 + \tilde{2}x_2 - x_3 + x_4 \le \tilde{30}, \\
&\qquad\qquad x_1, x_2, x_3, x_4 \ge 0.
\end{aligned}$$

The fuzzy numbers in [3, 4] are assumed to be triangular fuzzy numbers and given as follows:
$\tilde{2} = (0,2,3), \tilde{3} = (2,3,4), \tilde{4} = (3,4,5), \tilde{5} = (4,5,6), \tilde{6} = (5,6,7)$, $\tilde{8} = (6,8,10), \tilde{9} = (8,9,10), \tilde{30} = (28,30,32), \tilde{35} = (33,35,37)$, and $\tilde{48} = (45,48,49)$.
The problems related to $\overline{\tilde{Z}}_\alpha^L$ and $\overline{\tilde{Z}}_\alpha^U$ are as follows:

$$\begin{aligned}
LP(u^k) : \overline{\tilde{Z}}_{\alpha,k}^L = &\min \frac{1}{3}(1 + (2\alpha) + (2+\alpha))x_1 + \frac{1}{3}((2+\alpha) + (8+\alpha) + (8+\alpha))x_2 \\
&+ \frac{1}{3}((2\alpha) + (2+\alpha) + (8+\alpha))x_3 + \frac{1}{3}((2+\alpha) + (4+\alpha) + 1)x_4, \\
&s.t.\ 0.5((2\alpha) + (-8-\alpha))x_3 + 0.5((2\alpha) + (5+\alpha))x_4 \\
&\qquad \le ((45+3\alpha) - (4-\alpha)x_1 + x_2)u_1^k \\
&\qquad\quad + ((33+2\alpha) - (3-\alpha)x_1 - (5-\alpha)x_2)u_2^k \\
&\qquad\quad + ((-32+2\alpha) + x_1 - (-2\alpha)x_2)u_3^k \\
&\qquad \le 0.5((3-\alpha) + (-8-\alpha))x_3 + 0.5((3-\alpha) + (7-\alpha))x_4, \\
&\qquad\qquad x_1, x_2, x_3, x_4 \ge 0,
\end{aligned}$$

where u^k's are the vertex points of the set
$U = \{u = (u_1, u_2, u_3) \ge o \mid u_1 + (3-\alpha)u_2 + u_3 \ge -2.5 - \alpha, (4-\alpha)u_1 + (-2\alpha)u_2 - u_3 \ge 5 - \alpha\}$ and

$$
\begin{aligned}
LP(u^k): \overline{\tilde{Z}}^{U}_{\alpha,k} = \min \frac{1}{3}&(1 + (3-\alpha) + (4-\alpha))x_1 \\
&+ \frac{1}{3}((4-\alpha) + (10-\alpha) + (10-\alpha))x_2 \\
&+ \frac{1}{3}((3-\alpha) + (4-\alpha) + (10-\alpha))x_3 \\
&+ \frac{1}{3}((4-\alpha) + (6-\alpha) + 1)x_4, \\
s.t.\ 0.5((-10&+3\alpha)x_3 + (5+3\alpha)x_4) \\
&\leq ((49-\alpha) - (2+\alpha)x_1 + x_2)u_1^k \\
&\quad + ((37-2\alpha) - 2\alpha.x_1 - (3+\alpha)x_2)u_2^k \\
&\quad + ((-28-2\alpha) + x_1 + (3-\alpha)x_2) \\
&\leq 0.5((-5-2\alpha)x_3 + (10-2\alpha)x_4), \\
&x_1, x_2, x_3, x_4 \geq 0,
\end{aligned}
$$

where u^k's are the vertex points of the following set:
$U = \{u = (u_1, u_2, u_3) \geq o \mid u_1 + (2\alpha)u_2 + u_3 \geq -2.5 - \alpha, (2+\alpha)u_1 + (-3+\alpha)u_2 - u_3 \geq 5 - \alpha\}$.
In the above models, $\alpha \in [0,1]$, $w_1 = w_2 = w_3 = \frac{1}{3}$, and $w'_1 = w'_2 = 0.5$. The compromise values $\tilde{f}_{11}, \tilde{f}_{12}, \tilde{f}_{13}, \tilde{f}_{21}$, and $\tilde{f}_{22}$ are computed using the values of $(\tilde{f}_{11})^L_\alpha, (\tilde{f}_{11})^U_\alpha, (\tilde{f}_{12})^L_\alpha, (\tilde{f}_{12})^U_\alpha, (\tilde{f}_{13})^L_\alpha, (\tilde{f}_{13})^U_\alpha$, $(\tilde{f}_{21})^L_\alpha, (\tilde{f}_{21})^U_\alpha, (\tilde{f}_{22})^L_\alpha, (\tilde{f}_{22})^U_\alpha$, and x_i^* obtained from solving two models $LP(u^k)$. The optimal solutions of problems related to $\bar{\tilde{Z}}^L_\alpha$ and $\bar{\tilde{Z}}^U_\alpha$ for different values α are presented in Table 2. The values of leader and follower objectives for different values α are showed in Tables 3 and 4, respectively.

Table 2: Optimal solutions of problems related to $\bar{\tilde{Z}}^L_\alpha$ and $\bar{\tilde{Z}}^U_\alpha$ for different values α.

α	x^* **optimal of problem related to** $\bar{\tilde{Z}}^L_\alpha$	x^* **optimal of problem related to** $\bar{\tilde{Z}}^U_\alpha$
0	(11.25,0,0,0)	(24.5,0,0,0)
0.25	(12.2,0,0,0)	(21.66667,0,0,0)
0.5	(13.28571,0,0,0)	(19.4,0,0,0)
0.75	(14.53846,0,0,0)	(17.54545,0,0,0)
1	(16,0,0,0)	(16,0,0,0)

A comparison among the proposed method and the methods in [3, 4] is presented in Table 5.

As it is shown in Table 5, the results obtained from the proposed method in terms of fuzzy numbers are more accurate than methods of finding a solution with the highest membership degree. In the proposed method, the decision-maker can consider all of the solutions in all the membership de-

Table 3: Values of leader objectives for different values α.

α	$\tilde{f}_{11}$		$\tilde{f}_{12}$		$\tilde{f}_{13}$	
	$\left(\tilde{f}_{11}\right)_{\alpha}^{L}$	$\left(\tilde{f}_{11}\right)_{\alpha}^{U}$	$\left(\tilde{f}_{12}\right)_{\alpha}^{L}$	$\left(\tilde{f}_{12}\right)_{\alpha}^{U}$	$\left(\tilde{f}_{13}\right)_{\alpha}^{L}$	$\left(\tilde{f}_{13}\right)_{\alpha}^{U}$
0	11.25	24.5	0	73.5	22.5	98
0.25	12.2	21.66667	6.1	59.58334	27.45	81.25001
0.5	13.28571	19.4	13.28571	48.5	33.21428	67.9
0.75	14.53846	17.54545	21.80769	39.47726	39.98077	57.02271
1	16	16	32	32	48	48

Table 4: Values of follower objectives for different values α.

α	$\tilde{f}_{21}$		$\tilde{f}_{22}$	
	$\left(\tilde{f}_{21}\right)_{\alpha}^{L}$	$\left(\tilde{f}_{21}\right)_{\alpha}^{U}$	$\left(\tilde{f}_{22}\right)_{\alpha}^{L}$	$\left(\tilde{f}_{22}\right)_{\alpha}^{U}$
0	56.25	171.5	45	147
0.25	64.05	146.25	51.85	124.5834
0.5	73.07141	126.1	59.7857	106.7
0.75	83.59615	109.6591	69.05769	92.11361
1	96	96	80	80

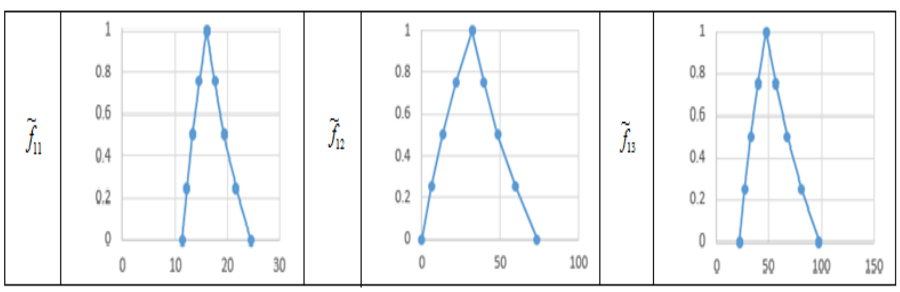


Figure 2: Fuzzy compromise objective values of leaders by the proposed approach

Table 5: Comparison among the proposed method and two other methods.

The method of Baky Eid, and El Sayed, in [3]		**The method of Pramanik and Dey in [4]**		**The proposed method**
$f_{11} = 29$	$\mu_{11} = 1$	$f_{11} = 37$	$\mu_{11} = 0.902$	$\tilde{f}_{11}$ in Figure 2
$f_{12} = 70.9483$	$\mu_{12} = 0.917$	$f_{12} = 90$	$\mu_{12} = 0.815$	$\tilde{f}_{12}$ in Figure 2
$f_{13} = 88.2743$	$\mu_{13} = 0.821$	$f_{13} = 108.25$	$\mu_{13} = 0.692$	$\tilde{f}_{13}$ in Figure 2
$f_{21} = 80.912$	$\mu_{21} = 0.527$	$f_{21} = 78.25$	$\mu_{21} = 0.496$	$\tilde{f}_{21}$ in Figure 3
$f_{22} = 111.27$	$\mu_{22} = 0.81$	$f_{22} = 105.5$	$\mu_{22} = 0.795$	$\tilde{f}_{22}$ in Figure 3

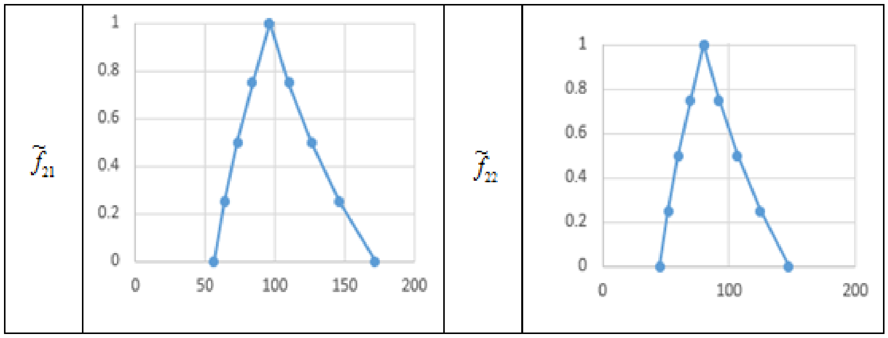


Figure 3: Fuzzy compromise objective values of followers by the proposed approach

grees. Moreover, the method of Pramanik and Dey in [4] did not follow the basic concepts of bilevel programming. Also, the obtained compromise solutions improve the compromise solutions in [3] for degrees 1, 0.917, 0.821, and 0.81, considerably. Moreover, the obtained compromise solution covers the compromise solution in [3] for degree 0.578.

7 Conclusion

This paper showed how Zadeh's extension principle can be efficiently applied to solve the FMOLBP problem. Using the structure of bilevel programming and concept of extension principle, two crisp multiobjective linear three-level programming problems were designed to compute the lower and upper bound of the α-level of the compromise objective value. Due to the strongly NP-hardness and high computational complexity of the problem, there is no efficient algorithm to solve it. The developed methods cannot completely solve the problem. To overcome the deficiencies, a pair of crisp multiobjective linear three-level programming problems was proposed to construct the compromise fuzzy objective functions of leader and follower. The main advantages of this method are access to fuzzy compromise objectives of leader and follower with less computational complexity with respect to approaches based on the rejection of solution and resolve model successively until finding a satisfactory decision or based on FGP. The proposed method is based on the weighting method, dual theory, and vertex enumeration approaches that preserve the linearity of the model. A comparison among the proposed method and the methods of Pramanik and Dey [4] and Baky Eid, and El Sayed, [3] showed the efficiency of our method.

Acknowledgements

The author is very grateful to the editor-in-chief and anonymous referees for their comments and suggestions, which have been very helpful in improving the paper.

References

1. Abdelaziz, F.B. and Mejri, S. *Multiobjective bi-level programming for shared inventory with emergency and backorders*, Ann. Oper. Res., 267, (2018) 47–63.

2. Abo-Sinna, M.A. and Baky, I.A. *Interactive balance space approach for solving bilevel multiobjective programming problems*, Adv. Model. Anal. B 49 (3-4) (2006) 43–62.

3. Audet, C., Haddad, J. and Savard, G. *A note on the definition of a linear bilevel programming solution*, Appl. Math. Comput. 181 (2006) 351–355.

4. Baky, I.A. *Fuzzy goal programming algorithm for solving decentralized bilevel multiobjective programming problems*, Fuzzy Set. Syst. 160(18) (2009) 2701–2713.

5. Baky, I.A. *Solving multilevel multiobjective linear programming problems through fuzzy goal programming approach*, Appl. Math. Model. 34(9) (2010) 2377–2387.

6. Baky, I.A., Eid, M.H. and El Sayed, M.A. *Bi-level multiobjective programming problem with fuzzy demands: A fuzzy goal programming algorithm*, OPSEARCH. 51, (2014) 280–296.

7. Bard, J. *Practical bilevel optimization: Algorithms and applications*, Amsterdam, Kluwer Academic Publishers, 1998.

8. Bard, J. and Falk, J. *An explicit solution to the programming problem*, Comput. Oper. Res. 9 (1982) 77–100.

9. Bard, J.F. and Moore, J.T. *A branch and bound algorithm for the bilevel programming problem*, SIAM J. Sci. Stat. Comput. 11 (1990) 281–292.

10. Bazaraa, M.S., Jarvis, J.J. and Sherali, H.D. *Linear programming and network flows, Second edition*, Wiley, New York, 1990.

11. Bellman, R.E. and Zadeh, L.A. *Decision-making in a fuzzy environment*, Manag. Sci. 17 (1970) 141–164.

12. Bialas, W.F. and Karwan, M.H. *Two-level linear programming*, Manag. Sci. 30(8) (1984) 1004–1020.

13. Budnitzkia, A. *The solution approach to linear fuzzy bilevel optimization problems*, Optimization 64(5) (2015) 1195–1209.

14. Candler, W. and Townsley, R. *A linear two-level programming problem*, Comput. Oper. Res. 9 (1982) 59–76.

15. Colson, B. and Marcotte, P. and Savard, G. *An overview of bilevel optimization*, Ann. Oper. Res. 153 (2007) 235–256.

16. Gang, J., Tu, Y., Lev, B., Xu, J., Shen, W. and Yao, L. *A Multi-objective bi-level location planning problem for stone industrial parks*, Comput. Oper. Res. 56 (2015) 8–21.

17. Hansen, P., Jaumard, B. and Savard, G. *New branch-and-bound rules for linear bilevel programming*, SIAM J. Sci. Statist. Comput. 13 (1992) 1194–1217.

18. Kamal, M., Gupta, S., Chatterjee, P., Pamucar, D. and Stevic, Z. *Bi-level multi-objective production planning problem with multi-choice parameters: A fuzzy Goal programming algorithm*, Algorithms 12(7) (2019) 143.

19. Liu, Q.-M. and Yang, Y.-M. *Interactive programming approach for solving multilevel multiobjective linear programming problem*, J. Intell. Fuzzy Syst. 35(1) (2018) 55–61.

20. Lu, J., Shi, C. and Zhang, G. *An extended branch and bound algorithm for bilevel multifollower decision making in a referential-uncooperative situation*, Int. J. Inf. Tech. Decis. Mak. 6(2) (2007) 371–388.

21. Mersha, A.G. and Dempe, S. *Linear bilevel programming with upper level constraints depending on the lower level solution*, Appl. Math. Comput. 180 (2006) 247–254.

22. Moitra, B.N. and Pal, B.B. *A fuzzy goal programming approach for solving bilevel programming problems*, Pal, N.R., Sugeno, M. (eds.) In: Advances in Soft Computing – AFSS. vol. 2275 of Lecture Notes in Computer Science, PP.91–98, Springer, Berlin, Germany, 2002.

23. Mohamed, R.H. *The relationship between goal programming and fuzzy programming*, Fuzzy Set. Syst. 89 (1997) 215–222.

24. Peric, T., Babic, Z. and Omerovi, M. *A fuzzy goal programming approach to solving decentralized bilevel multiobjective linear fractional programming problems*, Croat. Oper. Res. Rev. 10 (2019) 65–74.

25. Pramanik, S. and Dey, P.P. *Bi-level multiobjective programming problem with fuzzy parameters*, Int. J. Comput. Appl. 30(11) (2011) 13–20.

26. Pramanik, S. and Roy, T.K. *Fuzzy goal programming approach to multilevel programming problems*, Eur. J. Oper. Res. 176(2) (2007) 1151–1166.

27. Sakawa, M. *Fuzzy sets and interactive multiobjective optimization*, Plenum press, New York, 1993.

28. Sakawa, M., Nishizaki, I. and Uemura, Y. *Interactive fuzzy programming for two-level linear fractional programming problems with fuzzy parameters*, Fuzzy Set. Syst. 115 (2000) 93–103.

29. Sakawa, M., Nishizaki, I. and Uemura, Y. *Interactive fuzzy programming for multilevel linear programming problems with fuzzy parameters*, Fuzzy Set. Syst. 109 (2000) 3–19.

30. Shi, C., Lu, J. and Zhang, G. *An extended Kth-best approach for linear bilevel programming*, Appl. Math. Comput. 164 (2005) 843–855.

31. Shi, C., Lu, J., Zhang, G. and Zhou, H. *An extended branch and bound algorithm for linear bilevel programming*, Appl. Math. Comput. 180 (2006) 529–537.

32. Shi, C., Zhang, G. and Lu, J. *On the definition of linear bilevel programming solution*, Appl. Math. Comput. 160(1) (2005) 169–176.

33. Shi, C., Zhang, G. and Lu, J. *An extended Kuhn-Tucker approach for linear bilevel programming*, Appl. Math. Comput. 162(1) (2005) 51–63.

34. Shih, H.S., Lai, Y.J. and Lee, E.S. *Fuzzy approach for multilevel programming problems*, Comput. Oper. Res. 23 (1983) 73–91.

35. Shih, H.S. and Lee, E.S. *Compensatory fuzzy multiple level decision making*, Fuzzy Set. Syst. 14 (2000) 71–87.

36. Toksari, M.D. and Bilim, Y. *Interactive fuzzy goal programming based on Jacobian matrix to solve decentralized bi-level multi-objective fractional programming problems*, Int. J. Fuzzy Syst. 17 (2015) 499–508.

37. Von Stackelberg, H. *The theory of the market economy*, Oxford University Press, York, Oxford, 1952.

38. Wan, Zh.P. and Fei, P.Sh. *The theory and algorithm of optimization*, The Wuhan University Press, 2004.

39. Wang, G., Wang, X., Wan, Z. and Lv, Y. *A globally convergent algorithm for a class of bilevel nonlinear programming problem*, Appl. Math. Comput. 188 (2007) 166–172.

40. Zadeh, L.A. *Fuzzy sets*, Inform. Control, 8 (1965) 338–353.

41. Zadeh, L.A. *Fuzzy sets as a basis for a theory of possibility*, Fuzzy Set. Syst. 1 (1978) 3–28.

42. Zhang, G. and Lu, J. *The definition of optimal solution and an extended Kuhn-Tucker approach for fuzzy linear bilevel programming*, The IEEE Computational Intelligence Bulletin 5 (2005) 1–7.

43. Zhang, G. and Lu, J. *Model and approach of fuzzy bilevel decision making for logistics planning problem*, J. Enterp. Inf. Manag. 20 (2007) 178–197.

44. Zhang, G., Lu, J. and Dillon, T. *A branch-and-bound algorithm for fuzzy bilevel decision making*, in: D. Ruan, P. D'hondt, P. Fantoni, M. De Cock, M. Nachtegael, E. Kerre (Eds.), Applied Artificial Intelligence, World Scientific, Singapore, 2006, pp. 291–298.

45. Zhang, G., Lu, J. and Dillon, T. *An approximation branch and bound approach for fuzzy linear bilevel decision making*, The 1st International Symposium Advances in Artifical Intelligence and Applications (AAIA 06) Wisla, Poland, November 6–10, 2006.

46. Zhang, G., Lu, J. and Dillon, T. *Decentralized multiobjective bilevel decision making with fuzzy demands*, Knowl-Based Syst. 20 (2007) 495–507.

47. Zheng, Y., Wan, Z. and Wang, G. *A fuzzy interactive method for a class of bilevel multiobjective programming problem*, Expert Syst. Appl. 38 (2011) 10384–10388.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 33–54
DOI:10.22067/ijnao.2021.11309.0

Research Article

The time-dependent diffusion equation: An inverse diffusivity problem

S.H. Tabasi*, H.D. Mazraeh, A.A. Irani, R. Pourgholi, and A. Esfahani

Abstract

We find a solution of an unknown time-dependent diffusivity $a(t)$ in a linear inverse parabolic problem by a modified genetic algorithm. At first, it is shown that under certain conditions of data, there exists at least one solution for unknown $a(t)$ in $(a(t), T(x,t))$, which is a solution to the corresponding problem. Then, an optimal estimation for unknown $a(t)$ is found by applying the least-squares method and a modified genetic algorithm. Results show that an excellent estimation can be obtained by the implementation of a modified real-valued genetic algorithm within an Intel Pentium (R) dual-core CPU with a clock speed of 2.4 GHz.

AMS subject classifications (2020): 65M32, 35K05.

Keywords: Inverse parabolic problem; Existence; Uniqueness; Green's function; Fixed point; Contraction mapping; Genetic algorithm.

---

*Corresponding author
Received 5 February 2020; revised 31 August 2020; accepted 26 October 2020
S. Hashem Tabasi
School of Mathematics and Computer Science, Damghan University, P.O.Box 36715-364, Damghan, Iran. e-mail: tabasi@du.ac.ir

Hassan Dana Mazraeh
School of Mathematics and Computer Science, Damghan University, P.O.Box 36715-364, Damghan, Iran. e-mail: dana@du.ac.ir

Arash Azimzadeh Irani
School of Mathematics and Computer Science, Damghan University, P.O.Box 36715-364, Damghan, Iran. e-mail: a.azimzadeh@du.ac.ir

Reza Pourgholi
School of Mathematics and Computer Science, Damghan University, P.O.Box 36715-364, Damghan, Iran. e-mail: pourgholi@du.ac.ir

Amin Esfahani
School of Mathematics and Computer Science, Damghan University, P.O.Box 36715-364, Damghan, Iran. e-mail: esfahani@du.ac.ir

1 Introduction

This article considers helium diffusion kinetics, which is important for materials in which helium measurements are made, particularly for thermochronology (see [2, 4, 3]). Solutions of an inverse problem for the helium production-diffusion equation are explored in this research article. This solution is a time-temperature path derived from apatite fission-track length distribution to characterize the response of apatite helium ages to thermal histories involving partial retention. Mathematically, the helium production-diffusion equation without any source of radiogenic production in a rectangle domain $\Omega = \{(x,y) : 0 < x < 1,\ 0 < y < 1\}$ can be written as (see [2, 4, 3])

$$\frac{\partial T}{\partial t} = a(t)\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \qquad (x,y) \in \Omega,\ t > 0, \tag{1a}$$

$$T(x,y,0) = f(x,y), \qquad (x,y) \in \overline{\Omega}, \tag{1b}$$

$$T(0,y,t) = g_0(y,t), \qquad 0 \le y \le 1,\ t \ge 0, \tag{1c}$$

$$T(1,y,t) = g_1(y,t), \qquad 0 \le y \le 1,\ t \ge 0, \tag{1d}$$

$$\frac{\partial T}{\partial y}(x,0,t) = h_0(x,t), \qquad 0 \le x \le 1,\ t \ge 0, \tag{1e}$$

$$\frac{\partial T}{\partial y}(x,1,t) = h_1(x,t), \qquad 0 \le x \le 1,\ t \ge 0, \tag{1f}$$

where $f(x,y)$, $g_0(y,t)$, $g_1(y,t)$, $h_0(x,t)$, and $h_1(x,t)$ are known functions and $a(t)$ is the time-dependent diffusivity and should be determined in this inverse problem.

Here the special case $h_0 = h_1 = 0$ is considered, that is,

$$\frac{\partial T}{\partial t} = a(t)\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \qquad (x,y) \in \Omega,\ t > 0, \tag{2a}$$

$$T(x,y,0) = f(x,y), \qquad (x,y) \in \Omega, \tag{2b}$$

$$T(0,y,t) = g_0(y,t), \qquad 0 \le y \le 1,\ t \ge 0, \tag{2c}$$

$$T(1,y,t) = g_1(y,t), \qquad 0 \le y \le 1,\ t \ge 0, \tag{2d}$$

$$\frac{\partial T}{\partial y}(x,0,t) = \frac{\partial T}{\partial y}(x,1,t) = 0, \qquad 0 \le x \le 1,\ t \ge 0, \tag{2e}$$

together with an overspecified condition for fixed points $x_0, y_0 \in (0,1)$, as Cannon considered in [7, p. 191],

$$a(t)\frac{\partial T}{\partial x}(x_0, y_0, t) = p(t) \qquad (t \ge 0). \tag{3}$$

Problem (1) in one-dimensional case arises from the heat conduction in a material that is eroded or damaged. This erosion can be caused, for example, by radioactive. Duo to the erosion, thermal characteristics of the material including its heat capacity, diffusivity, and conductivity change based on the level of the erosion. These characteristics can be related to time, where $a(t)$ is the thermal diffusivity, which is the ratio between the thermal conductivity and the heat capacity. Lesnic [14] investigated an inverse problem for determining the temperature and the time-dependent thermal diffusivity from various additional nonlocal information (see also [7, Chapter 13]). It should be remarked that the inverse problem (1) with $a(t) = 1$ has been investigated by several authors; see, for example, [5, 20, 8, 22, 23, 9] and for a two-dimensional case [21]. Since solving inverse diffusivity problems using evolutionary algorithms has not been investigated yet, the solution to these problems using a modified genetic algorithm is considered in this article in the hope that this effort triggers other authors' motivation to examine new artificial intelligence techniques.

2 General solution in term of Green's functions

For $\Omega = (0,1) \times (0,1)$ and $\overline{\Omega} = [0,1] \times [0,1]$, take

$$\begin{aligned} S_1 &= \{\mathbf{x} = (x,y) : x \in \{0,1\},\ 0 \le y \le 1\}, \\ S_2 &= \{\mathbf{x} = (x,y) : y \in \{0,1\},\ 0 \le x \le 1\}. \end{aligned}$$

Then $\partial\Omega = S_1 \cup S_2$. Consider the two-dimensional heat conduction problem with nonhomogeneous boundary conditions

$$\frac{\partial T}{\partial t}(\mathbf{x},t) = \nabla^2 T(\mathbf{x},t), \qquad \mathbf{x} \in \Omega,\ t > 0, \tag{4a}$$

$$T(\mathbf{x},0) = f(\mathbf{x}), \qquad \mathbf{x} \in \overline{\Omega}, \tag{4b}$$

$$T(\mathbf{x},t) = g(\mathbf{x},t), \qquad \mathbf{x} \in S_1,\ t \ge 0, \tag{4c}$$

$$\frac{\partial T}{\partial y}(\mathbf{x},t) = h(\mathbf{x},t), \qquad \mathbf{x} \in S_2,\ t \ge 0, \tag{4d}$$

where $\mathbf{x} = (x,y)$ and

$$g(\mathbf{x},t) = \begin{cases} g_0(y,t), & \mathbf{x} \in S_1, x = 0, \\ g_1(y,t), & \mathbf{x} \in S_1, x = 1, \end{cases} \qquad h(\mathbf{x},t) = \begin{cases} h_0(x,t), & \mathbf{x} \in S_2, y = 0, \\ h_1(x,t), & \mathbf{x} \in S_2, y = 1. \end{cases}$$

Here g_0, g_1, h_0, h_1, and f are known functions. It is known that (4) possesses a unique solution provided the given initial and boundary data are continuous functions; see, for example, [7, 10, 19]. The aim of this section is to find a representation of the solution of (4). To this end, as in [7], define, for $x \in \mathbb{R}$, $t > 0$,

$$K(x,t) = \frac{1}{\sqrt{4\pi t}} \exp\Big(\frac{-x^2}{4t}\Big), \quad \theta(x,t) = \sum_{m=-\infty}^{\infty} K(x+2m,t),$$

and define the function $\Phi(\mathbf{x},t) = K(x,t)K(y,t)$ for $\mathbf{x} = (x,y) \in \mathbb{R}^2$, $t > 0$ by

$$\Phi(\mathbf{x},t) = \frac{1}{4\pi t} \exp\Big(-\frac{||\mathbf{x}||^2}{4t}\Big).$$

It is well known from [12, Chapter 7], that

$$T(\mathbf{x},t) = \Phi *_{\mathbf{x}} f = \int_{\mathbb{R}^2} \Phi(\mathbf{x}-\boldsymbol{\xi},t) f(\boldsymbol{\xi}) d\boldsymbol{\xi} \qquad (\mathbf{x} \in \mathbb{R}^2,\ t > 0), \tag{5}$$

where $T(\mathbf{x},t)$ is a solution for a pure initial value problem in the equation

$$\begin{aligned} \frac{\partial T}{\partial t}(\mathbf{x},t) &= \nabla^2 T(\mathbf{x},t), \\ T(\mathbf{x},0) &= f(\mathbf{x}), \end{aligned}$$

for $\mathbf{x} \in \mathbb{R}^2$ and $t > 0$, provided that the function $f(\mathbf{x})$ is bounded and also (piecewise) continuous in $\mathbb{R}^2$. Now, we extend $T(\mathbf{x},t)$ continuously to $\overline{\Omega}$, and consider the following heat conduction problem in Ω with homogeneous boundary conditions:

$$\frac{\partial T}{\partial t}(\mathbf{x},t) = \nabla^2 T(\mathbf{x},t), \qquad \mathbf{x} \in \Omega,\ t > 0, \tag{6a}$$

$$T(\mathbf{x},0) = f(\mathbf{x}), \qquad \mathbf{x} \in \overline{\Omega}, \tag{6b}$$

$$T(\mathbf{x},t) = 0, \qquad \mathbf{x} \in S_1,\ t \geq 0, \tag{6c}$$

$$\frac{\partial T}{\partial y}(\mathbf{x},t) = 0, \qquad \mathbf{x} \in S_2,\ t \geq 0. \tag{6d}$$

To solve problem (6), one can first extend $f(\mathbf{x})$ to $[-1,1] \times [-1,1]$ such that f is odd in x and even in y. Then we extend f periodically to $\mathbb{R}^2$ by

$$f(x+2m, y+2n) = f(x,y), \qquad (n,m \in \mathbb{Z}).$$

If we define, for $x, \xi \in \mathbb{R}$, $t > 0$,

$$\Theta^{\pm}(x,\xi,t) = \theta(x-\xi,t) \mp \theta(x+\xi,t),$$

then we can obtain from (5) for $T(\mathbf{x},t)$ that

$$T(\mathbf{x},t) = \int_{\Omega} \Theta^{-}(x,\xi,t)\Theta^{+}(y,\eta,t) f(\xi,\eta) d\xi d\eta. \tag{7}$$

Now, define $\mathcal{G}$ by

$$\mathcal{G}(x,y,t;\xi,\eta) = \Theta^{+}(y,\eta,t)\Theta^{-}(x,\xi,t). \tag{8}$$

Since $\theta(x,t)$ is an even function with respect to the variable x for all values of $t>0$, it is easy to observe for every ξ,η,t that

$$\begin{aligned}\mathcal{G}(x,y,t;\xi,\eta) &= 0, \qquad (x,y)\in S_1,\\ \frac{\partial}{\partial y}\mathcal{G}(x,y,t;\xi,\eta) &= 0, \qquad (x,y)\in S_2.\end{aligned}$$

So $T(\mathbf{x},t)$, defined by (7), is, in fact, the solution of (6). This means that the function $\mathcal{G}$, defined by (8), is the Green function for problem (4), so that the solution of (4) can be written (see [19, Chapter 6] or [13, Chapter 7]) by

$$\begin{aligned}T(\mathbf{x},t) = &\int_\Omega \mathcal{G}(\mathbf{x},t;\boldsymbol{\xi})f(\boldsymbol{\xi})d\boldsymbol{\xi}\\ &+\int_0^t d\tau\int_{S_1}\frac{\partial\mathcal{G}}{\partial\mathbf{n}_1}(\mathbf{x},t-\tau;\boldsymbol{\xi})g(\boldsymbol{\xi},\tau)d\eta \qquad (9)\\ &-\int_0^t d\tau\int_{S_2}(\mathbf{x},t-\tau;\boldsymbol{\xi})h(\boldsymbol{\xi},\tau)d\xi, \qquad (10)\end{aligned}$$

where $\mathbf{x}=(x,y)$, $\boldsymbol{\xi}=(\xi,\eta)$, $\boldsymbol{\xi}\in S_1$ in (9), and $\boldsymbol{\xi}\in S_2$ in (10); and $(\partial/\partial\mathbf{n}_1)$, in (9), denotes the directional derivative on S_1. Therefore, one can write the solution $T(x,y,t)$ of (4) in the following form:

$$\begin{aligned}T(x,y,t) = &\int_\Omega \mathcal{G}(x,y,t;\xi,\eta)f(\xi,\eta)d\xi d\eta\\ &-2\int_\Omega g_0(\eta,\tau)\frac{\partial\theta}{\partial x}(x,t-\tau)\Theta^{+}(y,\eta,t-\tau)d\eta d\tau\\ &+2\int_\Omega g_1(\eta,\tau)\frac{\partial\theta}{\partial x}(x-1,t-\tau)\Theta^{+}(y,\eta,t-\tau)d\eta d\tau\\ &-2\int_\Omega h_0(\eta,\tau)\theta(y,t-\tau)\Theta^{-}(x,\xi,t-\tau)d\xi d\tau\\ &-2\int_\Omega h_1(\eta,\tau)\theta(y-1,t-\tau)\Theta^{-}(x,\xi,t-\tau)d\xi d\tau.\end{aligned}$$

Assuming that $a(t)$ is given, one should note that under some suitable hypotheses on the initial and boundary conditions, the existence of a unique solution to (6) can be proved.

Theorem 1. Let $\tau>0$, $\Omega=(0,1)$ and $f\in L^2(\Omega)$. Suppose also that $a\in L^\infty((0,\tau))$ and $a(t)\geq 0$, for almost every $t\in(0,\tau)$. Then there exists a unique function u of (6) satisfying

$$T\in L^2\left((0,\tau);H_0^1(\Omega)\right)\cap C\left([0,\tau];L^2(\Omega)\right)$$

and

$$\frac{\partial T}{\partial t} \in L^2\left((0,\tau); H_0^{-1}(\Omega)\right),$$

where $H_0^1(\Omega)$ and $H_0^{-1}(\Omega)$ are the usual Sobolev spaces.

Proof. The proof of Theorem 1 follows from [16, Theorem 7.1] (see also [1, 15, 17, 18]). □

Remark 1. If a is given and the assumptions of Theorem 1 hold, then one can obtain a unique solution $T = T(x, y, t; a)$ of (6). Hence some consistency conditions, in boundary, should hold between a and u in (3). Consequently, the coefficient a is uniquely determined.

Remark 2. One should note that Theorem 1 improves [3, Lemma 2.1].

3 Utilizing an appropriate transformation

Turn back to the main problem (2), in which $u(x, y, t)$ and $a(t)$ are unknown. Denote by $\mathcal{F}$ the family of all continuous functions $a : \mathbb{R} \to \mathbb{R}$ such that $a(t) > 0$, for all $t \in \mathbb{R}$. For $a \in \mathcal{F}$, let (see [7])

$$s = \alpha(t) = \int_0^t a(\tau)d\tau \qquad (t \geq 0).$$

Then $\alpha(0) = 0$ and $\alpha'(t) = a(t) > 0$, and thus $s = \alpha(t)$ is a nonnegative invertible function with inverse $t = \beta(s)$. By plugging

$$\begin{gathered} v(x, y, s) = u(x, y, \beta(s)), \\ \phi_0(y, s) = g_0(y, \beta(s)), \quad \phi_1(y, s) = g_1(y, \beta(s)), \\ b(s) = a(\beta(s)), \quad q(s) = p(\beta(s)), \end{gathered}$$

problem (2) is rephrased in the following form:

$$\begin{aligned} v_s &= v_{xx} + v_{yy}, && (x, y) \in \Omega,\ s > 0, && \text{(11a)} \\ v(x, y, 0) &= f(x, y), && (x, y) \in \overline{\Omega}, && \text{(11b)} \\ v(0, y, s) &= \phi_0(y, s), && 0 \leq y \leq 1,\ s \geq 0, && \text{(11c)} \\ v(1, y, s) &= \phi_1(y, s), && 0 \leq y \leq 1,\ s \geq 0, && \text{(11d)} \\ v_y(x, 0, s) &= v_y(x, 1, s) = 0, && 0 \leq x \leq 1,\ s \geq 0. && \text{(11e)} \end{aligned}$$

In this setting, the over-specified condition (3) implies that

$$b(s)v_x(0, y_0, s) = q(s) \qquad (s \geq 0).$$

By [7, Chapter 13], the solution of (11) is represented as follows:

$$
\begin{aligned}
v(x,y,s) = &\int_\Omega f(\xi,\eta)\mathcal{G}(x,y,s;\xi,\eta)d\xi d\eta \\
&- 2\int_0^s\int_0^1 \frac{\partial\theta}{\partial x}(x,s-\zeta)\Theta^+(y,\eta,s-\zeta)\phi_0(\eta,\zeta)d\eta d\zeta \\
&+ 2\int_0^s\int_0^1 \frac{\partial\theta}{\partial x}(x-1,s-\zeta)\Theta^+(y,\eta,s-\zeta)\phi_1(\eta,\zeta)d\eta d\zeta.
\end{aligned}
\tag{12}
$$

Applying the change of the variable $\zeta = \alpha(\tau)$ in (12), one obtains

$$
\begin{aligned}
u(x,y,t) = &\int_\Omega f(\xi,\eta)\mathcal{G}(x,y,\alpha(t);\xi,\eta)d\xi d\eta \\
&- 2\int_0^t\int_0^1 \frac{\partial\theta}{\partial x}(x,\alpha(t)-\alpha(\tau))\Theta^+(y,\eta,\alpha(t)-\alpha(\tau))g_0(\eta,\tau)d\eta a(\tau)d\tau \\
&+ 2\int_0^t\int_0^1 g_1(\eta,\tau)\frac{\partial\theta}{\partial x}(x-1,\alpha(t)-\alpha(\tau))\Theta^+(y,\eta,\alpha(t)-\alpha(\tau))d\eta a(\tau)d\tau.
\end{aligned}
\tag{13}
$$

For the sake of simplicity, from now on, let

$$
\int_\tau^t a = \int_\tau^t a(\zeta)d\zeta = \alpha(t) - \alpha(\tau).
$$

Using (13) and calculating $u_x(x,y,t)$, one writes $u_x(x,y,t) = J_1 + J_2 + J_3$, where

$$
\begin{aligned}
J_1 &= \int_\Omega f(\xi,\eta)\frac{\partial}{\partial x}\mathcal{G}(x,y,\alpha(t);\xi,\eta)d\xi d\eta, \\
J_2 &= -2\int_0^t\int_0^1 \frac{\partial^2\theta}{\partial x^2}(x,\int_\tau^t a)\Theta^+(y,\eta,\int_\tau^t a)g_0(\eta,\tau)d\eta a(\tau)d\tau, \\
J_3 &= 2\int_0^t\int_0^1 \frac{\partial^2\theta}{\partial x^2}(x-1,\int_\tau^t a)\Theta^+(y,\eta,\int_\tau^t a)g_1(\eta,\tau)d\eta a(\tau)d\tau.
\end{aligned}
$$

It is known that $\theta_t(x,t) = \theta_{xx}(x,t)$ (see [7]), so

$$
\frac{\partial^2\theta}{\partial x^2}(x,\int_\tau^t a) = \frac{\partial\theta}{\partial t}(x,\int_\tau^t a) = -\frac{1}{a(\tau)}\frac{\partial}{\partial\tau}\theta(x,\int_\tau^t a).
$$

Since $g_0(y,0) = g_1(y,0) = 0$ and $\theta(x,0) = 0$ for $x \neq 0$ (see [7]), integration by parts yields that

$$\begin{aligned} J_2 &= -2\int_0^t\int_0^1 -\frac{\partial}{\partial\tau}\theta(x,\int_\tau^t a)\Theta^+(y,\eta,\int_\tau^t a)g_0(\eta,\tau)d\eta d\tau \\ &= -2\int_0^t\int_0^1 \theta(x,\int_\tau^t a)\frac{\partial}{\partial\tau}\Big[\Theta^+(y,\eta,\int_\tau^t a)g_0(\eta,\tau)\Big]d\eta d\tau, \\ J_3 &= 2\int_0^t\int_0^1 \theta(x-1,\int_\tau^t a)\frac{\partial}{\partial\tau}\Big[\Theta^+(y,\eta,\int_\tau^t a)g_1(\eta,\tau)\Big]d\eta d\tau. \end{aligned}$$

Now by applying the over-specified condition (3), we have

$$\begin{aligned} \frac{p(t)}{a(t)} &= \int_\Omega f(\xi,\eta)\frac{\partial}{\partial x}\mathcal{G}(x_0,y_0,\alpha(t);\xi,\eta)d\xi d\eta, \\ &\quad -2\int_0^t\int_0^1 \theta(x_0,\int_\tau^t a)\frac{\partial}{\partial\tau}\Big[\Theta^+(y_0,\eta,\int_\tau^t a)g_0(\eta,\tau)\Big]d\eta d\tau \\ &\quad +2\int_0^t\int_0^1 \theta(x_0-1,\int_\tau^t a)\frac{\partial}{\partial\tau}\Big[\Theta^+(y_0,\eta,\int_\tau^t a)g_1(\eta,\tau)\Big]d\eta d\tau. \end{aligned} \tag{14}$$

Thus one obtains the following result.

Theorem 2. Suppose that the function $f(x,y)$ is piecewise continuous on $\overline{\Omega}$, that $g_0(y,t)$, $g_1(y,t)$, $(\partial/\partial t)g_0(y,t)$, and $(\partial/\partial t)g_1(y,t)$ are continuous functions for $0 \le y \le 1$, $t \ge 0$, and that $g_0(y,0) = g_1(y,0) = 0$ for all y [or just $y = y_0$]. Then problem (2) possesses a unique solution (u,a) if and only if the integral equation (14) has a unique solution $a \in \mathcal{F}$.

4 Uniqueness and existence of solution for a certain problem

Now, consider the problem

$$\frac{\partial T}{\partial t} = a(t)(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}), \qquad (x,y)\in\Omega,\ t>0, \tag{15a}$$

$$T(x,y,0) = 0, \qquad (x,y)\in\overline{\Omega}, \tag{15b}$$

$$T(0,y,t) = 0, \qquad 0\le y\le 1,\ t\ge 0, \tag{15c}$$

$$T(1,y,t) = g(y,t), \qquad 0\le y\le 1,\ t\ge 0, \tag{15d}$$

$$\frac{\partial T}{\partial y}(x,0,t) = \frac{\partial T}{\partial y}(x,1,t) = 0, \qquad 0\le x\le 1,\ t\ge 0, \tag{15e}$$

and the overspecified condition

$$a(t)\frac{\partial T}{\partial x}(0,0,t) = p(t), \qquad (t\ge 0).$$

The solution of (15) is as follows:

$$T(x,y,t) = +2\int_0^t\!\!\int_0^1 g(\eta,\tau)\frac{\partial\theta}{\partial x}(x-1,\int_\tau^t a)\Theta^+(y,\eta,\int_\tau^t a)d\eta a(\tau)d\tau.$$

The integral equation (14) becomes

$$\begin{aligned}\frac{p(t)}{a(t)} =& 2\int_0^t\!\!\int_0^1 \theta(-1,\int_\tau^t a)\frac{\partial}{\partial\tau}\Big[\Theta^+(0,\eta,\int_\tau^t a)g(\eta,\tau)\Big]d\eta d\tau \\ =& 2\int_0^t\!\!\int_0^1 \theta(-1,\int_\tau^t a)\Big[-a(\tau)\Theta_t^+ g + \Theta^+ g_\tau\Big]d\eta d\tau.\end{aligned} \tag{16}$$

Moreover,

$$\begin{aligned}\frac{p'(t)a(t)-p(t)a'(t)}{a(t)^2} =& 2a(t)\int_0^t d\tau \\ &\times\int_0^1 \theta_t(-1,\int_\tau^t a)\Big[-a(\tau)\Theta_t^+ g(\eta,\tau) + \Theta^+\frac{\partial g}{\partial\tau}(\eta,\tau)\Big] \\ &+ \theta(-1,\int_\tau^t a)\Big[-a(\tau)\Theta_{tt}^+ g(\eta,\tau) + \Theta_t^+\frac{\partial g}{\partial\tau}(\eta,\tau)\Big]d\eta.\end{aligned}$$

Remark 3. Since $\Theta^+(0,\eta,\cdot) = 2\theta(\eta,\cdot)$ and $\Theta_t^+(0,\eta,\cdot) = 2\theta_t(\eta,\cdot)$, equation (16) can be rephrased in the following forms:

$$\begin{aligned}a(t) =& \frac{p(t)}{2}\Big\{\int_0^t \theta(-1,\int_\tau^t a)\int_0^1 [\theta g_\tau - a(\tau)\theta_t g]d\eta d\tau\Big\}^{-1} \\ =& \frac{p(t)}{2}\Big\{\int_0^t \theta(-1,\int_\tau^t a)\int_0^1 [\theta g_\tau + a(\tau)\theta_{xx} g]d\eta d\tau\Big\}^{-1}.\end{aligned}$$

Lemma 1. If $g = g(t)$, then one can write (16) in the following form:

$$a(t) = \frac{p(t)}{2}\Big\{\int_0^t \theta(1,\int_\tau^t a)g'(\tau)d\tau\Big\}^{-1}. \tag{17}$$

Proof. If $g = g(t)$, then

$$\begin{aligned}\frac{\partial T}{\partial x}(0,0,t) =& 4\int_0^t \theta(-1,\int_\tau^t a)\Big\{\frac{g'(\tau)}{2} + a(\tau)g(\tau)\Big[\theta_x(1,\int_\tau^t a) - \theta_x(0,\int_\tau^t a)\Big]\Big\}d\tau \\ =& 2\int_0^t \theta(-1,\int_\tau^t a)g'(\tau)d\tau.\end{aligned}$$

□

Corollary 1. Under the assumption that $g = g(t)$ is continuously differentiable and that $g(0) = 0$, problem (15) possesses a unique solution (u,a) if and only if the nonlinear integral equation (17) possesses a unique positive solution $a(t)$ that is continuous for $0 \le t < T$.

Definition 1. Suppose that $g = g(t)$ is continuously differentiable and that $g(0) = 0$. For a positive continuous function $a(t)$, that is, $a \in \mathcal{F}$, define

$$(\mathcal{L}a)(t) = \frac{p(t)}{2}\left\{\int_0^t \theta(1, \int_\tau^t a) g'(\tau) d\tau\right\}^{-1}.$$

Therefore, a is a solution of (17) if and only if a is a fixed point of the operator $\mathcal{L}$, that is, $\mathcal{L}a = a$. In the rest of this section, it is proved that under certain conditions, $\mathcal{L}$ possesses a fixed point.

For a function h defined on $[0, T]$, let

$$\|h\|_t = \sup\{|h(s)| : 0 \leq s \leq t\}.$$

For a positive constant δ, if $\delta \leq a(t)$ and $\delta \leq b(t)$, then, using the mean value theorem, one obtains

$$\begin{aligned}
|(\mathcal{L}a)(t) - (\mathcal{L}b)(t)| &\leq \frac{|p(t)|}{2}\Big|\int_0^t \theta_t(1, \delta(t-\tau)) g'(\tau) d\tau\Big|^{-2} \\
&\qquad \times \Big|\int_0^t \theta_t(1, \gamma)\Big[\int_\tau^t a(s) - b(s) ds\Big] g'(\tau) d\tau\Big| \\
&\leq \frac{|p(t)|}{2}\Big|\int_0^t \theta_t(1, \delta(t-\tau)) g'(\tau) d\tau\Big|^{-2} \|\theta_t\|_t \|a - b\|_t t |g(t)|,
\end{aligned}$$

where $\int_\tau^t a \leq \gamma \leq \int_\tau^t b$.

Corollary 2. Suppose for some $r \in (0, 1)$ that

$$\frac{|tp(t)g(t)|}{2}\Big|\int_0^t \theta_t(1, \delta(t-\tau)) g'(\tau) d\tau\Big|^{-2} \|\theta_t\|_t \leq r \qquad (t \in [0, T]). \tag{18}$$

Then the mapping $a \mapsto \mathcal{L}a$ is a contraction.

The main result of this section will be obtained by using (17) and Corollaries 1 and 2.

Theorem 3. Suppose that $p(t)$ and $g = g(t)$ are continuously differentiable, that $g(0) = 0$, and that, for some $r \in (0, 1)$, (18) holds. Then (15) possesses a unique solution (T, a).

5 Discretization of the problem

In this section, the following linear inverse parabolic problem is considered:

$$\frac{\partial T}{\partial t}(x,t) = a(t)(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}), \quad 0 < x, y < 1, 0 < t < t_M, \tag{19a}$$

$$T(x,y,0) = f(x,y), \qquad 0 \leq x, y \leq 1, \tag{19b}$$

$$T(0,y,t) = g_0(y,t), \qquad 0 \leq y \leq 1, 0 \leq t \leq t_M, \tag{19c}$$

$$T(1,y,t) = g_1(y,t), \qquad 0 \leq y \leq 1, 0 \leq t \leq t_M, \tag{19d}$$

$$T(x,0,t) = h_0(x,t), \qquad 0 \leq x \leq 1, 0 \leq t \leq t_M, \tag{19e}$$

$$T(x,1,t) = h_1(x,t), \qquad 0 \leq x \leq 1, 0 \leq t \leq t_M, \tag{19f}$$

with the overspecified conditions for the fixed points x_0,$y_0 \in (0,1)$,

$$T(x_0,y_0,t) = s(t), \qquad 0 \leq t \leq t_M, \tag{19g}$$

where $f(x,y)$, $g_0(y,t)$, $g_1(y,t)$, $h_0(x,t)$, $h_1(x,t)$, and $\phi(t)$ are known functions while $a(t)$ and $T(x,y,t)$ are unknown in this inverse problem.

Now we can solve problem (19) in the least-square method. Also, we can consider the amount of difference between calculated values of the problem at the location of sensor and measured temperatures, which come from sensor as the cost function. When the cost function approaches to zero, the estimated value for $a(t)$ approaches to exact $a(t)$. Let

$$f(G) = \sum_{j=1}^{m} (T_j - s_j)^2, \tag{20}$$

where T_j, $j = 1,2,3,\ldots,m$, are calculated by solving the direct heat problem. To do this, consider a prior guess for $a(t)$. Also $s_j = s(t_j)$, $j = 1,2,3,\ldots,m$, are measured temperatures. Here, when (20) reaches its minimum, the estimated value for $a(t)$ is optimal.

Remark 4. In this article, we use an explicit finite difference approximation to discretize problem (19) as follows:

$$\begin{aligned}
T_{i,j,s+1} = ka_s(r_1 T_{i-1,j,s} - 2r_1 Ti,j,s + r_1 Ti+1,j,s + r_2 T_{i,j-1,s} - 2r_2 Ti,j,s \\
+ r_2 Ti,j+1,s) + T_{i,j,s}, \\
i = 1,\ldots,N-1,\ j = 1,\ldots,N-1,\ S = 0,\ldots,N-1,
\end{aligned}$$

$$\begin{aligned}
T_{i,j,0} &= f(ih,jk), \quad i = 1,\ldots,N-1,\ j = 1,\ldots,N-1, \\
T_{0,j,s} &= g_0(jk,sw), \quad j = 1,\ldots,N-1,\ s = 0,\ldots,N-1, \\
T_{1,j,s} &= g_1(jk,sw), \quad j = 1,\ldots,N-1,\ s = 0,\ldots,N-1, \\
T_{i,0,s} &= h_0(ih,sw), \quad i = 1,\ldots,N-1,\ s = 0,\ldots,N-1, \\
T_{1,j,s} &= h_1(ih,sw), \quad i = 1,\ldots,N-1,\ s = 0,\ldots,N-1,
\end{aligned}$$

where $x = ih$, $y = jk$, and $t = sw$.

6 Genetic algorithm

The genetic algorithm was introduced by Holland [11] and has been used to solve a wide range of problems. This algorithm is a searching method that has been inspired by Darwinian principles of biological evolution. This algorithm utilizes an initial stochastic set of candidate solutions as an initial population. Each individual of this set is a numeric vector that estimates a possible solution to the problem. Then, genetic operators such as *Mutation* and *Recombination* are successively applied to initial populations in order to simulate a biological evolution. These operators push all populations toward optimal solutions to the problems. After a predefined number of iterations or termination conditions, the best individual in the population is considered as an optimal solution to the problem.

In the classic genetic algorithms, the vectors are a binary string and present a possible solution to the problem. Indeed there are many problems in which a real-valued vector is needed as an optimal solution. In this article, a real-valued genetic algorithm (RVGA) is used to estimate unknown $a(t)$ in its corresponding time. The steps of our RVGA in this article are as follows:

Step 1. A random set of real-valued vectors are generated as an initial population.

Step 2. Each individual in the population is evaluated by the fitness function.

Step 3. Some individuals are selected based on their fitness by *Tournament Selection*.

Step 4. A recombination operator is applied to the selected individuals to generate offsprings.

Step 5. The Mutation operation is applied to offspring.

Step 6. The offsprings are evaluated accordingly.

Step 7. The population is updated.

Step 8. Steps 3–7 are repeated, until the termination condition satisfies.

7 A modified real-valued genetic algorithm to estimate unknown $a(t)$

In this research article, a modified genetic algorithm is presented to determine unknown $a(t)$. This algorithm employs a real-valued vector as a candidate solution to the problem. In this algorithm, a new step is added to the procedure of the original genetic algorithm after the *Mutation operator* to increase the capability of the algorithm for solving the problems of this article. In the presented algorithm, the jth element of each vector is a real number that estimates $a(j * .001)$ for $j = 0, 1, 2, \ldots, N-1$. Each vector is considered as a chromosome in the population, and accordingly each element of chromosomes is considered as a gene. Therefore, $g_{p,j}$ is the jth gene of chromosome of p. To reach an acceptable estimation for $a(t)$, equation (20) should reach its minimum. To meet this purpose, equation (20) is considered as the fitness function of the algorithm. When the algorithm stops, a vector with the lowest value of fitness is considered as the best solution of $a(t)$. Then this vector is interpolated to estimate $a(t)$ as a polynomial function.

The steps of our presented real-valued genetic algorithm (RVGA) are as follows:

Step 1. A random set of real-valued vectors is generated as the initial population.

Step 2. Each individual in the population is evaluated by the fitness function.

Step 3. Some individuals are selected based on their fitness by *Tournament Selection*.

Step 4. To apply the recombination operator on the pair of parents, we act as follows:

$$g_{ch1,j} = \alpha \times g_{p1,j} + (1-\alpha) \times g_{p2,j}, \qquad j = 1, 2, 3, \ldots, N-1,$$

$$g_{ch2,j} = \beta \times g_{p1,j} + (1-\beta) \times g_{p2,j}, \qquad j = 1, 2, 3, \ldots, N-1,$$

Step 5. To apply the *Mutation* operator on the offsprings, a gene of each offspring is selected randomly and a small random value is added to it.

Step 6. Fitness of offsprings is evaluated.

Step 7. All other offsprings are moved toward the best offspring. Then a small random value is added to a gene of each offspring selected randomly to ensure the variety of the population.

Step 8. The population is updated.

Step 9. Steps 3–7 are repeated, until the termination condition satisfies.

Table 1: Parameters of the real-valued genetic algorithm

Representation	Real-valued vector
Recombination	Arithmetic crossover
Recombination probability	100%
Mutation	Add a random value to one gene
Mutation probability	$1/n$ for each gene
Parents selection	Best 4 out of random 10
Survivor selection	All of the old generations replaced with new offsprings
Number of offsprings	4
Population size	20
Initialization	Random
Termination condition	Number of generations

Table 1 shows the parameters of the RVGA and their roles.

8 Numerical results and discussion

The main purpose of this section is to show the capability of our presented algorithm in the previous section for solving IPP. The capability of the algorithm is investigated by an example of IPP as follows.

Example 1. In this example, for $(x, y, t) \in (0,1) \times (0,1) \times (0,1)$, consider the IPP

$$\frac{\partial T}{\partial t}(x,y,t) = a(t)\Big(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\Big), \tag{22a}$$

$$T(x,y,0) = (\cos(\pi y) + \sin(\pi x)), \tag{22b}$$

$$T(0,y,t) = e^{-\pi^2 t^2}\cos(\pi y), \tag{22c}$$

$$T(1,y,t) = e^{-\pi^2 t^2}(\cos(\pi y) + \sin(\pi)), \tag{22d}$$

$$T(x,0,t) = e^{-\pi^2 t^2}(\sin(\pi x) + 1), \tag{22e}$$

$$T(x,1,t) = e^{-\pi^2 t^2}(\cos(\pi) + \sin(\pi x)), \tag{22f}$$

and with the following overspecified condition

$$s(t_j) = T(0.1, 0.1, t_j) + \sigma R, \quad t_j = (0.001)j, \quad j = 0,1,2,\ldots,999. \tag{22g}$$

In this problem, the values of $s(t_j)$'s are gained from the exact solution of the problem. It is evident that in the real-world application of this algorithm,

Table 2: Result of the execution of the modified RVGA, with 10 to 100 generations when the noise of the measurement is on the s (overspecified condition).

Gen.	Fitness	Time(s)	S
10	1.2936	371.1635	0.1821
20	1.1459	715.8062	0.2162
30	0.6429	1063.0171	0.0823
40	0.7662	1446.6704	0.0966
50	0.4227	2012.2701	0.0515
60	0.4109	2171.3517	0.0529
70	0.3383	2543.9981	0.0435
80	0.3379	2890.7852	0.0168
90	0.2945	3254.4581	0.0080
100	0.2407	3663.9104	0.0064

these values come from sensors. It is obvious that all sensors have a small error of measurement. Therefore, to simulate these errors, a small random value is added to $s(t_j)$'s to have noisy data.

In this example, $a(t) = 2t$ and

$$T(x, y, t) = e^{-\pi^2 t^2}(\cos(\pi y) + \sin(\pi x)).$$

Remark 5. In this article, we use the following equation to calculate total error S, in our numerical computations (see [6]):

$$S = \left[\frac{1}{(N-1)} \sum_{i=1}^{N} (\widehat{a_i} - a_i)^2\right]^{\frac{1}{2}},$$

where $(N - 1)$ is the length of estimated vector (chromosome), $\widehat{a_i}, i = 1, 2, \cdots, N$, come from interpolated function, and $a_i, i = 1, 2, \cdots, N$, are the exact values of $a(t)$.

In this section, the population size of the algorithm is 20. Each chromosome in the population has 1000 genes ($t = 0.001, 0.002, 0.003, \ldots, 0.999$). Table 2 shows the result of the execution of the modified RVGA, with 10 to 100 generations when the noise (noisy data = input data + (0.00001)rand(1)) is on s. To study the ability of the algorithm to solve an IPP, we have severally executed it while the noise of the measurement is on the different functions. Table 3 illustrates the result of those executions. Also, Figures 1–6 show the executions.

Remark 6. One should note that by tending to the real value of $a(t)$, the fitness value (the difference of the calculated and measured values in the sensor position) will be reduced (see (20)). Thus the total error S will decrease.

Table 3: Result of the different executions when the noise of the measurement is on the different functions.

Noisy data	Gen.	Fitness	Time(s)	S
$s(t_j)$ *and* f	100	0.1381	3551.8039	0.0216
$s(t_j)$ *and* g_0	100	0.4914	3602.6282	0.0525
$s(t_j)$ *and* g_1	100	0.3511	3649.0969	0.0385
$s(t_j)$ *and* h_0	100	0.4028	3704.7115	0.0627
$s(t_j)$ *and* h_1	100	0.3718	3679.4812	0.0373

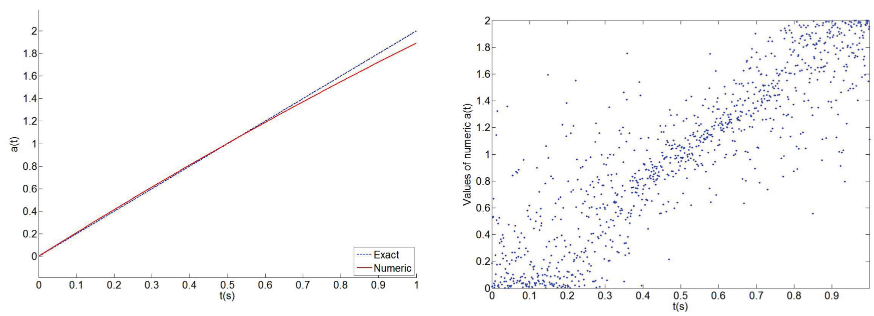


Figure 1: The exact and approximated $a(t)$ (Left) and the best chromosome (Right) with the noisy data on condition (22g).

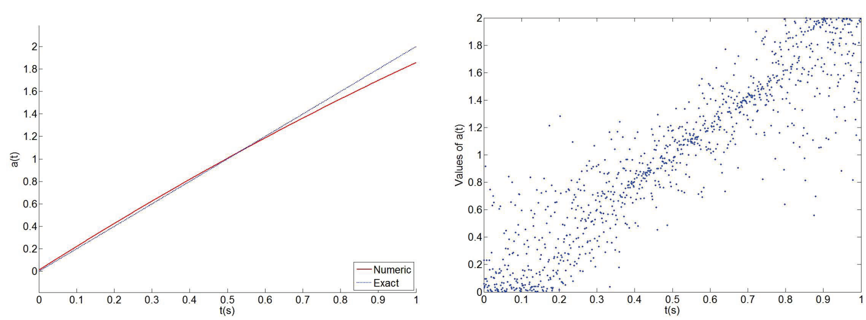


Figure 2: The exact and approximated $a(t)$ (Left) and the best chromosome (Right) with the noisy data on conditions (22g) and (22b).

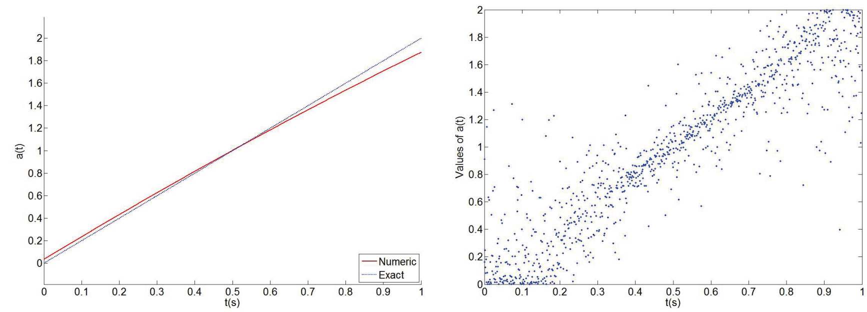


Figure 3: The exact and approximated $a(t)$ (Left) and the best chromosome (Right) with the noisy data on conditions (22g) and (22c).

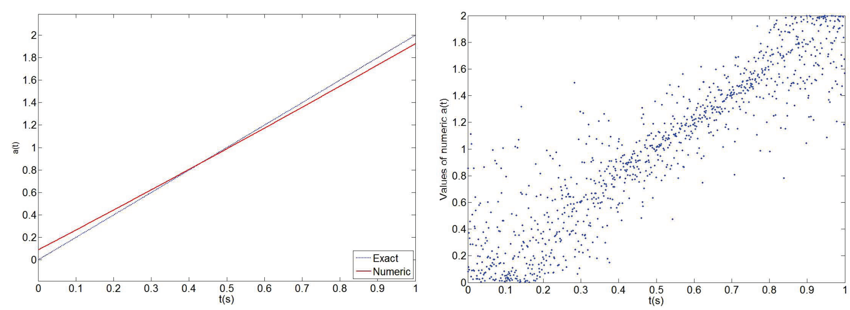


Figure 4: The exact and approximated $a(t)$ (Left) and the best chromosome (Right) with the noisy data on conditions (22g) and (22d).

8.1 Computational time and accuracy comparison

To study computational time and accuracy efficiency, the example of this section is solved by a general real-valued genetic algorithm. Table 4 shows the execution of a general real-valued genetic algorithm for the number of generations from 10 to 100. Also, Figures 7, 8, and 9 illustrate a comparative study of execution time, fitness, and total error (S) between the modified and general real-valued genetic algorithm, respectively.

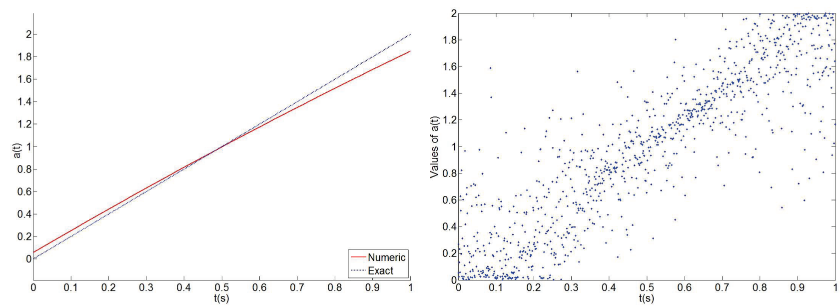


Figure 5: The exact and approximated $a(t)$ (Left) and the best chromosome (Right) with the noisy data on conditions (22g) and (22e).

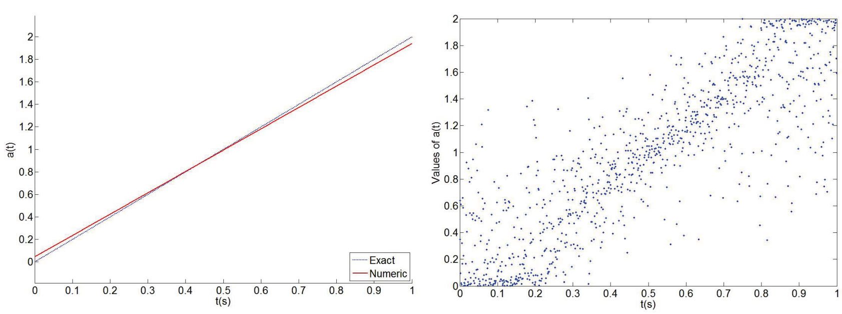


Figure 6: The exact and approximated $a(t)$ (Left) and the best chromosome (Right) with the noisy data on conditions (22g) and (22f).

Table 4: The execution of a general RVGA, with 10 to 100 generations when the noise of the measurement is on the s (overspecified condition).

Gen.	Fitness	Time(s)	S
10	3.0323	318.4011	0.3824
20	2.9706	672.6387	0.3792
30	1.4854	992.8147	0.2140
40	1.4103	1156.1270	0.2041
50	1.1419	1434.9134	0.2043
60	1.1557	1617.6324	0.1285
70	1.0357	1701.0975	0.1253
80	1.1049	1927.8542	0.1077
90	0.9340	2154.6946	0.1174
100	0.9267	2315.6700	0.0939

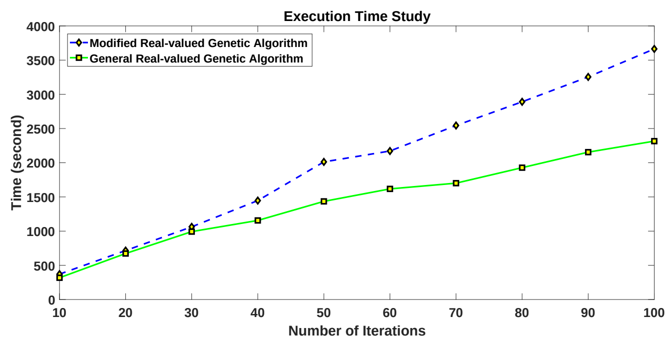


Figure 7: Comparative execution time study between the modified and a general RVGA.

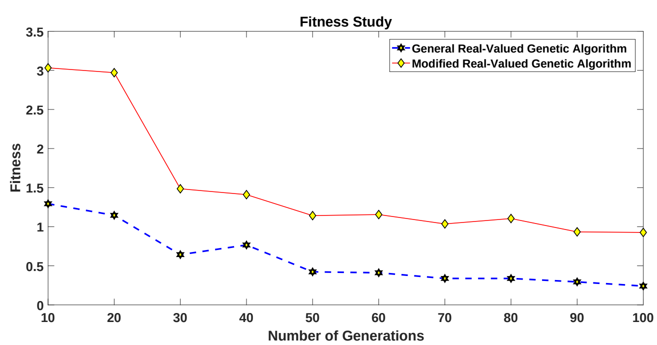


Figure 8: Comparative fitness study between the modified and a general RVGA.

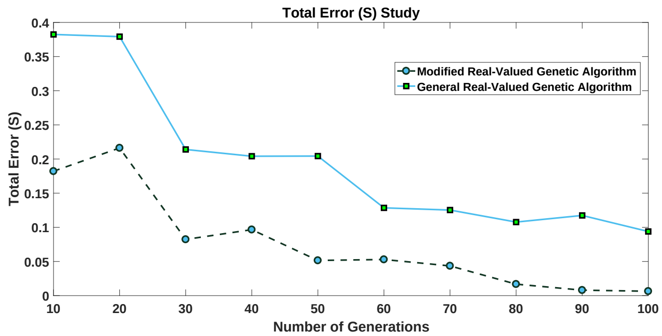


Figure 9: Comparative total error (S) study between the modified and a general RVGA.

It is evident from Figure 7 that the execution time of the modified and the general RVGA increase steadily as the number of generations grows from 10 to 100. Compared to the general RVGA, the modified RVGA generally has a higher run time due to an additional step. In contrast to the execution time, there are significant improvements in the fitness values and accordingly, the total errors (S) in the modified RVGA. This study suggests that the modified algorithm is more suitable for the applications that need a higher amount of accuracy.

9 Discussion

In Table 2, the amount of S (total error) significantly decreases while the number of generations (iterations) is increasing from 10 to 100. Our experiments showed that the amount of S fluctuates around 0.0064 after 100 generations. Therefore, an optimal solution can be obtained in problem (1) with 100 generations while the noise of measurement is only on s (overspecified condition). Table 3 illustrates the results of the algorithm when the noise of measurement is on one of the initial or boundary conditions in addition to s. It is evident that, the amount of total error increases when the number of noisy resources increases. Therefore, the amount of total error varies between 0.0216 and 0.0627.

10 Conclusion

1. The presented modified genetic algorithm is a capable method to find unknown $a(t)$ in an IPP even with noisy data.

2. Results showed that an acceptable estimation can be obtained by the presented algorithm with 100 generations within a CPU with a clock speed of 2.4 GHz.

3. The presented algorithm is stable with respect to a small perturbation in the input data.

4. The execution time of the presented algorithm is acceptable to be used in real-world applications.

References

1. Baiocchi, C. *Sui problemi ai limiti per le equazioni paraboliche del tipo del calore*, (Italian) Boll. Un. Mat. Ital. 19(3) (1964), 407–422.

2. Bao, G., Ehlers T., and Li, P. *Radiogenic source identification for the helium production-diffusion equation*, Commun. Comput. Phys. 14 (2013), 1–20.

3. Bao, G., Ehlers T., Li, P., and Zheng, G. *A nonuniqueness result for an inverse diffusivity problem in thermochronometry*, Submitted.

4. Bao, G. and Xu, X. *An inverse diffusivity problem for the helium production-diffusion equation*, Inverse Problems, 28(8) (2012), 085002, 15 pp.

5. Beck, J.V., and Murio, D.C. *Combined function specification-regularization procedure for solution of inverse heat condition problem*, AIAA J., 24 (1986), 180–185.

6. Cabeza, J.M.G., Garcia J.A.M., and Rodriguez, A.C. *A sequential algorithm of inverse heat conduction problems using singular value decomposition*, J. Therm. Sci. 44 (2005), 235–244.

7. Cannon, J.R. *The one-dimensional heat equation*, With a foreword by Felix E. Browder. Encyclopedia of Mathematics and its Applications, 23. Addison-Wesley Publishing Company, Advanced Book Program, Reading, MA, 1984.

8. Demir, A., and Bayrak, M. A. *Inverse problem for determination of an unknown coefficient in the time fractional diffusion equation*, Communications in Mathematics and Applications, 9(2) (2018), 229–237.

9. Foadian, S., Pourgholi, R., Tabasi, S. H., and Zeidabadi H. *Solving an inverse problem for a generalized time-delayed Burgers-fisher equation by haar wavelet method*, J. Appl. Anal. Comput. 10(2) (2020) 391–410.

10. Friedman, A. *Partial differential equations parabolic type*, Prentice-Hall, Inc., Englewood Cliffs, N.J. 1964.

11. Holland, J.H. *Adaptation in natural and artificial system: An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, Ann Arbor, Mich., 1975.

12. John, F. *Partial differential equations*, Springer-Verlag, New York, 1982.

13. Kythe, P.K., Puri, P., and Schaferkotter, M.R. *Partial differential equations and mathematica*, CRC Press, Boca Raton, FL, 1997.

14. Lesnic, D., Yousefi, S.A., and Ivanchov, M. *Determination of a time-dependent diffusivity from nonlocal conditions*, J. Appl. Math. Comput. 41 (2013), 301–320.

15. Lions, J.C. and Magenes, E. *Remarques sur les problèmes aux limites pour opérateurs paraboliques.* Compt. Rend. Acad. Sci. (Paris) 251 (1960), 2118–2120.

16. Lions, J.C. and Magenes, E. *Problèmes aux limites non homogènes. II*, (French) Ann. Inst. Fourier (Grenoble) 11 (1961), 137–178.

17. Lions, J.C. and Magenes, E. *Problèmes aux limites non homogènes. IV*. (French) Ann. Scuola Norm. Sup. Pisa Cl. Sci. 15(3) (1961), 311–326.

18. Lions, J.C., and Magenes, E. *Problèmes aux limites non homogènes el applications*, Dunod, Paris,1968.

19. Ozisik, M.N. *Heat conduction*, John Wiley & Sons, Inc., New York, 1993.

20. Pourgholi, R., Azizi, N., Gasimov, Y.S., Aliev, F., and Khalafi, H.K. *Removal of numerical instability in the solution of an inverse heat conduction problem*, Commun. Nonlinear Sci. Numer. Simul. 14 (6) (2009), 2664–2669

21. Pourgholi, R., Foadian, S.A., and Esfahani, A. *Haar basis method to solve some inverse problems for two-dimensional parabolic and hyperbolic equations*, TWMS J. App. Eng. Math. 3 (2013), 10–32.

22. Zeidabadi, H., Pourgholi, R., and Tabasi, S.H. *Solving a nonlinear inverse system of Burgers equations*, Int. J. Nonlinear Anal. Appl. 10(1) (2019), 35–54.

23. Zeidabadi, H., Pourgholi R., and Tabasi S.H. *A hybrid scheme for time fractional inverse parabolic problem*, Waves Random Complex Media, 30(2) (2020), 354–368.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 55–72
DOI:10.22067/ijnao.2021.11327.0

Research Article

Approximate solution for a system of fractional integro-differential equations by Müntz Legendre wavelets

Y. Barazandeh

Abstract

We use the Müntz Legendre wavelets and operational matrix to solve a system of fractional integro-differential equations. In this method, the system of integro-differential equations shifts into the systems of the algebraic equation, which can be solved easily. Finally, some examples confirming the applicability, accuracy, and efficiency of the proposed method are given.

AMS subject classifications (2020): 45D05, 42C10, 65G99.

Keywords: System of fractional integro-differential equations; Caputo fractional derivative; Müntz Legendre method.

1 Introduction

The theory of fractional calculus and especially fractional differential equations has recently become a popular topic, and many natural phenomena are modeled by it in [26, 21, 28, 11, 17, 2, 3]. The fractional integro-differential equations (FIDEs) are generalized integro-differential equations. In general, it is very difficult to obtain an exact solution for most FIDEs, so the use of approximate methods seems necessary. Many studies have used approximate methods to solve FIDE problems, such as Adomian decomposition method [9, 18], variational iteration method [10, 29, 22], the generalized differential transform method [1, 23], the homotopy perturbation method [36, 27], the collocation method [12, 19, 14, 34, 16], and block-pulse functions (BPFs) method [4, 7, 32, 33].

One type of the numerical methods that has been used effectively for solving FIDEs is wavelets methods, which can be referred to in as Legendre

Received 28 July 2020; revised 17 October; accepted 4 November 2020
Yousef Barazandeh
Department of Mathematics, Lorestan University, Khorramabad, Iran.
e-mail: barazandeh.yu@fs.lu.ac.ir

wavelets [35], Chebyshev wavelets [37], Haar wavelets [31], Bernoulli wavelet [13, 30], and Müntz Legendre wavelets [5]. Consider the following nonlinear system of FIDEs, where $s, t \in [0, 1], 0 < p, q \leq 1$, and D^p and D^q represent Caputo derivative:

$$\begin{cases} D^p y_1(t) = f_{11}(t, y_1(t), y_2(t)) + \int_0^t f_{12}(s, y_1(s), y_2(s))ds, \\ D^q y_2(t) = f_{21}(t, y_1(t), y_2(t)) + \int_0^t f_{22}(s, y_1(s), y_2(s))ds. \end{cases} \tag{1}$$

In this article, Müntz Legendre wavelets and operational matrix have been used to obtain a numerical solution for relation (1).

The organization of the paper is as follows: In Section 2, some basic results from the fractional calculus and the definition of Müntz Legendre wavelets are given. In Section 3, a numerical method based on Müntz Legendre wavelets for an approximation system of FIDEs and its convergence analysis is presented. In Section 4, the mentioned method has been examined by some examples. Finally, the conclusion is given in Section 5.

2 Preliminaries

In this section, some basic results from the fractional calculus and Müntz polynomial are given.

2.1 Caputo derivative and Riemann–Liouville integral

Definition 1 (see [25]). The fractional derivative of $f(t)$ in Caputo sense for $p, t \in [0, 1]$ is defined as

$$D^p f(t) = \frac{1}{\Gamma(1-p)} \int_0^t (t-\tau)^{-p} f'(\tau) d\tau,$$

where $f(t)$ is an unknown function in an appropriate functional space and Γ is the gamma function.

Definition 2 (see [25]). The Riemann–Liouville fractional integral of order p can be defined as follows:

$$I^p f(t) = \frac{1}{\Gamma(p)} \int_0^t (t-s)^{p-1} f(s) ds.$$

Remark 1 (see [25]). The following relationships exist between Caputo derivative and Riemann–Liouville integral:

$$D^p I^p f(t) = f(t),$$

$$I^p D^p f(t) = f(t) - \sum_{m=0}^{n-1} \frac{f^{(m)}(0^+)}{m!} t^m, \qquad t \geq 0, \quad n-1 < p < n. \tag{2}$$

2.2 Müntz polynomial and Müntz Legendre wavelets

Theorem 1 (see [24]). The sequence $\{t^{\lambda_i}\}_{i=0}^{\infty}$,with $0 \leq \lambda_0 < \lambda_1 < \cdots \to \infty$ is fundamental in $L_2[0,1]$ if and only if

$$\sum_{k=1}^{\infty} {\lambda_k}^{-1} = \infty.$$

The classical Müntz polynomial is represented as

$$\sum_{k=0}^{N} a_k t^{\lambda_k},$$

where $a_k \in \mathbb{R}$.
The Müntz Legendre polynomial is an orthogonalized Müntz polynomial respect to the Lebesgue measure in $[0,1]$. Assume that $\Lambda_n = \{\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_n\}$ and that

$$Re(\lambda_k) > -\frac{1}{2} \ \ (k \in \mathbb{N}_0), \quad \lambda_k \neq \lambda_j \ (k \neq j).$$

Then we can represent Müntz Legendre polynomial as follows (see [8, 20]):

$$\mathcal{L}_n(t) = \sum_{k=0}^{n} c_{k,n} t^{\lambda_k}, \qquad c_{k,n} = \frac{\prod_{j=0}^{n-1}(\lambda_k + \lambda_j + 1)}{\prod_{j=0, j\neq k}^{n}(\lambda_k - \lambda_j)} \qquad (n \in \mathbb{N}_0).$$

In this paper, we set $\lambda_k = k\gamma$ and $\gamma = 1$.

Müntz Legendre wavelets are defined on $[0,1]$ as (see [5]):

$$\varphi_{nm}(t) = \begin{cases} \sqrt{2^{k-1}(1+2m\gamma)} + \mathcal{L}_m(2^{k-1}t - n - 1, \gamma), & \dfrac{n-1}{2^{k-1}} \leq t < \dfrac{n}{2^{k-1}}, \\ 0 & \text{otherwise,} \end{cases}$$

where $n = 1, 2, \ldots, 2^{k-1}$, $m = 0, 1, 2, \ldots, M-1 (k, M \in \mathbb{N})$, and $\mathcal{L}_m(t, \gamma)$ is the well-known Müntz Legendre of order m, that in which $\lambda_k = k\gamma$.

3 Approximation of function by using Müntz Legendre wavelets

Any function y belonging to $L_2[0,1]$ can be expanded by Müntz Legendre wavelets as follows (see [5]):

$$y(t) \simeq \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} a_{nm}\varphi_{nm}(t). \tag{3}$$

Assume that

$$\{\varphi_{1,0}(t), \ldots, \varphi_{1,M-1}(t), \varphi_{2,0}(t), \ldots, \varphi_{2,M-1}(t), \ldots, \\ \varphi_{2^{k-1},0}(t), \ldots, \varphi_{2^{k-1},M-1}(t)\}$$

is a set of Müntz Legendre wavelets and that

$$\mathrm{X} = \mathrm{span}\{\varphi_{1,0}(\mathrm{t}), \ldots, \varphi_{1,\mathrm{M}-1}(\mathrm{t}), \varphi_{2,0}(\mathrm{t}), \ldots, \varphi_{2,M-1}(t), \ldots, \\ \varphi_{2^{k-1},0}(t), \ldots, \varphi_{2^{k-1},M-1}(t)\}.$$

Since X is a finite-dimensional function space of $L_2[0,1]$, so y has the best unique approximation $y_{m'} \in \mathrm{X}$ such that $||y - y_{m'}|| \leq ||y - x||$ for all $x \in \mathrm{X}$. Moreover, there are unique coefficients a_{nm} such that

$$y(t) \simeq y_{m'} = \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} a_{nm}\varphi_{nm}(t) = A_{nm}^{T}\Phi_{nm}(t), \tag{4}$$

where

$$A_{nm} = \Big[a_{1,0}, \ldots, a_{1,M-1}, a_{2,0}, \ldots, a_{2,M-1}, \ldots, a_{2^{k-1},0}, \ldots, a_{2^{k-1},M-1}\Big]^T,$$

$$\Phi_{nm}(t) = \Big[\varphi_{1,0}(t), \ldots, \varphi_{1,M-1}(t), \varphi_{2,0}(t), \ldots, \varphi_{2,M-1}(t), \ldots \\ \varphi_{2^{k-1},0}(t), \ldots, \varphi_{2^{k-1},M-1}(t)\Big]^T.$$

Equation (4) can be rewritten as

$$y(t) \simeq y_{m'}(t) = \sum_{i=1}^{m'} a_i\varphi_i(t) = A_{m'}^{T}\Phi_{\mathrm{m}'}(\mathrm{t}), \tag{5}$$

where

$$A_{m'} = \Big[a_1, \ldots, a_M, a_{M+1}, \ldots, a_{2M}, \ldots a_{2^{k-1}(M+1)}, \ldots, a_{m'}\Big]^T,$$

$$\Phi_{m'} = \Big[\varphi_1(t), \dots, \varphi_M(t), \varphi_{M+1}(t), \dots, \varphi_{2M}(t), \dots \varphi_{2^{k-1}(M+1)}(t), \dots, \varphi_{m'}(t)\Big]^T,$$

$$a_i = a_{nm}, \qquad \varphi_i = \varphi_{nm}, \qquad i = (n-1)M + m + 1, \qquad m' = 2^{k-1}M.$$

Suppose that $t_i = \frac{2i-1}{2m'}$, $i = 1, 2, 3, \dots, m'$ are collocation points. We define the Müntz Legendre wavelets Ψ as follows:

$$\Psi_{m' \times m'} = \Big[\Phi(\frac{1}{2\mathrm{m}'}) \;\; \Phi(\frac{3}{2\mathrm{m}'}) \;\; \Phi(\frac{5}{2\mathrm{m}'}) \;\; \cdots \;\; \Phi(\frac{2\mathrm{m}'-1}{2\mathrm{m}'})\Big].$$

Theorem 2 (see [6]). Suppose that $y(t) \in \mathbb{C}^M[0,1]$. that $y_{m'}(t) \in \mathrm{X}$ is the approximation of $y(t)$ via wavelet, and that there exists $D \in \mathbb{N}$ such that $|y^{(M)}(t)| < D$ for all $t \in [0,1]$. Then

$$\|e_y\| = \|y(t) - y_{m'}(t)\| \le \frac{D}{M! 2^{M(k-1)}}.$$

Theorem 3. Suppose that in (1), f_{ij} $(i, j = 1, 2)$ satisfies the Lipschitz condition and $y_{i_{m'}}$ is an approximate solution of $y_i(t)$ by the Müntz Legendre wavelets method. If $k, M \to \infty$, then $\|e_i\| = \|y_i - y_{i_{m'}}\| \to 0$.

Proof. Let $r_i(t)$ $(i = 1, 2)$ be perturbation terms and let $\lceil p_i \rceil = m_i$. we can write $y_{i_{m'}}$ and $y_{i_{m'}} + e_i$ as follows:

$$\begin{aligned} y_{i_{m'}}(t) =& r_i(x) + \sum_{k=1}^{m_i - 1} \frac{{y_i}^{(k)}}{k!} t^k + \frac{1}{\Gamma(p_i)} \int_0^t (t-x)^{p_i - 1} f_{i1}(x, y_{1_{m'}}(x), y_{2_{m'}}(x)) dx \\ &+ \frac{1}{\Gamma(p_i)} \int_0^t \int_0^x (t-x)^{p_i-1} f_{i2}(s, y_{1_{m'}}(s), y_{2_{m'}}(s)) ds dx, \end{aligned} \tag{6}$$

$$\begin{aligned} y_{i_{m'}}(t) + e_i(t) =& \sum_{k=1}^{m_i-1} \frac{{y_i}^{(k)}}{k!} t^k \\ &+ \frac{1}{\Gamma(p_i)} \int_0^t (t-x)^{p_i-1} f_{i1}(x, y_{1_{m'}}(x) + e_1(x), y_{2_{m'}}(x)) + e_2(x) dx \\ &+ \frac{1}{\Gamma(p_i)} \int_0^t \int_0^x (t-x)^{p_i-1} f_{i2}(s, y_{1_{m'}}(s) \\ &\qquad + e_1(s), y_{2_{m'}}(s) + e_2(s)) ds dx. \end{aligned} \tag{7}$$

By subtracting (6) from (7) and employing the Lipschitz condition, we have

$$\begin{aligned} \|e_i\| \le& \|r_i\| + \frac{1}{\Gamma(p_i)} L_1(\|e_1\| + \|e_2\|) \int_0^t (t-x)^{p_i-1} dx \\ &+ \frac{1}{\Gamma(p_i)} L_1(\|e_1\| + \|e_2\|) \int_0^t \int_0^x (t-x)^{p_i-1} ds dx. \end{aligned} \tag{8}$$

By setting $\max\{L_1, L_2\} = L$, $|y_i^{(M)}(t)| < D_i (i = 1, 2)$, $D = \max\{D_1, D_2\}$ and using Theorem 1, we can write

$$\begin{aligned}\|e_i\| &\le \|r_i\| + L(\|e_1\| + \|e_2\|)(\frac{1}{\Gamma(p_i+1)} + \frac{1}{\Gamma(p_i+2)}) \\ &\le \|r_i\| LD \frac{1}{M!2^{M(k-1)}}(\frac{1}{\Gamma(p_i+1)} + \frac{1}{\Gamma(p_i+2)}).\end{aligned} \tag{9}$$

Since $k, M \to \infty$, then $\|r_i\| \to 0$. We can conclude that $\|e_i\| \to 0$ $(i = 1, 2)$. □

3.1 Operational matrix of the fractional integration

In this section, we review the fractional-order operational matrix of integration related to the Müntz Legendre wavelets. For this purpose, the required definitions and lemmas are given from [30].

Definition 3. For $m' = 2^{k-1}M$ and $i = 1, 2, \ldots, m'$, the set of BPFs is defined as

$$\hat{b}_i(t) = \begin{cases} 1, & \frac{(i-1)}{m'} \le t < \frac{i}{m'}, \\ 0 & \text{otherwise.} \end{cases}$$

We use the following properties of BPFs:

$$\hat{b}_i(x)\hat{b}_j(x) = \begin{cases} \hat{b}_i(x), & i = j, \\ 0, & i \ne j, \end{cases}$$

$$\int_0^x \hat{b}_i(x)\hat{b}_j(x)dx = \begin{cases} \frac{1}{m'}, & i = j, \\ 0, & i \ne j. \end{cases}$$

Definition 4. Suppose that $U = [u_1, u_2, \ldots, u_{m'}]^T$ and
$V = [v_1, v_2, \ldots, v_{m'}]^T$. We define

$$U \otimes D = [u_1v_1, u_2v_2, \ldots, u_{m'}v_{m'}]^T.$$

Lemma 1. Let $\hat{B}_{m'} = [\hat{b_1}, \hat{b_2}, \ldots, \hat{b_{m'}}]^T$, and suppose that $f(t), g(t) \in L_2[0, 1]$ can be written as $f(t) = F^T\hat{B}_{m'}(t)$ and $g(t) = G^T\hat{B}_{m'}(t)(F^T = [f_1, f_2, \ldots, f_{m'}], G^T = [g_1, g_2, \ldots, g_{m'}])$. Then

$$f(x)g(x) \approx F^T\hat{B}_{m'}(t)G^T\hat{B}_{m'}(t) = (F^T \otimes G^T)\hat{B}_{m'}(t), \tag{10}$$

$$f(x)^2 \approx (F^T\hat{B}_{m'}(t))^2 = (F^T)^2\hat{B}_{m'}(t). \tag{11}$$

The Riemann–Liouville fractional integration of order α of BPFs can be presented as (see [15]):

$$I^p \hat{B}_{m'}(t) \approx \mathcal{F}^p \hat{B}_{m'}(t), \tag{12}$$

where $\mathcal{F}^p$ is the BPFs operational matrix with

$$\mathcal{F}^p = (\frac{1}{m'})^p \frac{1}{\Gamma(p+2)} \begin{bmatrix} 1 & f_1 & f_2 & \cdots & f_{m'-1} \\ 0 & 1 & f_1 & \cdots & f_{m'-2} \\ 0 & 0 & 1 & \cdots & f_{m'-3} \\ 0 & 0 & 0 & \cdots & f_{m'-4} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix},$$

in which $f_k = (k+1)^{p+1} - 2k^{p+1} + (k-1)^{p+1}$, $k = 1, 2, \ldots, m'-1$.
We now derive the operational matrix of fractional integration of Müntz Legendre wavelets. The integration of Müntz Legendre wavelets $\Phi_{m'}(x)$ can be expressed as

$$I\Phi_{m'}(t) = \int_0^t \Phi_{m'}(s)ds \approx \mathcal{P}_{m'}\Phi_{m'}(t), \tag{13}$$

where the m'-square matrix $\mathcal{P}_{m'}$ is called Müntz Legendre wavelets operational matrix. Also

$$I^p\Phi_{m'}(t) \approx \mathcal{P}^p_{m'}\Phi_{m'}(t), \tag{14}$$

where $\mathcal{P}^p_{m'}$ is called Müntz Legendre wavelets fractional integral operational matrix.
The Müntz Legendre wavelets can be expanded into m'-set BPFs as

$$\Phi_{m'}(t) \approx \Psi_{m'}\hat{B}_{m'}(t). \tag{15}$$

Considering (12), we can write (14) and (15) as

$$\begin{aligned} &I^p\Phi_{m'}(t) \approx I^p\Psi_{m'}\hat{B}_{m'}(t) = \Psi_{m'}I^p\hat{B}_{m'}(t) \approx \Psi_{m'}\mathcal{F}^p\hat{B}_{m'}(t), \\ &\mathcal{P}^p_{m'}\Phi_{m'}(t) \approx I^p\Phi_{m'}(t) \approx \Psi_{m'}\mathcal{F}^p\hat{B}_{m'}(t) \approx \Psi_{m'}\mathcal{F}^p\Psi^{-1}_{m'}\Phi_{m'}(t). \end{aligned}$$

Finally, we conclude

$$\mathcal{P}^p_{m'} \approx \Psi_{m'}\mathcal{F}^p\Psi^{-1}_{m'}.$$

4 Examples

In this section, some examples are solved by using the proposed method with different parameters.

Example 1. Consider

$$\begin{cases} D^{\frac{1}{2}}y_1(t) = \Gamma(\frac{3}{2}) + \int_0^t [y_1(s) + y_2(s)]ds, \\ D^{\frac{1}{2}}y_2(t) = y_1(t) + y_2(t) - \Gamma(\frac{3}{2}), \end{cases} \tag{16}$$

where $y_1(0) = 0, y_2(0) = 0$.

Exact solutions for system (16) are $y_1(t) = \sqrt{t}$ and $y_2(t) = -\sqrt{t}$. Indeed

$$D^{\frac{1}{2}}y_1(t) \approx G_{m'}^T \Phi_{m'}(t), \quad D^{\frac{1}{2}}y_2(t) \approx H_{m'}^T \Phi_{m'}(t), \tag{17}$$

where

$$G_{m'}^T = [g_1, g_2, g_3, \dots, g_{m'}], \quad H_{m'}^T = [h_1, h_2, h_3, \dots, h_{m'}].$$

By using the initial conditions and (2), (14), (15), and (17), we have

$$\begin{cases} y_1(t) = I^{\frac{1}{2}}D^{\frac{1}{2}}y_1(t) + y_1(0) \approx G_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Phi_{m'}(t) \approx G_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Psi_{m'} \hat{B}_{m'}(t), \\ y_2(t) = I^{\frac{1}{2}}D^{\frac{1}{2}}y_2(t) + y_2(0) \approx H_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Phi_{m'}(t) \approx H_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Psi_{m'} \hat{B}_{m'}(t). \end{cases} \tag{18}$$

Then, from (13) and (18), we obtain

$$\begin{aligned} \int_0^t y_1(s)ds &\approx \int_0^t G_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Phi_{m'}(s)ds = G_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \int_0^t \Phi_{m'}(s)ds \\ &\approx G_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \mathcal{P}_{m'}^{1} \Phi_{m'}(t) \approx G_{m'}^T \mathcal{P}_{m'}^{1+\frac{1}{2}} \Psi_{m'} \hat{B}_{m'}(t), \end{aligned} \tag{19}$$

and similarly

$$\int_0^t y_2(s)ds \approx H_{m'}^T \mathcal{P}_{m'}^{1+\frac{1}{2}} \Psi_{m'} \hat{B}_{m'}(t). \tag{20}$$

By replacing (17)–(20) into (16), we obtain

$$\begin{cases} G_{m'}^T \Psi_{m'} \hat{B}_{m'}(t) \\ = \Gamma(\frac{3}{2})\big[1, 1, \dots, 1\big]_{1\times m'} \hat{B}_{m'}(t) + (G_{m'}^T \mathcal{P}_{m'}^{1+\frac{1}{2}} \Psi_{m'} + H_{m'}^T \mathcal{P}_{m'}^{1+\frac{1}{2}} \Psi_{m'}) \hat{B}_{m'}(t), \\ H_{m'}^T \Psi_{m'} \hat{B}_{m'}(t) \\ = (G_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Psi_{m'} + H_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Psi_{m'}) \hat{B}_{m'}(t) - \Gamma(\frac{3}{2})\big[1, 1, \dots, 1\big]_{1\times m'} \hat{B}_{m'}(t). \end{cases} \tag{21}$$

Relation (21) can be rewritten as follows:

$$\begin{cases} G_{m'}^T \Psi_{m'} = \Gamma(\frac{3}{2})\big[1, 1, \dots, 1\big]_{1\times m'} + G_{m'}^T \mathcal{P}_{m'}^{1+\frac{1}{2}} \Psi_{m'} + H_{m'}^T \mathcal{P}_{m'}^{1+\frac{1}{2}} \Psi_{m'}, \\ H_{m'}^T \Psi_{m'} = G_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Psi_{m'} + H_{m'}^T \mathcal{P}_{m'}^{\frac{1}{2}} \Psi_{m'} - \Gamma(\frac{3}{2})\big[1, 1, \dots, 1\big]_{1\times m'}. \end{cases} \tag{22}$$

By solving the nonlinear system (22), we can obtain the matrices of coefficients G and H.

In Figure 1, the exact and approximate solutions using the proposed method ($k = 6, M = 2$) are plotted and the numerical values are given in Table 1.

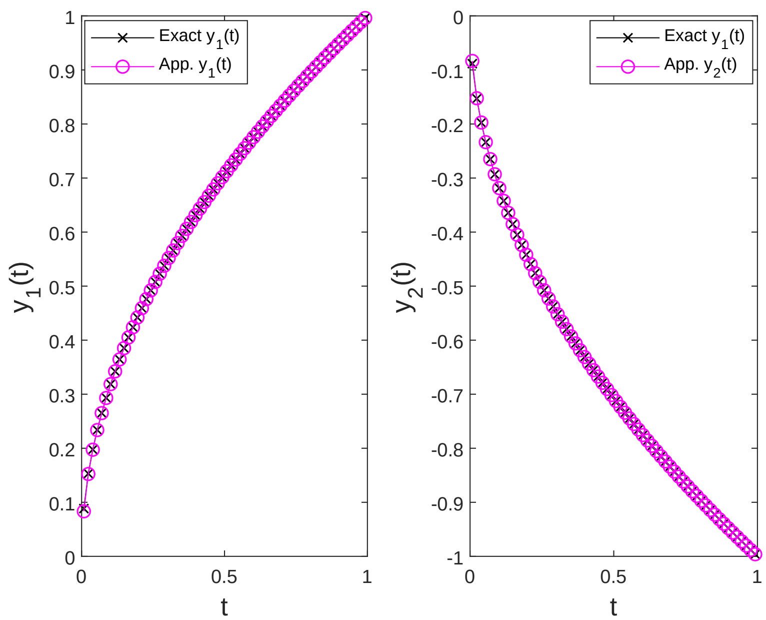


Figure 1: Exact and approximate solution of $y_1(t)$ and $y_2(t)$ in Example 1.

Table 1: Exact and approximate solution and absolute error (AE) for $y_1(t)$ and $y_2(t)$ in Example 1.

t	Ex. $y_1(t)$	Ap. $y_1(t)$	AE $y_1(t)$	Ex. $y_2(t)$	Ap. $y_2(t)$	AE $y_2(t)$
2.3438e-02	1.5309e-01	1.5237e-01	7.2418e-04	-1.5309e-01	-1.5237e-01	7.2418e-04
1.0156e-01	3.1869e-01	3.1861e-01	7.8660e-05	-3.1869e-01	-3.1861e-01	7.8660e-05
1.7969e-01	4.2390e-01	4.2386e-01	3.3400e-05	-4.2390e-01	-4.2386e-01	3.3400e-05
2.5781e-01	5.0775e-01	5.0773e-01	1.9431e-05	-5.0775e-01	-5.0773e-01	1.9431e-05
3.3594e-01	5.7960e-01	5.7959e-01	1.3062e-05	-5.7960e-01	-5.7959e-01	1.3062e-05
4.1406e-01	6.4348e-01	6.4347e-01	9.5455e-06	-6.4348e-01	-6.4347e-01	9.5455e-06
4.9219e-01	7.0156e-01	7.0155e-01	7.3654e-06	-7.0156e-01	-7.0155e-01	7.3654e-06
5.7031e-01	7.5519e-01	7.5518e-01	5.9049e-06	-7.5519e-01	-7.5518e-01	5.9049e-06
6.4844e-01	8.0526e-01	8.0525e-01	4.8706e-06	-8.0526e-01	-8.0525e-01	4.8706e-06
7.2656e-01	8.5239e-01	8.5238e-01	4.1065e-06	-8.5239e-01	-8.5238e-01	4.1065e-06
8.0469e-01	8.9704e-01	8.9704e-01	3.5232e-06	-8.9704e-01	-8.9704e-01	3.5232e-06
8.8281e-01	9.3958e-01	9.3958e-01	3.0660e-06	-9.3958e-01	-9.3958e-01	3.0660e-06
9.7656e-01	9.8821e-01	9.8821e-01	2.6353e-06	-9.8821e-01	-9.8821e-01	2.6353e-06

Example 2. Consider

$$\begin{cases} D^p y_1(t) = \frac{1}{3} y_1(t) y_2(t) + \frac{1}{2} {y_2}^2(t) + 2y_2(t) - \int_0^t [y_1(s) + y_2(s)] ds, \\ D^q y_2(t) = \frac{1}{3} y_1(t) y_2(t) - y_1(t) + 1 - \int_0^t [y_1(s) - 2y_2(s)] ds, \end{cases} \tag{23}$$

where $y_1(0) = 0, y_2(0) = 0$, and $0 < p, q \leq 1$.

Exact solutions for the above system when $p = q = 1$ are $y_1(t) = t^2$ and $y_2(t) = t$. The exact solutions of $y_1(t)$ and $y_2(t)$ for $p, q \in (0, 1)$ are unknown. Let

$$D^p y_1(t) \approx U_{m'}^T \Phi_{m'}(t), \quad D^q y_2(t) \approx V_{m'}^T \Phi_{m'}(t), \tag{24}$$

where

$$U_{m'}^T = [u_1, u_2, u_3, \ldots, u_{m'}], \quad V_{m'}^T = [v_1, v_2, v_3, \ldots, v_{m'}].$$

By using the initial conditions and (2), (14), (15), and (24), we have

$$\begin{cases} y_1(t) = I^p D^p y_1(t) + y_1(0) \approx U_{m'}^T \mathcal{P}_{m'}^p \Phi_{m'}(t) \approx U_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \hat{B}_{m'}(t), \\ y_2(t) = I^q D^q y_2(t) + y_2(0) \approx V_{m'}^T \mathcal{P}_{m'}^q \Phi_{m'}(t) \approx V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} \hat{B}_{m'}(t). \end{cases} \tag{25}$$

Then, from (10), (11), (13), and (25), we obtain

$$\begin{aligned} y_1(t) y_2(t) &\approx (U_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \hat{B}_{m'}(t))(V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} \hat{B}_{m'}(t)) \\ &= (U_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}) \hat{B}_{m'}(t), \end{aligned} \tag{26}$$

$$y_2^2(t) \approx \left(V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} \hat{B}_{m'}(t)\right)^2 = \left(V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}\right)^2 \hat{B}_{m'}(t), \tag{27}$$

$$\begin{aligned} \int_0^t y_1(s) ds &\approx \int_0^t U_{m'}^T \mathcal{P}_{m'}^p \Phi_{m'}(s) ds = U_{m'}^T \mathcal{P}_{m'}^p \int_0^t \Phi_{m'}(s) ds \\ &\approx U_{m'}^T \mathcal{P}_{m'}^p \mathcal{P}_{m'}^1 \Phi_{m'}(t) \approx U_{m'}^T \mathcal{P}_{m'}^{1+p} \Psi_{m'} \hat{B}_{m'}(t), \end{aligned} \tag{28}$$

and in the same way

$$\int_0^t y_2(s) ds \approx V_{m'}^T \mathcal{P}_{m'}^{1+q} \Psi_{m'} \hat{B}_{m'}(t). \tag{29}$$

By replacing (24) and (26)–(29) into (23), we obtain

$$\begin{cases} U_{m'}^T \Psi_{m'} \hat{B}_{m'}(t) = & \frac{1}{3} (U_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}) \hat{B}_{m'}(t) \\ & + \frac{1}{2} \left(V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}\right)^2 \hat{B}_{m'}(t) + 2 V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} \hat{B}_{m'}(t) \\ & - U_{m'}^T \mathcal{P}_{m'}^{1+p} \Psi_{m'} \hat{B}_{m'}(t) - V_{m'}^T \mathcal{P}_{m'}^{1+q} \Psi_{m'} \hat{B}_{m'}(t), \\ & \\ V_{m'}^T \Psi_{m'} \hat{B}_{m'}(t) = & \frac{1}{3} (U_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes V_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}) \hat{B}_{m'}(t) \\ & - U_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \hat{B}_{m'}(t) + \left[1, 1, \ldots, 1\right]_{1 \times m'} \hat{B}_{m'}(t) \\ & - U_{m'}^T \mathcal{P}_{m'}^{1+p} \Psi_{m'} \hat{B}_{m'}(t) + 2 V_{m'}^T \mathcal{P}_{m'}^{1+q} \Psi_{m'} \hat{B}_{m'}(t). \end{cases} \tag{30}$$

Relation (30) can be rewritten as follows:

$$\begin{cases} U_{m'}^T\Psi_{m'} = & \frac{1}{3}(U_{m'}^T\mathcal{P}_{m'}^p\Psi_{m'}\otimes V_{m'}^T\mathcal{P}_{m'}^q\Psi_{m'}) + \frac{1}{2}(V_{m'}^T\mathcal{P}_{m'}^q\Psi_{m'})^2 \\ & +2V_{m'}^T\mathcal{P}_{m'}^q\Psi_{m'} - U_{m'}^T\mathcal{P}_{m'}^{1+p}\Psi_{m'}\hat{B}_{m'}(t) - V_{m'}^T\mathcal{P}_{m'}^{1+q}\Psi_{m'}, \\ V_{m'}^T\Psi_{m'} = & \frac{1}{3}(U_{m'}^T\mathcal{P}_{m'}^p\Psi_{m'}\otimes V_{m'}^T\mathcal{P}_{m'}^q\Psi_{m'}) - U_{m'}^T\mathcal{P}_{m'}^p\Psi_{m'}\hat{B}_{m'}(t) \\ & +\left[1,1,\ldots,1\right]_{1\times m'} - U_{m'}^T\mathcal{P}_{m'}^{1+p}\Psi_{m'} + 2V_{m'}^T\mathcal{P}_{m'}^{1+q}\Psi_{m'}. \end{cases} \tag{31}$$

By solving the nonlinear system (31), we can obtain the matrices of coefficients U and V. In Figure 2, the exact solutions for $p = q = 1$ and approximate solutions for different values of p and q are plotted. In Figure 2, the approximate values are obtained by using the proposed method ($k = 5, M = 3$), and also the numerical values are given in Tables 2 and 3.

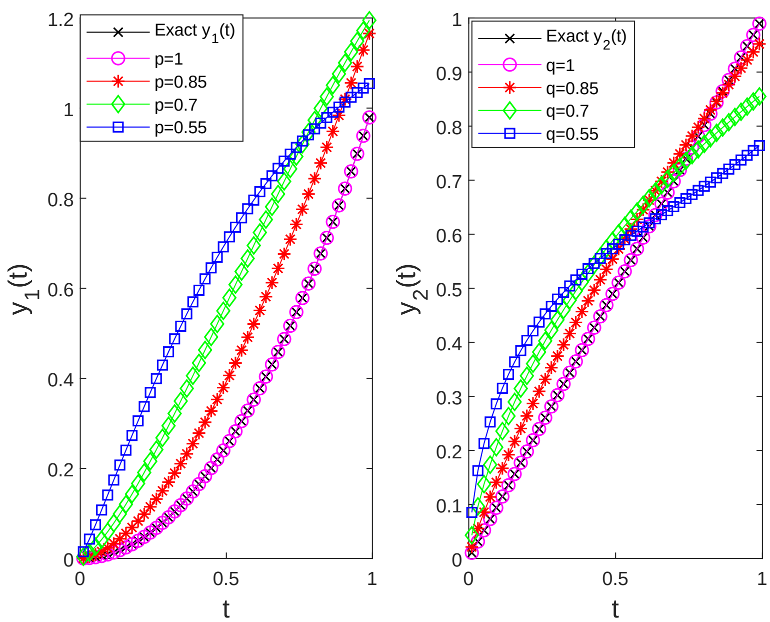


Figure 2: Numerical solution of $y_1(t)$ and $y_2(t)$ for different values p and q in Example 1.

Example 3. Consider

$$\begin{cases} D^p y_1(t) = y_1^2(t) + y_2^2(t) - \int_0^t y_1(s)ds, \\ D^q y_2(t) = -\frac{1}{2}y_2^2(t) - y_1(t) + \frac{1}{2} + \int_0^t y_1(s)y_2(s)ds, \end{cases} \tag{32}$$

where $y_1(0) = 0, y_2(0) = 1$ and $0 < p, q \leq 1$.

Table 2: Exact and approximate solution and absolute error (AE) for $y_1(t)$ in Example 1.

t	Ex.$[p=1]$	Ap.$[p=1]$	AE$[p=1]$	Ap.$[p=0.85]$	Ap.$[p=0.7]$	Ap.$[p=0.55]$
1.0417e-02	1.0851e-04	2.1680e-04	1.0829e-04	9.0701e-04	3.7273e-03	1.4986e-02
9.3750e-02	8.7891e-03	8.8949e-03	1.0582e-04	2.3452e-02	5.9316e-02	1.4077e-01
1.7708e-01	3.1359e-02	3.1460e-02	1.0172e-04	6.8756e-02	1.4291e-01	2.7310e-01
2.6042e-01	6.7817e-02	6.7913e-02	9.5890e-05	1.3193e-01	2.4150e-01	3.9923e-01
3.4375e-01	1.1816e-01	1.1825e-01	8.8208e-05	2.1044e-01	3.4943e-01	5.1564e-01
4.2708e-01	1.8240e-01	1.8248e-01	7.8552e-05	3.0242e-01	4.6286e-01	6.2076e-01
5.1042e-01	2.6053e-01	2.6059e-01	6.6793e-05	4.0627e-01	5.7884e-01	7.1406e-01
5.9375e-01	3.5254e-01	3.5259e-01	5.2791e-05	5.2059e-01	6.9495e-01	7.9567e-01
6.7708e-01	4.5844e-01	4.5848e-01	3.6398e-05	6.4402e-01	8.0916e-01	8.6627e-01
7.6042e-01	5.7823e-01	5.7825e-01	1.7458e-05	7.7524e-01	9.1976e-01	9.2690e-01
8.4375e-01	7.1191e-01	7.1191e-01	4.1961e-06	9.1295e-01	1.0253e+00	9.7893e-01
9.2708e-01	8.5948e-01	8.5945e-01	2.8742e-05	1.0558e+00	1.1248e+00	1.0240e+00
9.8958e-01	9.7928e-01	9.7923e-01	4.9161e-05	1.1656e+00	1.1950e+00	1.0543e+00

Table 3: Exact and approximate solution and absolute error (AE) for $y_2(t)$ in Example 1.

t	Ex.$[q=1]$	Ap.$[q=1]$	AE$[q=1]$	Ap.$[q=0.85]$	Ap.$[q=0.7]$	Ap.$[q=0.55]$
1.0417e-02	1.0417e-02	1.0416e-02	7.5926e-07	2.1272e-02	4.2951e-02	8.5180e-02
9.3750e-02	9.3750e-02	9.3743e-02	7.0114e-06	1.4078e-01	2.0576e-01	2.8594e-01
1.7708e-01	1.7708e-01	1.7707e-01	1.3592e-05	2.4035e-01	3.1438e-01	3.8446e-01
2.6042e-01	2.6042e-01	2.6040e-01	2.0462e-05	3.3126e-01	4.0256e-01	4.5249e-01
3.4375e-01	3.4375e-01	3.4372e-01	2.7585e-05	4.1619e-01	4.7775e-01	5.0415e-01
4.2708e-01	4.2708e-01	4.2705e-01	3.4929e-05	4.9638e-01	5.4348e-01	5.4588e-01
5.1042e-01	5.1042e-01	5.1037e-01	4.2466e-05	5.7259e-01	6.0188e-01	5.8140e-01
5.9375e-01	5.9375e-01	5.9370e-01	5.0169e-05	6.4530e-01	6.5441e-01	6.1329e-01
6.7708e-01	6.7708e-01	6.7703e-01	5.8012e-05	7.1488e-01	7.0223e-01	6.4344e-01
7.6042e-01	7.6042e-01	7.6035e-01	6.5975e-05	7.8160e-01	7.4628e-01	6.7336e-01
8.4375e-01	8.4375e-01	8.4368e-01	7.4034e-05	8.4566e-01	7.8743e-01	7.0429e-01
9.2708e-01	9.2708e-01	9.2700e-01	8.2170e-05	9.0727e-01	8.2650e-01	7.3726e-01
9.8958e-01	9.8958e-01	9.8950e-01	8.8310e-05	9.5195e-01	8.5492e-01	7.6385e-01

Exact solutions for the above system when $p = q = 1$ are $y_1(t) = \sin(t)$ and $y_2(t) = \cos(t)$. The exact solutions of $y_1(t)$ and $y_2(t)$ for $p, q \in (0, 1)$ are unknown.
Let

$$D^p y_1(t) \approx W^T_{m'} \Phi_{m'}(t), \quad D^q y_2(t) \approx R^T_{m'} \Phi_{m'}(t), \tag{33}$$

where

$$W^T_{m'} = [w_1, w_2, w_3, \ldots, w_{m'}], \quad R^T_{m'} = [r_1, r_2, r_3, \ldots, r_{m'}].$$

By using the initial conditions and (2), (14), (15), and (33), we have

$$\begin{cases} y_1(t) = I^p D^p y_1(t) + y_1(0) \approx W^T_{m'} \mathcal{P}^p_{m'} \Phi_{m'}(t) \approx W^T_{m'} \mathcal{P}^p_{m'} \Psi_{m'} \hat{B}_{m'}(t), \\ y_2(t) = I^q D^q y_2(t) + y_2(0) \approx R^T_{m'} \mathcal{P}^q_{m'} \Phi_{m'}(t) + 1 \approx R^T_{m'} \mathcal{P}^q_{m'} \Psi_{m'} \hat{B}_{m'}(t) + 1, \end{cases} \tag{34}$$

Then, from (10), (11), (13), and (34), we obtain

$$
\begin{aligned}
y_1(t)y_2(t) &\approx (W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \hat{B}_{m'}(t))(R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} \hat{B}_{m'}(t) + 1) \\
&= (W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}) \hat{B}_{m'}(t) + W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \hat{B}_{m'}(t),
\end{aligned} \tag{35}
$$

$$
\begin{aligned}
y_1^2(t) &\approx (W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'})^2 \hat{B}_{m'}(t), \\
y_2^2(t) &\approx (R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'})^2 \hat{B}_{m'}(t) + 2R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} \hat{B}_{m'}(t) + 1,
\end{aligned} \tag{36}
$$

$$
\int_0^t y_1(s)ds \approx \int_0^t W_{m'}^T \mathcal{P}_{m'}^p \Phi_{m'}(s)ds \approx W_{m'}^T \mathcal{P}_{m'}^{1+p} \Psi_{m'} \hat{B}_{m'}(t), \tag{37}
$$

$$
\begin{aligned}
\int_0^t y_1(s)y_2(s)ds &\approx \int_0^t (W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}) \hat{B}_{m'}(s)ds \\
&\quad + \int_0^t W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'}(s) \hat{B}_{m'}(s)ds \\
&= (W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}) \int_0^t \hat{B}_{m'}(s)ds \\
&\quad + W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \int_0^t \hat{B}_{m'}(s)ds \\
&\approx (W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}) \int_0^t \Psi_{m'}^{-1} \Phi_{m'}(s)ds \\
&\quad + W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \int_0^t \Psi_{m'}^{-1} \Phi_{m'}(s)ds \\
&\approx (W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'}) \Psi_{m'}^{-1} \mathcal{P}_{m'} \Psi_{m'} \hat{B}_{m'}(t) \\
&\quad + W_{m'}^T \mathcal{P}_{m'}^{1+p} \Psi_{m'} \hat{B}_{m'}(t).
\end{aligned} \tag{38}
$$

By replacing (33)–(38) into (32), we obtain

$$
\begin{cases}
W_{m'}^T \Psi_{m'} = & (W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'})^2 + (R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'})^2 \\
& +2R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} - W_{m'}^T \mathcal{P}_{m'}^{1+p} \Psi_{m'} + \big[1, 1, \ldots, 1\big]_{1 \times m'}, \\
\\
R_{m'}^T \Psi_{m'} = & -\frac{1}{2}(R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'})^2 - R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} - W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \\
& -(W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'} \otimes R_{m'}^T \mathcal{P}_{m'}^q \Psi_{m'} + W_{m'}^T \mathcal{P}_{m'}^p \Psi_{m'}) \Psi_{m'}^{-1} \mathcal{P}_{m'} \Psi_{m'}.
\end{cases} \tag{39}
$$

By solving the nonlinear system (39), we can obtain the matrices of coefficients W and R. In Figure 3, the exact solutions for $p = q = 1$ and approximate solutions for different values of p and q are plotted. In Figure 3, the approximate values are obtained by using the proposed method ($k = 4, M = 6$), and also the numerical values are given in Tables 4 and 5.

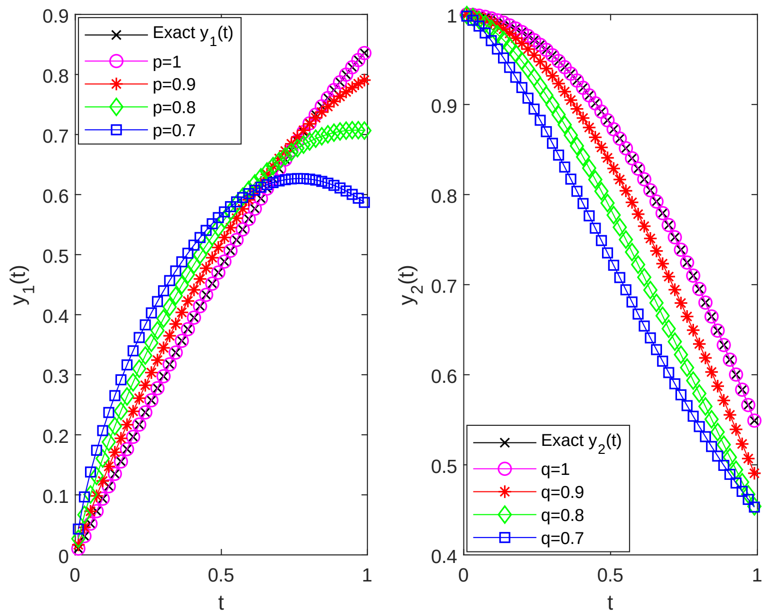


Figure 3: Numerical solution of $y_1(t), y_2(t)$ for different values p and q in Example 2.

Table 4: Exact and approximate solution and absolute error (AE) for $y_1(t)$ in Example 2.

t	Ex.$[p=1]$	Ap.$[p=1]$	AE $[p=1]$	Ap.$[p=0.9]$	Ap.$[p=0.8]$	Ap.$[q=0.7]$
1.0417e-02	1.0416e-02	1.0415e-02	1.6952e-06	1.6784e-02	2.6928e-02	4.2987e-02
9.3750e-02	9.3613e-02	9.3597e-02	1.5248e-05	1.2310e-01	1.6051e-01	2.0698e-01
1.7708e-01	1.7616e-01	1.7613e-01	2.8760e-05	2.1677e-01	2.6388e-01	3.1639e-01
2.6042e-01	2.5748e-01	2.5744e-01	4.2191e-05	3.0362e-01	3.5318e-01	4.0297e-01
3.4375e-01	3.3702e-01	3.3696e-01	5.5500e-05	3.8452e-01	4.3146e-01	4.7285e-01
4.2708e-01	4.1422e-01	4.1415e-01	6.8655e-05	4.5957e-01	4.9979e-01	5.2851e-01
5.1042e-01	4.8854e-01	4.8846e-01	8.1641e-05	5.2857e-01	5.5854e-01	5.7118e-01
5.9375e-01	5.5947e-01	5.5938e-01	9.4461e-05	5.9123e-01	6.0778e-01	6.0154e-01
6.7708e-01	6.2652e-01	6.2642e-01	1.0715e-04	6.4720e-01	6.4744e-01	6.1997e-01
7.6042e-01	6.8922e-01	6.8910e-01	1.1978e-04	6.9609e-01	6.7729e-01	6.2671e-01
8.4375e-01	7.4714e-01	7.4701e-01	1.3249e-04	7.3749e-01	6.9709e-01	6.2197e-01
9.2708e-01	7.9987e-01	7.9973e-01	1.4544e-04	7.7098e-01	7.0653e-01	6.0602e-01
9.8958e-01	8.3580e-01	8.3564e-01	1.5548e-04	7.9065e-01	7.0661e-01	5.8697e-01

Table 5: Exact and approximate solution and absolute error (AE) for $y_2(t)$ in Example 2.

t	Ex.$[q=1]$	Ap.$[q=1]$	AE$[q=1]$	Ap.$[q=0.85]$	Ap.$[q=0.7]$	Ap.$[q=0.55]$
1.0417e-02	9.9995e-01	9.9989e-01	5.4234e-05	9.9972e-01	9.9927e-01	9.9815e-01
9.3750e-02	9.9561e-01	9.9556e-01	5.3356e-05	9.9152e-01	9.8413e-01	9.7089e-01
1.7708e-01	9.8436e-01	9.8431e-01	5.1034e-05	9.7364e-01	9.5657e-01	9.3023e-01
2.6042e-01	9.6628e-01	9.6624e-01	4.7223e-05	9.4758e-01	9.2039e-01	8.8251e-01
3.4375e-01	9.4150e-01	9.4146e-01	4.1891e-05	9.1427e-01	8.7751e-01	8.3059e-01
4.2708e-01	9.1018e-01	9.1014e-01	3.5016e-05	8.7448e-01	8.2939e-01	7.7641e-01
5.1042e-01	8.7254e-01	8.7251e-01	2.6593e-05	8.2892e-01	7.7726e-01	7.2155e-01
5.9375e-01	8.2885e-01	8.2883e-01	1.6622e-05	7.7829e-01	7.2227e-01	6.6736e-01
6.7708e-01	7.7940e-01	7.7940e-01	5.1134e-06	7.2329e-01	6.6547e-01	6.1505e-01
7.6042e-01	7.2455e-01	7.2456e-01	7.9178e-06	6.6465e-01	6.0790e-01	5.6571e-01
8.4375e-01	6.6467e-01	6.6469e-01	2.2457e-05	6.0308e-01	5.5059e-01	5.2032e-01
9.2708e-01	6.0017e-01	6.0021e-01	3.8493e-05	5.3936e-01	4.9452e-01	4.7980e-01
9.8958e-01	5.4904e-01	5.4909e-01	5.1503e-05	4.9063e-01	4.5390e-01	4.5307e-01

5 Conclusion

In this paper, we introduced the Müntz Legendre wavelets to approximate the solution of a system of FIDEs and examined it by some numerical examples. Calculated absolute errors for different values of k and M indicated that by increasing k and M, the absolute errors decrease. The algorithm presented here can be easily used for different types of FIDEs.

References

1. Arikoglu, A. and Ozkol, I. *Solution of fractional integro-differential equations by using fractional differential transform method*, Chaos Solitons Fractals 40 (2009), 521–529.

2. Aslefallah, M. and Rostamy, D. *A numerical scheme for solving space-fractional equation by finite differences theta-method*, Int. J. Adv. Appl. Math. Mech. 1 (4) (2014), 1–9.

3. Aslefallah, M., Rostamy, D., and Hosseinkhani, K. *Solving time-fractional differential diffusion equation by theta-method*, Int. J. Adv. Appl. Math. Mech. 2 (1) (2014), 1–8.

4. Babolian, E., Maleknejad, K., and Mordad, M. *A numerical method for solving Fredholm-Volterra integral equations in two-dimensional spaces using block pulse functions and an operational matrix*, J. Comput. Appl. Math. 235 (14) (2011), 3965–3971.

5. Bahmanpour, M., Tavassoli Kajani, M., and Maleki, M. *Solving Fredholm integral equations of the first kind using Müntz wavelets*, Appl. Numer. Math. 143 (2019), 159–171.

6. Biazar, J., and Ebrahimi, H. *Chebyshev wavelets approach for non-linear systems of Volterra integral equations*, Comput. Math. with Appl. 63 (2012), 608–616.

7. Ebadian, A. and Khajehnasiri, A.A. *Block-pulse functions and their applications to solving systems of higher-order nonlinear Volterra integro-differential equations*, Electron. J. Differ. Equ. 54 (2014), 1–9.

8. Esmaeili, S., Shamsi, M., and Luchko, Y. *Numerical solution of fractional differential equations with a collocation method based on Müntz polynomials*, Comput. Math. Appl. 62(3) (2011), 918–929.

9. Hashim, I. *Adomian decomposition method for solving BVPs for fourth-order integro-differential equations*, J. Comput. Appl. Math. 193 (2006), 658–64.

10. He, J.H. *Approximation analytical solution for seepage flow with fractional derivatives in porous media*, Comput. Methods Appl. Mech. Eng. 167 (1988), 57–68.

11. Hilfer, R. *Applications of fractional calculus in physics*, World Scientific, Singapore, 2000.

12. Huang, L., Li, X.F., and Zhao, Y.L. *Approximate solution of fractional integro-differential equations by Taylor expansion method*, Comput. Math. Appl. 62 (2011), 1127–1134.

13. Keshavarz, E., Ordokhani, Y., and Razzaghi, M. *Bernoulli wavelet operational matrix of fractional order integration and its applications in solving the fractional order differential equations*, Appl. Math. Model. 38 (2014), 6038–6051.

14. Khader, M.M. *Numerical treatment for solving fractional Riccati differential equations*, J. Egypt Math. Soc. 21 (2013), 32–37.

15. Li, Y.L. and Sun, N. *Numerical solution of fractional differential equations using the generalized block pulse operational matrix*, Comput. Math. Appl. 62 (2011), 1046–1054.

16. Ma, X.H. and Huang, C.M. *Spectral collocation method for linear fractional integro-differential equations*, Appl. Math. Model. 38 (2014), 1434–1448.

17. Meerschaert, M.M., Benson, D.A., and Bäumer, B. *Multidimensional advection and fractional dispersion*, Phys. Rev. E 59 (1999), 5026–5028.

18. Mittal, R.C. and Nigam, R. *Solution of fractional calculus and fractional integro-differential equations by Adomian decomposition method*, Int. J. Appl. Math. Mech. 4 (2) (2008), 87–94.

19. Mokhtary, P. and Ghoreishi, F. *The L^2-convergence of the Legendre spectral Tau matrix formulation for nonlinear fractional integro-differential equations*, Numer. Algorithms 58 (2011), 475–496.

20. Mokhtary, P., Ghoreishi, F., and Srivastava, H.M. *The Müntz-Legendre Tau method for fractional differential equations*, Appl. Math. Model. 40 (2016), 671–684.

21. Nadzharyan, T.A., Sorokin, V.V., and Stepanov, G.V. *A fractional calculus approach to modeling rheological behavior of soft magnetic elastomers*, Polymer 92 (2010), 179–188.

22. Nawaz, Y. *Variational iteration method and homotopy perturbation method for fourth-order fractional integro-differential equations*, Comput. Math. Appl. 61 (2011), 2330–2341.

23. Nazari, D.and Shahmorad, S. *Application of the fractional differential transform method to fractional order integro-defferential equations with nonlocal boundary conditions*, J. Comput. Appl. Math. 234 (3) (2010), 883–891.

24. Pinkus, A. *Density in approximation theory*, Surv. Approx. Theory 1 (2005), 1–45.

25. Podlubny, I. *Fractional differential equation*, Academic Press, San Diego, 1999.

26. Rana, S., Bhattacharya, S., and Pal, J. *Paradox of enrichment: A fractional differential approach with memory*, Phys. A Stat. Mech. Appl. 392 (2013), 3610–3621.

27. Sayevand, K., Fardi, M., and Moradi, E. *Convergence analysis of homotopy perturbation method for Volterra integro-differential equations of fractional order*, Alex. Eng. J. 52 (2013), 807–812.

28. Sutulo, S. and Soares, C.G. *On applicability of mathematical models based on fractional calculus to ship dynamics*, in: Proceedings of the Conference on Control Applications in Marine Systems (2010), 15–17.

29. Sweilam, N.H., Khader, M.M., and Al-Bar, R.F. *Numerical studies for a multi-order fractional differential equation*, Phys. Lett. A. 371 (2007) 26–33.

30. Wang, J., Xu, T., Wei, Y., and Xie, J. *Numerical simulation for coupled systems of nonlinear fractional order integro-differential equations via wavelets method*, Appl. Math. Comput. 324 (2018), 36–50.

31. Wang, L.F., Ma, Y.P., and Meng, Z.J. *Haar wavelet method for solving fractional partial differential equations numerically*, Appl. Math. Comput. 227 (2014), 66–76

32. Xie, J., Huang, Q., and Zhao, F. *Numerical solution of nonlinear Volterra–Fredholm–Hammerstein integral equations in two-dimensional spaces based on Block Pulse functions*, J J. Comput. Appl. Math. 317 (2017), 565–572.

33. Xie, J. and Yi, M. *Numerical research of nonlinear system of fractional Volterra-Fredholm integral-differential equations via Block-Pulse functions and error analysis*, J. Comput. Appl. Math. 345 (2019), 159–167.

34. Yang, Y., Chen, Y.P., and Huang, Y.Q. *Convergence analysis of the Jacobi spectral-collocation method for fractional integro-differential equations*, Acta Mathematica Scientia 34B (3) (2014), 673–690.

35. Yi, M.X., Wang, L.F., and Huang, J. *Legendre wavelets method for the numerical solution of fractional integro-differential equations with weakly singular kernel*, Appl. Math. Model. 40 (2016), 3422–3437.

36. Zhang, X.D., Tang, B., and He, Y.N. *Homotopy analysis method for higher-order fractional integro-differential equations*, Comput. Math. Appl. 62 (2011), 3194–3203.

37. Zhu, L. and Fan, Q.B. *Solving fractional nonlinear Fredholm integro-differential equations by the second kind Chebyshev wavelets*, Commun. Nonlinear Sci. Numer. Simul. 17 (2012), 2333–2341.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 73–94
DOI:10.22067/ijnao.2021.11328.0

Research Article

Monotonicity-preserving splitting schemes for solving balance laws

F. Khodadosti, J. Farzi* and M.M. Khalsaraei

Abstract

In this paper, some monotonicity-preserving (MP) and positivity-preserving (PP) splitting methods for solving the balance laws of the reaction and diffusion source terms are investigated. To capture the solution with high-accuracy and resolution, the original equation with reaction source term is separated through the splitting method into two sub-problems including the homogeneous conservation law and a simple ordinary differential equation (ODE). The resulting splitting methods preserve monotonicity and positivity property for a normal CFL condition. A trenchant numerical analysis made it clear that the computing time of the proposed methods decreases when the so-called MP process for the homogeneous conservation law is imposed. Moreover, the proposed methods are successful in recapturing the solution of the problem with high-resolution in the case of both smooth and non-smooth initial profiles. To show the efficiency of proposed methods and to verify the order of convergence and capability of these methods, several numerical experiments are performed through some prototype examples.

AMS subject classifications (2020): 65N08, 35L65, 65L20.

Keywords: Balance laws; Splitting method; Monotonicity-preserving.

---

*Corresponding author
Received 26 August 2020; revised 22 October 2020; accepted 12 November 2020
Fayyaz Khodadosti
Department of Applied Mathematics, Faculty of Basic Sciences, Sahand University of Technology, Tabriz, Iran, e-mail: fayyaz64dr@gmail.com

Javad Farzi
Department of Applied Mathematics, Faculty of Basic Sciences, Sahand University of Technology, Tabriz, Iran, e-mail: farzi@sut.ac.ir

Mohammad Mehdizadeh Khalsaraei
Faculty of Mathematical Science, University of Maragheh, Maragheh, Iran, e-mail: Muhammad.mehdizadeh@gmail.com

1 Introduction

Balance laws appear in many different areas such as fluid dynamics, gas dynamics, chemical reactions [13, 19, 1, 14, 29, 35, 37]. These equations are hyperbolic and are sometimes called inhomogeneous conservation laws in the literature on physics. In particular, these equations are described as follows:

$$\begin{cases} u_t + g(u)_x = f(u), \\ u_0(x) = u(x,0), \ x \in I, \end{cases} \tag{1}$$

where $g(u)$ and $f(u)$ are smooth functions of u and I is the computational domain. The characterization and design of high-order and high-resolution schemes is a challenging task in the numerical analysis [12, 19, 1, 24, 25, 26, 29, 4]. The main goal of this paper is, therefore, to design schemes with high-accuracy and resolution to obtain the true solution of the balance laws including reaction and diffusion terms [19, 14, 7]. One way to deal with such problems is to split the original problem into two sub-problems and then solve each problem independently by applying an appropriate technique, which is sometimes called the fractional step approach [5, 28, 29, 32]. To this end, the problem (1) splits into the following sub-problems

$$\begin{cases} P_1: \ u_t + g(u)_x = 0, \\ P_2: \ u_t = f(u). \end{cases}$$

In other words, the original problem (1) splits into the homogeneous conservation law P_1 and the ODE problem P_2. Mathematically, each sub-problem is computationally solved by a standard numerical scheme. However, solving the initial value sub-problem P_1 by a suitable numerical approach such as high-resolution shock-capturing methods, and then considering its solution as an initial profile for the sub-problem P_2 will eventually lead to the solution of the original problem as we solve sub-problem P_2 with an ODE solver such as Runge-Kutta (RK) or multi-step methods. For a semi-discrete discretization such as linear multi-step methods, we can study the global error propagation [10]. This procedure is regarded as a simple method to deal with this problem and often produces good results. Also, using this process the high-resolution methods such as the total variation diminishing (TVD), weighted-ENO (WENO) and MP methods can be applied directly for solving the homogeneous conservation law P_1 [1, 29].

In the past few decades, some numerical methods for evaluating the solution of the balance laws have been applied. For instance, Strang [4] proposed a splitting method for solving the balance law equations. Hundsdorfer and Verwer in [19] investigated one-step and multi-step methods using a splitting method to solve the advection-reaction equations. Moreover, they analyzed and extended the operator splitting method for a wide range of balance laws including advection-diffusion-reaction equations in one- and two-dimensional spaces [1]. LeVeque in [29] introduced an analysis of numerical methods for solving balance equations. In [23], the corrected operator splitting method for solving nonlinear parabolic equations of a convection-diffusion type has been presented. Langseth et al. in [28] studied the order of convergence for operator splitting for balance laws. In [35], an A-stability achievement for operator splitting type time integration methods applied to advection-diffusion-reaction equations with possibly indefinite source terms has been achieved. Nuri et al. obtained the numerical behavior of a modified Burger's equation using cubic B-spline collocation finite element method after splitting the equation with Strang splitting technique in [40]. Holly and Polatera in [18] presented a numerical method to model the contaminant dispersion in two-dimensional tidal currents. In fact, they applied the high-order bi-cubic Hermite interpolation with characteristics to obtain the true solution in the presence of the advection part as well as the well-known Crank-Nicolson scheme for the diffusion part. One of the methods of discretization of the homogeneous conservation laws is the use of MP methods. Some numerical schemes with the monotonicity-preserving property are provided in the references [15, 5, 9, 2, 15, 17, 31, 17]. Suresh and Huynh [5] proposed an accurate MP scheme to homogeneous conversation laws. In the MP scheme, starting with an original numerical flux computed by any high-order scheme, a local interval is computed by enlarging the first-order MP interval. Then, the original numerical flux is preserved or replaced based on the relation between the original numerical flux and this local [8, 6, 16, 2, 13, 5]. This process is, however, capable of preserving the monotonicity property and, in contrast to conventional TVD methods, maintains the accuracy of solution in the extremum points. The MP scheme has a simple algorithm to achieve a high-order of accuracy. Moreover, this scheme is widely applied to solve hyperbolic partial differential equations [1, 6, 2, 15, 16, 5]. In this paper, we focus on designing the MP splitting schemes based on the idea of the MP scheme for solving balance laws. This process has some advantages including preserving monotonicity and positivity with high-accuracy, having a high speed of implementation due to the MP process [5], higher resolution at discontinuous points and yielding accurate results in the smooth region.

The outline of this paper is as follows. Details on the construction of the operator splitting schemes are presented in Section 2. In Section 3, the construction of the MP process for homogeneous conservation laws will be described in detail. The main results for the MP and PP splitting process for balance laws will be discussed in Section 4. To examine the constructed methods, some numerical experiments are given in Section 5, and finally, some concluding remarks are given in Section 6.

2 Operator splitting schemes

Here, we discuss the discretization of the problem (1) corresponding to sub-problems P_1 and P_2. For this reason, we consider two types of the operator splitting schemes. The first type is the so-called *basic splitting scheme* and is applied to the sub-problems P_1 and P_2 as follows:

$$\frac{d}{dt}u^*(t) = L\big(t, u^*\big)(t), \qquad \text{for} \quad t_n < t \leq t_{n+1}, \;\; u^*(t_n) = u_n, \tag{2}$$

$$\frac{d}{dt}u^{**}(t) = F\big(t, u^{**}\big)(t), \quad \text{for} \;\; t_n < t \leq t_{n+1}, \;\; u^{**}(t_n) = u^*(t_{n+1}), \tag{3}$$

where u_n stands for the approximating $u(t_n)$, $L\big(t, u(t)\big)$ denotes the discretized homogeneous operator for conservation laws and $F\big(t, u(t)\big)$ is the operator for reaction source term. Finally, the solution of the problem (1) is obtained as we set $u^{n+1} := u^{**}(t_{n+1})$ in the interval $[t_n, t_{n+1}]$. Also, the basic splitting scheme for the sub-problems P_1 and P_2 provides an error bound for the solution of order $\mathcal{O}(\Delta t)$ [19, 4], where Δt is the splitting time step. The second type is the *Strang splitting method* and is implemented as follows:

$$\frac{d}{dt}u^*(t) = L\big(t, u^*\big)(t), \quad t_n < t \leq t_{n+\frac{1}{2}}, \;\; u^*(t_n) = u_n, \tag{4}$$

$$\frac{d}{dt}u^{**}(t) = F\big(t, u^{**}\big)(t), \quad t_n < t \leq t_{n+1}, \;\; u^{**}(t_n) = u^*(t_{n+\frac{1}{2}}), \tag{5}$$

$$\frac{d}{dt}u^{***}(t) = L\big(t, u^{***}\big)(t), \quad t_{n+\frac{1}{2}} < t \leq t_{n+1}, \;\; u^{***}(t_{n+\frac{1}{2}}) = u^{**}(t_{n+1}), \tag{6}$$

where the operators $L\big(\cdot, u(\cdot)\big)$ and $F\big(\cdot, u(\cdot)\big)$ are the same as those in basic splitting scheme. The final solution of the problem (1) is obtained by setting $u^{n+1} = u^{***}(t_{n+1})$. Strang splitting method introduces an order $\mathcal{O}(\Delta t^2)$ [19, 29, 40]. The homogeneous conservation laws in the sub-problem P_1 are discretized via step-size $\Delta x = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ in a uniform grid and is defined by the points $x_j = j\Delta x$, $j = 1, 2, ..., N$, with cell boundaries given by $x_{j+\frac{1}{2}} = x_j + \frac{\Delta x}{2}$ in the conservation form

$$\frac{du_j(t)}{dt} = -\frac{1}{\Delta x}\big(f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}\big),$$

where $f_{j\pm\frac{1}{2}}$ are numerical fluxes. Our aim is to calculate u_j^{n+1} at time level t_{n+1}. It is worth noting that the algorithm for solving equation involving time is based on the well-known three-stage, third-order TVD-RK3 method [15, 36, 5]. Consider $\hat{f}_{j\pm\frac{1}{2}}$ as the original numerical fluxes which are approximations of the point value $u\big(x_{j\pm\frac{1}{2}}, t\big)$ at the time level t_n. We use an original numerical flux with a five-point stencil. Since, we need at least five points

Table 1: The constant coefficients β_l for $k = 3, 4, 5$.

k	β_{-5}	β_{-4}	β_{-3}	β_{-2}	β_{-1}	β_0	β_1	β_2	β_3	β_4	β_5
3			$-\frac{1}{140}$	$\frac{5}{84}$	$-\frac{101}{420}$	$\frac{319}{420}$	$\frac{107}{210}$	$-\frac{19}{210}$	$\frac{1}{105}$		
4		$\frac{1}{630}$	$-\frac{41}{2520}$	$\frac{199}{2520}$	$-\frac{641}{2520}$	$\frac{1879}{2520}$	$\frac{275}{504}$	$-\frac{61}{504}$	$\frac{11}{504}$	$-\frac{1}{504}$	
5	$-\frac{1}{2772}$	$\frac{61}{13860}$	$-\frac{703}{27720}$	$\frac{371}{3960}$	$-\frac{7303}{27720}$	$\frac{20417}{27720}$	$\frac{15797}{27720}$	$-\frac{4003}{27720}$	$\frac{947}{27720}$	$-\frac{17}{3080}$	$\frac{1}{2310}$

to distinguish between extremum and discontinuity points. Therefore, one choice is to consider the original numerical flux of order $\mathcal{O}(\Delta x^5)$ as follows:

$$\hat{f}_{j+\frac{1}{2}} = \frac{1}{60}\Big(2g_{j-2} - 13g_{j-1} + 47g_j + 27g_{j+1} - 3g_{j+2}\Big). \tag{7}$$

Furthermore, to increase spatial accuracy, the original numerical fluxes of the $(2k+1)$-th order in the following form

$$\hat{f}_{j+\frac{1}{2}} = \sum_{l=-k}^{k} \beta_l g_{j+l}, \;\; k = 3, 4, 5,$$

can be used. The constant coefficients β_l are shown in Table 1. Moreover, compact numerical schemes are relatively high-order with a small stencil. However, these numerical schemes can be applied as another choice [15]. This process produces some oscillations in the numerical solution as the above-mentioned numerical fluxes are assembled. To overcome this difficulty, we will discuss the monotonicity property for homogeneous conservation laws in the next section to obtain accurate numerical schemes inspired by the process of MP as proposed in [5].

3 Construction of the MP process for conservation laws

In this section, we will study the construction of the MP process for conservation laws with numerical flux (7). To this end, the numerical flux (7) is considered to be the original numerical flux $\hat{f}_{j+\frac{1}{2}}$. Because of this, some oscillations near the discontinuous points are produced as we utilize the numerical flux $\hat{f}_{j+\frac{1}{2}}$. Therefore, we limit the numerical flux endowed with MP process in such a way that the oscillations near the discontinuous points are damped. To construct an MP process for the numerical flux (7), some definitions are required. The first definition is the minmod function for n arguments as follows:

$$\text{minmod}\big(z_1, z_2, ..., z_n\big) = s. \min\big(\mid z_1 \mid, \mid z_2 \mid, ..., \mid z_n \mid\big),$$

where

$$s = \frac{1}{2}\Big(\text{sign}(z_1)+\text{sign}(z_2)\Big)\times\left|\frac{1}{2}\Big(\text{sign}(z_1)+\text{sign}(z_3)\Big)...\frac{1}{2}\Big(\text{sign}(z_1)+\text{sign}(z_n)\Big)\right|.$$

The second one is the interval $\big[\min(z_1, z_2, ..., z_n), \max(z_1, z_2, ..., z_n)\big]$ denoted by $I\big[z_1, z_2, ..., z_n\big]$. The third one is the local curvature

$$d^{MM}_{j+\frac{1}{2}} = \text{minmod}\big(d_j, d_{j+1}\big),$$

where

$$d_j = u_{j+1} - 2u_j + u_{j-1}.$$

For simplicity of notation, we omitted the superscript n, e.g., u^n_j denoted by u_j. We consider the original numerical flux of five-order in the formula (7). Then, this original numerical flux is maintained or replaced according to the following limiting procedures. More precisely, if we suppose $u_j \leq \hat{f}_{j+\frac{1}{2}} \leq u_{j+1}$ some simple calculations will show that the derived solution, u^{n+1}_j, lies somewhere between u_{j-1} and u_j, which yields MP-property. Therefore, these ensure that $\hat{f}_{j+\frac{1}{2}}$ lies between u_j and u^{UL} as follows:

$$u^{UL} = u_j + \alpha\big(u_j - u_{j-1}\big),$$

where $\alpha \geq 2$. According to the above-mentioned assumptions, we derive a first-order MP interval $I[u_j, u^{MP}]$, where u^{MP} is defined in the following form

$$u^{MP} = u_j + \text{minmod}\Big(u_{j+1} - u_j, \alpha\big(u_j - u_{j-1}\big)\Big).$$

For simulation, the condition for which the original numerical flux $\hat{f}_{j+\frac{1}{2}}$ lies in $I[u_j, u^{MP}]$, is equivalent to

$$\big(\hat{f}_{j+\frac{1}{2}} - u_j\big)\big(\hat{f}_{j+\frac{1}{2}} - u^{MP}\big) \leq \mu,$$

with a tolerance $\mu = 10^{-10}$. For $\hat{f}_{j+\frac{1}{2}} \in I[u_j, u^{MP}]$ the accuracy of the MP method decreases to the first order in the vicinity of the extremum points. To increase accuracy in the vicinity of the extremum points, the intervals $I[u_j, u_{j+1}]$ and $I[u_j, u^{UL}]$ were enlarged as $I\big[u_j, u_{j+1}, u^{MD}\big]$ and $I\big[u_j, u^{UL}, u^{LC}\big]$, respectively, where

$$u^{MD} = \frac{1}{2}\big(u_j + u_{j+1}\big) - \frac{1}{2}d^{MM}_{j+\frac{1}{2}},$$

and

$$u^{LC} = \frac{1}{2}\big(3u_j - u_{j-1}\big) + \frac{4}{3}d^{MM}_{j-\frac{1}{2}}.$$

Two intervals $I\big[u_j, u^{UL}, u^{LC}\big]$ and $I\big[u_j, u_{j+1}, u^{MD}\big]$ enlarge only for the local non-monotonicity numerical data, and degenerate to $I[u_j, u^{UL}]$ and $I[u_j, u_{j+1}]$ for the local monotonicity numerical data [5]. In practice, we replace $d^{MM}_{j+\frac{1}{2}}$ with $d^{M4}_{j+\frac{1}{2}}$ and define

$$d^{M4}_{j+\frac{1}{2}} = \text{minmod}\Big(4d_j - d_{j+1}, 4d_{j+1} - d_j, d_j, d_{j+1}\Big).$$

Intersection $I[u^{\min}, u^{\max}]$ of the two intervals $I[u_j, u^{UL}, u^{LC}]$ and $I[u_j, u_{j+1}, u^{MD}]$ can be computed by

$$\begin{aligned} u^{\min} &= \max\Big[\min\big(u_j, u_{j+1}, u^{MD}\big), \min\big(u_j, u^{UL}, u^{LC}\big)\Big], \\ u^{\max} &= \min\Big[\max\big(u_j, u_{j+1}, u^{MD}\big), \max\big(u_j, u^{UL}, u^{LC}\big)\Big], \end{aligned}$$

which indicate the local interval $I[u^{\min}, u^{\max}]$ for the MP process. The numerical original flux is replaced by the nearest bound of the interval if the original numerical flux is outside of the interval. Therefore, we have

$$f_{j+\frac{1}{2}} = \hat{f}_{j+\frac{1}{2}} + \text{minmod}\Big(u^{\max} - \hat{f}_{j+\frac{1}{2}}, u^{\min} - \hat{f}_{j+\frac{1}{2}}\Big).$$

Summarizing the above scenario, accuracy and monotonicity are preserved, which means the MP process is of high-order and high-resolution in the case of the smooth and non-smooth initial profiles, respectively. In comparison with the other existing methods such as the fifth-order weighted-ENO (WENO5) and the third-order essentially nonoscillatory (ENO3), the speed of the MP algorithm is of high rate, which makes the MP algorithm an efficient and effective procedure to solve the homogeneous conservation laws [5]. It is well-known that the MP scheme has higher accuracy and less dissipation compared with the mentioned schemes [1, 6, 5]. Moreover, the high-order WENO methods may not have monotonicity property but the MP scheme preserves monotonicity property [1]. Therefore, due to the mentioned reasons, we consider the MP method to discretize the homogeneous conservation laws part in the splitting process.

4 Main results

4.1 Monotonicity preserving

In this section, we determine results for the basic splitting (2), (3) and the Strang splitting (4)−(6) with the above mentioned MP process for the homogeneous conservation laws part. These schemes are called the MP basic splitting scheme and the MP Strang splitting scheme, respectively. We state the following theorem which determines the condition of monotonicity-preserving on the MP basic splitting scheme.

Theorem 1. The MP basic splitting scheme is monotonicity-preserving under the CFL condition $\nu \leq \frac{1}{1+\alpha}$, where ν is CFL number.

Proof. Suppose that the sub-problem (2) is computed with the TVD-RK3 method in time and the MP scheme of five-order in space. Therefore, we suppose that $u_{j-1} \leq \hat{f}_{j-\frac{1}{2}} \leq u_j$, and $u_j \leq \hat{f}_{j+\frac{1}{2}} \leq u^{UL}$. For increasing data, these two assumptions imply that $(u^{UL} - u_{j-1}) \leq (\alpha+1)(u_j - u_{j-1})$. After applying the relation $(u^{UL} - u_{j-1}) \leq (\alpha+1)(u_j - u_{j-1})$, we will have

$$0 \leq (\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}) \leq (u^{UL} - u_{j-1}) \leq (\alpha+1)(u_j - u_{j-1}). \tag{8}$$

Multiplying $-\nu$ and then adding u_j to the relation (8), we will obtain

$$(1-\nu(1+\alpha))u_j + \nu(\alpha+1)u_{j-1} \leq u^* \leq u_j,$$

where $u^* = u_j - \nu(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}})$ is the first step of the TVD-RK3 method. To hold $u_{j-1} \leq u^* \leq u_j$, the inequality $u_{j-1} \leq (1-\nu(1+\alpha))u_j + \nu(\alpha+1)u_{j-1}$ must be satisfied, which in turn yields

$$\nu(\alpha+1)(u_j - u_{j-1}) \leq (u_j - u_{j-1}).$$

Therefore, we have $\nu(\alpha+1) \leq 1$. Since the second and third steps of the TVD-RK3 method are combinations of the first step and u_j with positive weights, the final step $u^*(t_{n+1})$ is also monotonic. Consequently, the first step of the basic splitting scheme under the CFL condition $\nu(\alpha+1) \leq 1$ is monotonicity-preserving. The sub-problem (3) is computed with the TVD-RK3 method with initial condition $u^{**}(t_n) = u^*(t_{n+1})$ and also the TVD-RK3 method is monotonic; It is well known that if sub-problems is solved by a high resolution or a monotonicity-preserving method, then the splitting methods (such as the basic splitting, Strang splitting and dimension splitting schemes) are monotonicity-preserving and convergent to the correct solution of original equation [4, 1, 29]. however, the MP basic splitting is monotonicity-preserving under the CFL condition $\nu \leq \frac{1}{1+\alpha}$. □

Here, we present the following theorem to estimate the error of the MP basic splitting method.

Theorem 2. Suppose (2), (3) are computed with TVD-RK3 in time and the MP process of five-order in space. Then, in the absence of the splitting error, the global order of the error can be introduced as $\mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta x^5)$.

Proof. The computation of equation (2) with initial data u_n yields approximations $u^*(t_{n+1})$ with an order error of the form $\mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta x^5)$ [5]. More pointedly, the evaluation of equation (3) with TVD-RK3 method would result in local error of $\mathcal{O}(\Delta t^3)$ in time [36]. Consequently, Due to the stability of sub-schemes the global order of the error can be obtained as $\mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta x^5)$, when the splitting error is neglected. □

Also, to make sure that the convergency of the MP basic splitting method is held, we state the following lemma.

Lemma 1. The MP basic splitting scheme is convergence to the true solution of original equation (1).

Proof. As was mentioned above, subproblem (2) is computed with a high-resolution method in space (i.e., the MP process of five-order) and the method of TVD-RK3 in time with initial condition $u^*(t_n) = u_n$ in interval $[t_n, t_{n+1}]$, while the subproblem is solved in interval $[t_n, t_{n+1}]$ only by the TVD-RK3 method with monotone initial data $u^{**}(t_n) = u^*(t_{n+1})$. Therefore, because sub-problems are solved using high-resolution methods (we know that high-resolution methods converge to the correct solution of the problem [29]) and the use of the well-known basic splitting method, also, because there is consistency between the approximation of the splitting method (2), (3) with the solution of original equation (1), the convergence of basic MP splitting method to the true solution of original equation (1) is guaranteed [4, 1, 29, 4]. □

4.2 Positivity preserving

Next, we will investigate positivity property for the basic splitting (2), (3) sophistically in detail. We consider the case that the solutions $u(x,t)$ of the basic splitting (2), (3) are nonnegative for all times $t \geq 0$.

In the following, we state a lemma for positivity property of the basic splitting (2), (3) obtained using the MP method and the TVD-RK3 method.

Lemma 2. The MP basic splitting scheme will be positivity-preserving under the CFL condition $\nu \leq \frac{1}{1+\alpha}$.

Proof. Suppose that for $t > 0$ the problem (2) is solved with initial condition $u^*(t_n) = u_n \geq 0$ in interval $[t_n, t_{n+1}]$ using the MP process in space and the TVD-RK3 method in time. Therefore, the solution of the MP method is monotonicity-preserving under the CFL condition $\nu \leq \frac{1}{1+\alpha}$. However, the problem (3) is solved in interval $[t_n, t_{n+1}]$ by the TVD-RK3 method with monotone initial data $u^{**}(t_n) = u^*(t_{n+1})$. Since the TVD-RK3 method with positive weights is positivity-preserving, therefore, $u^{n+1} := u^{**}(t_{n+1}) \geq 0$. As a result, the MP basic splitting is positivity-perserving under the CFL condition $\nu \leq \frac{1}{1+\alpha}$. □

It is interesting to note that, the above-mentioned results are also valid for the MP Strang splitting method, i.e., the MP Strang splitting method is convergent to the true solution of original equation (1) and also preserves monotonicity and positivity under the CFL condition $\nu \leq \frac{1}{1+\alpha}$.

4.3 The case for diffusion source term

In this section, we study the MP splitting process for balance laws with diffusion source term in the following form

$$u_t + f(u)_x = \eta \, u_{xx}, \tag{9}$$

where $f(u)$ is the flux function, and η denoting the diffusion coefficient. By applying the basic splitting method, the equation (9) is divided into two sub-problems as follows:

$$u_t^* + f(u^*)_x = 0, \qquad \text{for} \quad t_n \leq t \leq t_{n+1}, \;\; u^*(x, t_n) = u(x, t_n), \tag{10}$$

$$u_t^{**} = \eta \, u_{xx}^{**}, \quad \text{for} \;\; t_n \leq t \leq t_{n+1}, \;\; u^{**}(x, t_n) = u^*(x, t_n), \tag{11}$$

where u^* and u^{**} stand for concentration in the case of homogeneous conservation law and the diffusion process, respectively. Executing the MP process, equation (10) can simply be solved by using the initial condition of equation (9). The solution of this problem can be regarded as an initial condition for equation (11). Using the Crank-Nicolson method [8, 23], equation (11) can be numerically solved through an initial condition derived from the previous step. Consequently, the solution to the original problem (9) is obtained. It turns out that the proposed process can stably and consistently estimate the true solution of the problem (9). This process can, however, be applied to solving a wide variety of balance laws with possibly nonlinear source terms. As it is well-known, the effective and applicable strategy for solving the considered problem turns out to be implemented by the MP process. This method is a robust and powerful technique for treating balance laws. Moreover, it may be used in investigating problems involving diffusion source terms. To shed light on this issue, we will present the following remark which is, in essence, similar to the one given by Lemma 1.

Remark 1. It is notable that the proposed method is not only able to treat problems in the presence of reaction source terms, but also robustly and rigorously reconstructs the true solution as some diffusion source terms are imposed. Indeed, the proposed method can therefore retrieve the true solution of the original equation (9) by means of splitting method like what is mentioned in Lemma 1. This consideration makes us assert that this method can actually guarantee the convergency of the solution to the original equation (9).

5 Numerical results

In this section, some test problems are provided to show the efficiency and effectiveness of the MP splitting schemes. In all tests, $\alpha = 4$ and $\mu = 10^{-10}$ are considered. Meanwhile, all tests were performed by MATLAB routine codes.

Test 1. Consider the following equation

$$u_t + \big(au\big)_x = -\beta u, \quad a = 1 \text{ and } \beta = 1,\ 0 \leq x \leq 1,$$

with initial condition

$$u_0(x) = \sin^4\big(\pi x\big),$$

and the periodic boundary conditions, $u(0,t) = u(1,t)$, at final time $t_f = 1$. The results of the MP basic splitting scheme and the MP Strang splitting scheme are demonstrated with $\frac{\Delta t}{\Delta x} = 0.05$ and $\frac{\Delta t}{\Delta x} = 0.4$ in tables 2 and 3, respectively. The numerical results confirm the good efficiency and accuracy of the MP splitting schemes for this test. It is worth emphasizing that the numerical results for large CFL number $\frac{\Delta t}{\Delta x} = 0.4$ are also satisfactory.

Test 2. We solve

$$u_t + \big(au\big)_x = -\beta u, \quad a = 1 \text{ and } \beta = 10, \quad 0 \leq t \leq \frac{1}{2},$$

with initial condition given by

$$u_0(x) = 1000\cos\big(\pi x\big)^2,$$

and the periodicity conditions in the boundaries on the interval [0, 1], at final time $t_f = \frac{1}{2}$ and discrete points $N = 40$. Numerical solution of the advection-reaction equation with the MP basic splitting method for $\frac{\Delta t}{\Delta x} = 0.4$ is depicted in Figure 2. From this figure, one can deduce that the proposed method is successful in retrieving the exact solution in a careful manner. Moreover, in comparison with the third order TVD basic splitting scheme [1], our proposed method is not only able to recover the true solution of the problem accurately, but also it preserves the accuracy of the numerical solution at extremum points. This situation also happens for other test problems mainly because the spatial part of the homogeneous problem is discretized by the MP-process. Yet the lack of such property in the third order TVD-scheme makes its efficiency unsatisfactory. Furthermore, we consider the advection-reaction equation as follows:

$$u_t + \left(au\right)_x = -\beta u, \quad a = 1 \text{ and } \beta = 1, \tag{12}$$

with initial condition

$$u_0(x) = \left(\sin\left(\pi x\right)\right)^{100}, \tag{13}$$

and the periodicity conditions in the interval $[0, 1]$, at final time $t_f = 2$ and discrete points $N = 64$. Numerical solution of the advection-reaction equation (12)-(13) with the MP basic splitting method for $\frac{\Delta t}{\Delta x} = 0.2$ is illustrated in Figure 1. In this test, we can deduce that the numerical solution of the MP basic splitting method, which preserves nonlinear stability such as monotonicity and positivity. To be more precise, this test is given specifically to confirm the positivity property of the MP basic splitting for CFL condition $\nu \leq \frac{1}{1+\alpha}$ (as shown in Lemma 2) and it shows the MP basic splitting method is monotonicity-preserving for this CFL condition (as shown in Theorem 1).

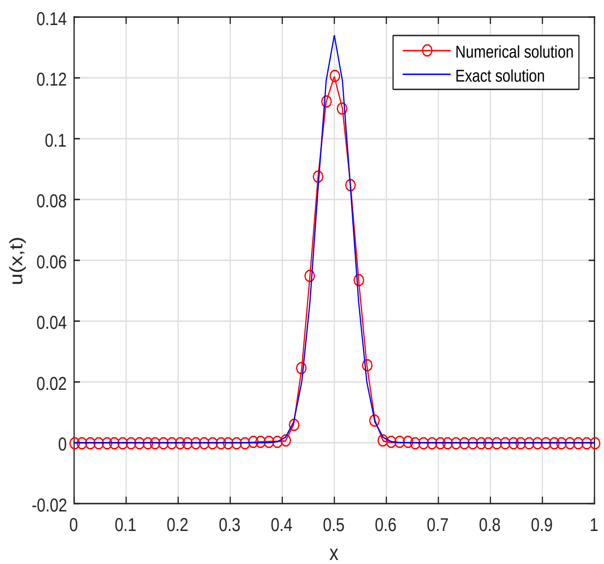


Figure 1: Numerical solution of the advection-reaction equation (12)-(13) with the MP basic splitting scheme at final time $t_f = 2$ and $N = 64$, $\frac{\Delta t}{\Delta x} = 0.2$.

Test 3. Consider the following equation governed by

$$u_t + \big(au\big)_x = -\beta u, \quad a = 1 \text{ and } \beta = 1,$$

with the initial condition

$$u_0(x) = \operatorname{sign}(x),$$

and the periodic boundary conditions in the interval $[0, 2]$. The results are shown at final time $t_f = 1$ in Figure 6. We get a good behavior for the MP Strang splitting scheme (even for CFL$= 0.4$). It is obvious that the MP Strang splitting scheme produces sharp results in the vicinity of the shock.

Test 4. Let us take the equation

$$u_t + \big(au\big)_x = u\big(1 - u\big), \quad a = 1,$$

with the initial condition

$$u_0(x) = \exp\big(-0.01x^2\big),$$

and the periodic boundary conditions in the interval $[0, 2]$. The results of the MP Strang splitting scheme are illustrated in Figure 4 with $\frac{\Delta t}{\Delta x} = 0.4$, at final time $t_f = 1$ and $N = 100$. However, for the final time $t_f = 1$ and discrete points $N = 100$, the MP Strang splitting scheme is compared to the nonstandard finite difference (NSFD) scheme [33], which is stable and also is of first-order in time and space as follows:

$$\frac{u_j^{n+1} - u_j^n}{\phi(\Delta t)} + \frac{u_j^n - u_{j-1}^n}{\phi(\Delta x)} = u_{j-1}^n\big(1 - u_j^{n+1}\big),$$

where $\phi(h) = \exp(h) - 1$, and $\Delta t = \Delta x$ is an imperative condition to hold this scheme. We find that the MP Strang splitting scheme has a good resolution and convergence. Moreover, the result obtained by the MP Strang splitting scheme is free of the spurious oscillations.

Test 5. We apply the MP Strang splitting scheme to the equation

$$u_t + \Big(\frac{1}{2}u^2\Big)_x = u\big(1 - u\big), \tag{14}$$

with the initial value

$$u_0(x) = \begin{cases} 0.1 + 0.1\sin\big(2\pi x\big), & \text{if } 0 \le x \le 1, \\ 0.1, & \text{else,} \end{cases} \tag{15}$$

and the periodic boundary conditions in the interval $[0, 2]$. The numerical solution is depicted at $t_f = 1.5$ in Figure 5. We can see that the MP Strang splitting scheme can solve high-accuracy and resolution. Obviously, the dissipation of the MP Strang splitting scheme is much small.

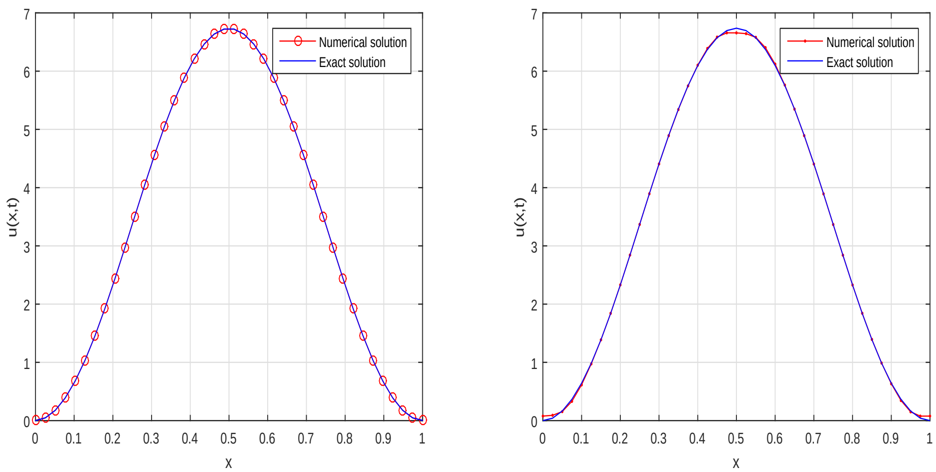


Figure 2: Numerical solution of the advection-reaction equation with the MP basic splitting scheme (left), the third order TVD basic splitting scheme (right) at final time $t_f = \frac{1}{2}$ and $N = 40$, $\frac{\Delta t}{\Delta x} = 0.4$ in Test 2.

Table 2: Estimated L_2-errors and L_2-orders for advection-reaction equation with the MP basic splitting scheme in Test 1.

N	$\frac{\Delta t}{\Delta x} = 0.05$		$\frac{\Delta t}{\Delta x} = 0.4$	
	L_2-error	L_2-order	L_2-error	L_2-order
20	6.13e−04	-	7.64e−04	-
40	2.06e−05	4.89	5.43e−05	3.96
80	6.58e−07	4.96	4.98e−06	3.45
160	2.15e−08	4.94	5.64e−07	3.14

Test 6. In this test, we apply the MP basic splitting scheme mentioned in subsection 4.3 to the equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = \eta\, u_{xx}, \quad \eta = 1, \tag{16}$$

with the initial condition

$$u_0(x) = \sin(x), \tag{17}$$

Table 3: Estimated L_2-errors and L_2-orders for advection-reaction equation with the MP Strang splitting scheme in Test 1.

N	$\frac{\Delta t}{\Delta x}=0.05$		$\frac{\Delta t}{\Delta x}=0.4$	
	L_2-error	L_2-order	L_2-error	L_2-order
20	6.10e−04	-	6.42e−04	-
40	2.04e−05	4.90	2.46e−05	4.70
80	6.50e−07	4.98	1.18e−06	4.38
160	2.04e−08	4.99	8.81e−08	3.80

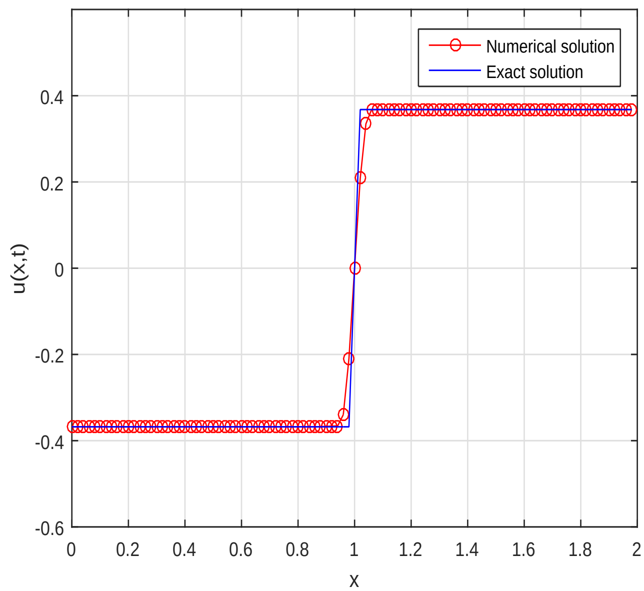


Figure 3: Numerical solution of the advection-reaction equation with the MP Strang splitting scheme at final time $t_f = 1$ and $N = 200$, $\frac{\Delta t}{\Delta x} = 0.4$ in Test 3.

and the boundary conditions $u(-4,t) = u(4,t) = 0$ in the interval $[-4,4]$. The results of the MP basic splitting scheme are illustrated in Figure 6 with $\frac{\Delta t}{\Delta x} = 0.4$, at final time $t_f = 4$ and $N = 100$. We can deduce that numerical solution given by the MP basic splitting scheme nicely converges to the true solution.

Test 7. Now we consider the one-dimensional modified Burger's equation (MBE) in the following form

$$u_t + u^{\beta}(u)_x - \eta u_{xx} = 0, \quad 0 \le x \le 1, \;\; t \ge 1, \;\; \beta = 2, \tag{18}$$

with the initial condition

$$u(x, t_0 = 1) = \frac{x}{1 + \frac{1}{c} e^{\frac{x^2}{4\eta}}},\ t \geq 1,\ 0 \leq x \leq 1,\ 0 \leq c \leq 1, \tag{19}$$

and the boundary conditions

$$\begin{cases} u(0,t) = u(1,t) = 0, \\ u_x(0,t) = u_x(1,t) = 0, \\ u_{xx}(0,t) = u_{xx}(1,t) = 0. \end{cases}$$

To apply the MP basic splitting, we split the original equation (18) into two the sub-problems P_1 and P_2 as follows:

$$\begin{cases} P_1 :\ u_t + \left(\frac{u^{\beta+1}}{\beta+1}\right)_x = 0, \\ \\ P_2 :\ u_t = \eta u_{xx}. \end{cases}$$

Then, we get the solution of the MP basic splitting scheme mentioned in subsection 4.3 in the interval $[0, 1]$. The results of the MP basic splitting scheme are illustrated in Table 4 with $\Delta t = 0.01$, $\Delta x = 0.0625$, $\eta = 0.01$, $c = 0.5$, at final time $t_f = 2$. From Table 4 with these parameters, we can see that the MP basic splitting scheme error norms L_∞ and L_2 are in general better than the proposed methods in references [34, 40]. MATLAB code of the MP process for this test is placed in the Appendix.

Table 4: A comparison of the MP basic splitting scheme error norms L_∞ and L_2 with the proposed methods in references [34, 40].

Schemes	L_∞-norm	L_2-norm
$S_{\Delta t}$[40]	$7.4741e-04$	$3.2033e-04$
[34]	$7.5978e-04$	$3.4748e-04$
MP basic splitting scheme	$2.2677e-04$	$1.4944e-04$

6 Conclusion

In this paper, the MP splitting schemes for solving balance laws are proposed. The MP splitting schemes are introduced to preserve the monotonicity and positivity properties of the numerical scheme for balance laws with linear and

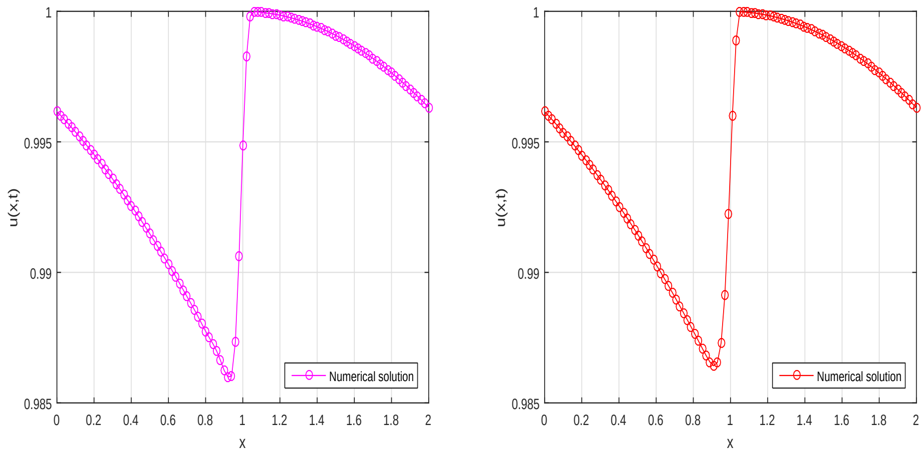


Figure 4: Numerical solution of the advection-reaction equation with the MP Strang splitting method (left), the NSFD method (right) at final time $t_f = 1$ and $N = 100$ in Test 4.

nonlinear source terms. In these methods, the homogeneous conservation laws step is discretized by the fifth order MP method, and the reaction step and time integration are discretized by the third-order TVD-RK3 method. The numerical results for balance laws as well as the case for diffusion source term are also provided results that verify the applicability and efficiency of the proposed schemes. It is worth studing the MP procedure to the interface problems that have a wide application in science and engineering [11].

Acknowledgements

Authors are grateful to there anonymous referees and editor for their constructive comments.

APPENDIX: MATLAB code of the MP process

```
function [fplus] = MP(u)
B1 = 1/60;
B2 = 4/3;
alpha =4.0;
eps = 10^-10;
```

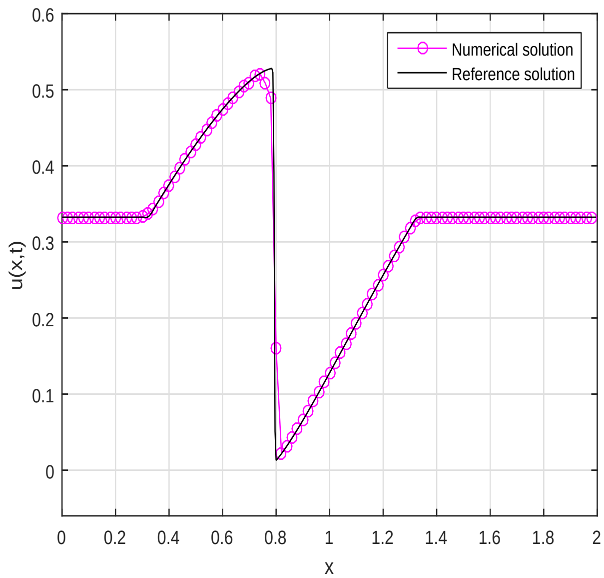


Figure 5: Numerical solution of the problem (14)-(15) with the MP Strang splitting method at final time $t_f = 1.5$ and $N = 100$, $\frac{\Delta t}{\Delta x} = 0.4$ in Test 5.

```
fhat =B1*(2 *u(1)-13*u(2)+47*u(3)+27*u(4)-3*u(5));
uMP = u(3)+DMM((u(4)-u(3)),alpha*(u(3)-u(2)));
if ((fhat-u(3))*(fhat-uMP)<=eps)
   fplus=fhat ;
else
DJM1 = (u(1)-2*u(2)+u(3));
DJ   = (u(2)-2*u(3)+u(4));
DJP1 = (u(3)-2*u(4)+u(5));
 DM4JPH = DM4(4*DJ-DJP1,4*DJP1-DJ,DJ,DJP1);
 DM4JMH = DM4(4*DJ-DJM1,4*DJM1-DJ,DJ,DJM1);
uUL = u(3)+alpha*(u(3)-u(2));
uAV = 0.5*(u(3)+u(4));
uMD = uAV-0.5*DM4JPH;
uLC = u(3)+0.5*(u(3)-u(2))+B2*DM4JMH;
umin =  max(min(min(u(3),u(4)),uMD),min(min(u(3),uUL),uLC));
umax =  min(max(max(u(3),u(4)),uMD),max(max(u(3),uUL),uLC));
fplus = fhat+DMM(umin-fhat,umax-fhat);
end
return
function A=DM4(W,X,Y,Z)
```

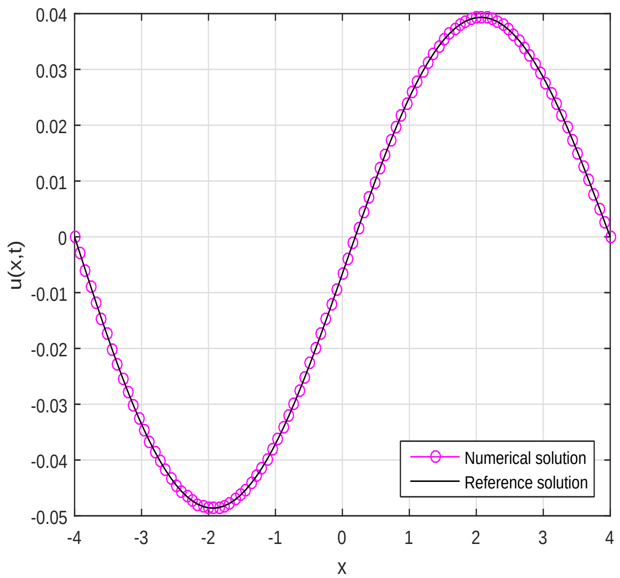


Figure 6: Numerical solution of the problem (16)-(17) with the MP basic splitting method at final time $t_f = 4$ and $N = 100$, $\frac{\Delta t}{\Delta x} = 0.4$ in Test 6.

```
 A=0.125*(sign(W)+sign(X))*abs((sign(W)+sign(Y))*(sign(W)+sign(Z)))*
min(min(abs(W),abs(X)),min(abs(Y),abs(Z)));
return
function B=DMM(X,Y)
 B = 0.5*(sign(X)+sign(Y))*min(abs(X),abs(Y));
return
```

References

1. Balsara, D.S. and Shu, C.W. *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, Journal of Computational Physics, 160 (2000), 405–452.

2. Burman, E. *A monotonicity preserving, nonlinear, finite element upwind method for the transport equation*, Applied Mathematics Letters, 49 (2015), 141–146.

3. Capdeville, G. *A high-order monotonicity-preserving scheme for hyperbolic conservation laws*, Computers & Fluids, 144 (2017), 86–116.

4. Crandall, M.G. and Andrew, M. *Monotone difference approximations for scalar conservation laws,* Mathematics of Computation, 149 (1980), 1–21.

5. Crandall, M. and Majda, A. *The method of fractional steps for conservation laws*, Numerische Mathematik, 34 (1980), 285–314.

6. Daru, V. and Tenaud, C. *High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations*, Journal of Computational Physics, 193 (2004), 563–594.

7. Du, Q., Huang, Z. and LeFloch, P.G. *Nonlocal conservation laws. A new class of monotonicity-preserving models*, SIAM Journal on Numerical Analysis, 55 (2017), 2465–2489.

8. Evans, L.C. *Partial differential equations*, American Mathematical Society, 1998.

9. Fang, J., Yao, Y., Li, Z. and Lu, L. *Investigation of low-dissipation monotonicity-preserving scheme for direct numerical simulation of compressible turbulent flows*, Computers & Fluids, 104 (2014), 55–72.

10. Farzi, J. *Global error estimation of linear multistep methods through the Runge-Kutta methods,* Iranian Journal of Numerical Analysis and Optimization, 6 (2016), 99–120.

11. Farzi, J., Hosseini, S.M. *High order immersed interface method for acoustic wave equation with discontinuous coefficients, Iranian Journal of Numerical Analysis and Optimization*, 4 (2014), 1–24.

12. Farzi, J. and Khodadosti, F. *A total variation diminishing high resolution scheme for nonlinear conservation laws,* Computational Methods for Differential Equations, 6 (2018), 456–470.

13. Gardner, L.R. and Dag, I. *A numerical solution of the advection-diffusion equation using B-spline finite element*, Proceedings International AMSE Conference, Lyon, France, (1994), 109–116.

14. Ha, C.T. and Lee, J.H. *A modified monotonicity-preserving high-order scheme with application to computation of multi-phase flows*, Computers & Fluids, 197 (2020), 1–29.

15. He, Z., Li, X., Fu, D. and Ma, Y. *A 5th order monotonicity-preserving upwind compact difference scheme*, Science China Physics Mechanics and Astronomy, 54 (2011), 511–522.

16. He, Z., Zhang, Y., Gao, F., Li, X. and Tian, B. *An improved accurate monotonicity-preserving scheme for the Euler equations*, Computers & Fluids, 140 (2016), 1–10.

17. Higueras, I., Ketcheson, D.I. and Kocsis, T.A. *Optimal monotonicity-preserving perturbations of a given Runge-Kutta method*, Journal of Scientific Computing, 76 (2018), 1337–1369.

18. Holly, F.M. and Usseglio-Polatera, J.M. *Dispersion simulation in two-dimensional tidal flow*, Journal of Hydraulic Engineering, 110 (1984), 905–926.

19. Hundsdorfer, W. and Verwer, J.G. *A note on splitting errors for advection-reaction equations*, Applied Numerical Mathematics, 18 (1997), 191–199.

20. Hundsdorfer, W. and Verwer, J.G. *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer Science & Business Media, 2013.

21. Huynh, H.T. *A piecewise-parabolic dual-mesh method for the Euler equations*, In 12th Computational Fluid Dynamics Conference, 1995.

22. Huynh, H.T. *Schemes and constraints for advection*, In Fifteenth International Conference on Numerical Methods in Fluid Dynamics, Springer, Berlin, Heidelberg, (1997) 498–503.

23. Karlsen, K.H. and Risebro, N.H. *Corrected operator splitting for nonlinear parabolic equations*, SIAM Journal on Numerical Analysis, 37 (2000), 980–1003.

24. Khalsaraei, M.M. *An improvement on the positivity results for 2-stage explicit Runge-Kutta methods,* Journal of Computational and Applied Mathematics, 235 (2010) 137–143.

25. Khalsaraei, M.M. and Khodadosti, F. *2-stage explicit total variation diminishing preserving Runge-Kutta methods,* Computational Methods for Differential Equations, 1 (2013), 30–38.

26. Khalsaraei, M.M. and Khodadosti, F. *Qualitatively stability of nonstandard 2-stage explicit Runge–Kutta methods of order two,* Computational Mathematics and Mathematical Physics, 56 (2016), 235–242.

27. Koren, B. *A robust upwind discretization method for advection, diffusion and source terms*, In Numerical methods for advection-diffusion problems, Amsterdam: Centrum voor Wiskunde en Informatica, (1993), 117–138.

28. Langseth, J.O., Tveito, A. and Winther, R. *On the convergence of operator splitting applied to conservation laws with source terms*, SIAM Journal on Numerical Analysis, 33 (1996), 843-863.

29. LeVeque, R.J. *Finite volume methods for hyperbolic problems*, Cambridge University Press; 2002.

30. LeVeque, R.J. and Yee, H.C. *A study of numerical methods for hyperbolic conservation laws with stiff source terms*, Journal of Computational Physics, 86 (1990), 187–210.

31. Li, X.L., Fu, D.X., Ma, Y.W. and Liang, X. *Direct numerical simulation of compressible turbulent flows*, Acta Mechanica Sinica, 26 (2010), 795–806.

32. Matsushita, S. and Aoki, T. *A weakly compressible scheme with a diffuse-interface method for low Mach number two-phase flows*, Journal of Computational Physics, 376 (2019), 838–862.

33. Mickens, R.E. *Nonstandard finite difference models of differential equations*, world scientific, 1994.

34. Oruc, O., Bulut, F. and Esen, A. *A Haar wavelet-finite difference hybrid method for the numerical solution of the modified Burgers equation*, Journal of Mathematical Chemistry, 53 (2015), 1592–1607.

35. Ropp, D.L. and Shadid, J.N. *Stability of operator splitting methods for systems with indefinite operators: Advection-diffusion-reaction systems*, Journal of Computational Physics, 228 (2009), 3508–3516.

36. Shu, C.W. and Osher, S. *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, 77 (1988), 439–471.

37. Speth, R.L., Green, W.H., MacNamara, S. and Strang, G. *Balanced splitting and rebalanced splitting*, SIAM Journal on Numerical Analysis, 51 (2013), 3084–3105.

38. Strang, G. *On the construction and comparison of difference schemes*, SIAM Journal on Numerical Analysis, 5 (1968), 506–517.

39. Suresh, A. and Huynh, H.T. *Accurate monotonicity-preserving schemes with Runge-Kutta time stepping*, Journal of Computational Physics, 136 (1997), 83–99.

40. Ucar, Y., Yagmurlu, N.M. and Celikkaya, I. *Operator splitting for numerical solution of the modified Burgers' equation using finite element method*, Numerical Methods for Partial Differential Equations, 35 (2019), 478–492.

41. Yu, Y., Tian, B. and Mo, Z. *Hybrid monotonicity-preserving piecewise parabolic method for compressible Euler equations*, Computers & Fluids, 159 (2017), 1–8.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 95–106
DOI:10.22067/ijnao.2021.11325.0

Research Article

Toeplitz-like preconditioner for linear systems from spatial fractional diffusion equations

N. Akhoundi

Abstract

The article deals with constructing a Toeplitz-like preconditioner for linear systems arising from finite difference discretization of the spatial fractional diffusion equations. The coefficient matrices of these linear systems have an $S+L$ structure, where S is a symmetric positive definite (SPD) matrix and L satisfies $\text{rank}(L) \leq 2$. We introduce an approximation for the SPD part S, which is called P_S, and then we show that the preconditioner $P = P_S + L$ has the Toeplitz-like structure and its displacement rank is 6. The analysis shows that the eigenvalues of the corresponding preconditioned matrix are clustered around 1. Numerical experiments exhibit that the Toeplitz-like preconditioner can significantly improve the convergence properties of the applied iteration method.

AMS subject classifications (2020): 65F10, 35R05, 65F08, 65M06.

Keywords: Fractional diffusion equation; Toeplitz-like matrix; Krylov subspace methods; PGMRES.

1 Introduction

In this article, we aim to propose a Toeplitz-like preconditioner for solving the linear system resulting from a finite difference approximation of an initial-boundary value problem of the spatial fractional diffusion equations (FDEs) that were introduced in [4]. In this article, the authors show that applying the finite difference method to the FDEs leads to the following linear system:

$$Ax = b, \tag{1}$$

Received 15 June 2020; revised 25 October 2020; accepted 24 November 2020
Nasser Akhoundi
School of mathematics and computer science, Damghan university, Damghan, Iran. e-mail: akhoundi@du.ac.ir

where the coefficient matrix of the linear system (1) is a dense Toeplitz-like matrix. As we see in [4], the coefficient matrix of (1) can be rewritten as the sum of a symmetric positive definite matrix S and a low rank matrix L, that is, $A = S + L$ (for more details, see [4, 2]). The matrix S has the following structure:

$$S = \eta I_n + GD_1G^T + G^TD_2G, \tag{2}$$

where $\eta > 0$, G is a Toeplitz matrix, and D_1 and D_2 are diagonal matrices with positive diagonal elements (more details are available in Section 2). In [4], the authors proposed a preconditioner by replacing G with its Strang's circulant approximation and two matrices D_1 and D_2 by their scalar (identity matrix) approximation. They showed the superlinear convergence of their preconditioner, when D_1 and D_2 are scalar identity matrices. In [2], we introduced a preconditioner based on the approximation of S defined in (2), and we proved that the proposed preconditioner is strong even though D_1 and D_2 are not a multiple of the identity matrix. In the following, we give some definitions about Toeplitz-like matrices.

Toeplitz-like matrices are defined by means of a displacement operator. Given an $n \times n$ matrix A, we consider the down-shift matrix of order n, $Z_n = \left(e_2 \; e_3 \; \cdots \; e_n \; 0_n\right)$, and the displacement operator ∇ defined by

$$\nabla(A) = A - Z_nAZ_n^T. \tag{3}$$

We define the displacement rank of matrix A as $rank(\nabla(A))$. The matrix A is said to have the Toeplitz-like structure if its displacement rank is low compared with its order n, that is, $rank(\nabla(A)) = m \ll n$. If the displacement rank of A is m, then we can rewrite $\nabla(A)$ as $\nabla(A) = CD^T$, where $C, D \in \mathbb{R}^{n\times m}$. Two matrices C and D are called the generators of the matrix A.

Throughout this article, we use the following notations: Capital letters, boldface lowercase letters, and regular lowercase letters denote matrices, vectors, and scalars, respectively. Moreover, I_n denotes the identity matrix of order n, while J_n denotes the $n\times n$ exchange matrix $J_n = \text{antidiag}(1, 1, \ldots, 1)$. We denote by e_j the jth column of identity matrix, and $\mathbf{0}$ denotes the zero vector of proper dimensions.

The organization of this article is as follows. In Section 2, we present the new preconditioner for the discretized linear systems arising from FDEs. Some computational remarks are given in Section 3. Numerical experiments are provided in Section 4 to prove the performance of our preconditioner. Finally, in Section 5, some concluding remarks are provided.

2 Preconditioning technique

The finite difference discretization of FDEs was developed in [4]. By introducing

$$a_i^{(\alpha)} = (i+1)^{1-\alpha} - i^{1-\alpha}, \qquad i = 0, 1, 2, \ldots,$$
$$g_0^{(\alpha)} = \frac{a_0^{(\alpha)}}{\Gamma(2-\alpha)}, \quad g_k^{(\alpha)} = \frac{a_k^{(\alpha)} - a_{k-1}^{(\alpha)}}{\Gamma(2-\alpha)}, \qquad k = 1, 2, 3, \ldots,$$

and

$$\mathbf{a} = -\frac{1}{\Gamma(2-\alpha)} \begin{pmatrix} a_0^{(\alpha)} \\ a_1^{(\alpha)} \\ \vdots \\ a_{n-1}^{(\alpha)} \end{pmatrix}, \qquad \mathbf{g} = \begin{pmatrix} g_1^{(\alpha)} \\ g_2^{(\alpha)} \\ \vdots \\ g_n^{(\alpha)} \end{pmatrix}, \tag{4}$$

the authors showed that the coefficient matrix of resulting linear system by finite difference discretization of FDEs possesses $A = S+L \in \mathbb{R}^{n\times n}$ structure, where

$$S = \eta I_n + G^{(\alpha)} D_+ G^{(\alpha)T} + G^{(\alpha)T} D_- G^{(\alpha)}, \tag{5}$$
$$L = c_1 \mathbf{a}\mathbf{g}^T + c_2 J\mathbf{a}(J\mathbf{g})^T, \tag{6}$$

in which $\eta > 0$, $D_\pm = \mathrm{diag}(d_{\pm,1}, d_{\pm,2}, \ldots, d_{\pm,n})$ with $d_{\pm,i} > 0$, $c_1, c_2 \geq 0$, and

$$G^{(\alpha)} = \begin{pmatrix} g_0^{(\alpha)} & 0 & \cdots & 0 \\ g_1^{(\alpha)} & g_0^{(\alpha)} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ g_{n-1}^{(\alpha)} & \cdots & g_1^{(\alpha)} & g_0^{(\alpha)} \end{pmatrix} \in \mathbb{R}^{n\times n}. \tag{7}$$

For simplicity, in the rest of the the superscript, (α) will be ignored. We see that the matrix S is a symmetric positive definite and that L is a matrix with low rank, that is, $rank(L) \leq 2$. Denote by $d_{\pm,\min}$ and $d_{\pm,\max}$ the smallest and the largest elements of $D_\pm$, respectively. Letting $\bar{d}_\pm = \frac{d_{\pm,\max}+d_{\pm,\min}}{2}$, in [2], we used the following preconditioner :

$$P_S = \eta I_n + \bar{d}_+ G G^T + \bar{d}_- G^T G. \tag{8}$$

This preconditioner is an approximation of S. We can replace S in A by P_S, obtaining so-called TL preconditioner

$$P = P_S + L \tag{9}$$

for the matrix A.

Theorem 1. If P is defined as (9), then $P^{-1}A = M+N$, where M is similar to $P_S^{-1}S$ and $rank(N) \leq 2$.

Proof. We can extend $\mathcal{S} = \mathrm{span}\{\mathbf{g}, J\mathbf{g}\}$ to an orthonormal basis of $\mathbb{R}^n$. Let $V = \begin{pmatrix}\mathbf{v}_1 \ \mathbf{v}_2 \cdots \mathbf{v}_n\end{pmatrix}$ be the matrix representation of such orthonormal basis, such that $\mathbf{v}_1 = \frac{\mathbf{g}}{\|\mathbf{g}\|_2}$ and $\mathbf{v}_2 = \frac{J\mathbf{g}+h\mathbf{g}}{\|J\mathbf{g}+h\mathbf{g}\|_2}$, where $h = -\frac{(J\mathbf{g})^T\mathbf{g}}{\|\mathbf{g}\|_2^2}$. Suppose that V

is partitioned as $V = \begin{pmatrix} V_1 \ V_2 \end{pmatrix}$, where $V_1 = \begin{pmatrix} \mathbf{v}_1 \ \mathbf{v}_2 \end{pmatrix}$ and $V_2 = \begin{pmatrix} \mathbf{v}_3 \ \mathbf{v}_4 \cdots \mathbf{v}_n \end{pmatrix}$. From (6), we see that $LV_2 = 0$. We have

$$V^T P_S^{-1} L V = \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} P_S^{-1} L \begin{pmatrix} V_1 \ V_2 \end{pmatrix} = \begin{pmatrix} V_1^T P_S^{-1} L V_1 & 0 \\ V_2^T P_S^{-1} L V_1 & 0 \end{pmatrix}. \tag{10}$$

If we define $K = I_2 + V_1^T P_S^{-1} L V_1$ and

$$Q = V \begin{pmatrix} I_2 - K^{-1} V_1^T & 0 \\ -V_2^T P_S^{-1} L V_1 K^{-1} V_1^T & 0 \end{pmatrix} V^T,$$

then

$$\begin{aligned} (I + P_S^{-1} L)^{-1} &= V \left[I + V^T P_S^{-1} L V \right]^{-1} V^T \\ &= V \left[I + \begin{pmatrix} V_1^T P_S^{-1} L V_1 & 0 \\ V_2^T P_S^{-1} L V_1 & 0 \end{pmatrix} \right]^{-1} V^T \\ &= V \begin{pmatrix} K^{-1} & 0 \\ -V_2^T P_S^{-1} L V_1 K^{-1} & I_{n-2} \end{pmatrix} V^T \\ &= I + Q. \end{aligned}$$

We note that $QV_2 = 0$. By this assumption, we can prove our assertion in the following

$$\begin{aligned} P^{-1} A &= (P_S + L)^{-1} (S + L) = (I + P_S^{-1} L)^{-1} P_S^{-1} S (I + S^{-1} L) \\ &= (I + P_S^{-1} L)^{-1} P_S^{-1} S (I + P_S^{-1} L)(I + P_S^{-1} L)^{-1} (I + S^{-1} L). \end{aligned} \tag{11}$$

In (11), if we define $M = (I + P_S^{-1} L)^{-1} P_S^{-1} S (I + P_S^{-1} L)$, then

$$\begin{aligned} P^{-1} A &= M (I + Q)(I + S^{-1} L) \\ &= M + N, \end{aligned}$$

where $N = M(S^{-1} L + Q + Q S^{-1} L)$. We see that M is similar to $P_S^{-1} S$. It can be easily verified that $NV_2 = 0$. Hence $rank(N) \leq 2$. $\square$

Theorem 2. [2] If $\lambda \in \sigma(P_S^{-1} S)$, then $|1 - \lambda| < k < 1$, where $k = \frac{1}{\hat{\eta}+1} < 1$ with $\hat{\eta} = \frac{\eta}{2\kappa \|\tilde{G}\|_2^2}$ and $\kappa = \max\{\frac{d_{+,\max} + d_{+,\min}}{2}, \frac{d_{-,\max} + d_{-,\min}}{2}\}$.

Based on Theorems 1 and 2, the eigenvalues of the preconditioned matrix $P^{-1} A$ are clustered around one, and thus Krylov subspace methods with the proposed preconditioner coverage very quickly.

Remark 1. The main differences between our proposed preconditioner P and the proposed preconditioner in [2] P_S are as follows:

1- P_S is an approximation of S in (5), while P consists of two parts. The first part P_S is an approximation of the matrix S and the second one is the matrix L (the rest of the matrix A). Hence, we expect that P is more accurate than P_S.

2- In the next section, in order to show the computational efficiency of preconditioner P, we prove that P has the Toeplitz-like structure, and its generators will be computed precisely. By using them, we can use fast algorithms to compute P^{-1}. Hence, in comparison with the circulant approximation of P_S in [2], the P^{-1} is computed straightforward.

In Section 4, numerical experiments show that the efficiency of the proposed preconditioner P.

3 Some computational results

In this section, we show that the matrix P defined in (9) has the Toeplitz-like structure, and we construct its generators. To this end, we define the following temporary vectors:

$$\bar{\mathbf{g}} = \begin{pmatrix} g_0 \; g_1 \; \cdots \; g_{n-1} \end{pmatrix}^T \qquad \bar{t} = \begin{pmatrix} g_1 \; g_2 \; \cdots \; g_{n-1} \end{pmatrix}^T. \tag{12}$$

The following relations can be easily deduced from (4) and (12)

$$\mathbf{a} - \mathbf{g} = \frac{1}{\Gamma(2-\alpha)} \begin{pmatrix} -a_0 - (a_1 - a_0) \\ -a_1 - (a_2 - a_1) \\ \vdots \\ -a_{n-1} - (a_n - a_{n-1}) \end{pmatrix} = Z^T\mathbf{a} + a_n e_n, \tag{13}$$

$$\bar{\mathbf{g}} = g_0\mathbf{e}_1 + Z\mathbf{g}, \tag{14}$$

$$Z^T\bar{\mathbf{g}} = \mathbf{g} - g_n e_n = \begin{pmatrix} \bar{\mathbf{t}} \\ 0 \end{pmatrix}. \tag{15}$$

We use the following lemmas in our subsequent discussion.

Lemma 1. Let G be the $n \times n$ matrix defined in (7). Then the following relations hold:

(i) $\nabla(GG^T) = \bar{\mathbf{g}}\bar{\mathbf{g}}^T$,

(ii) $\nabla(G^TG) = \mathbf{e}_1\bar{\mathbf{g}}^TG + G^T\bar{\mathbf{g}}\mathbf{e}_1^T - \bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1\mathbf{e}_1^T - JZ^T\bar{\mathbf{g}}\bar{\mathbf{g}}^TZJ$,

(iii) $\nabla(P_S) = \eta\mathbf{e}_1\mathbf{e}_1^T + \bar{d}_+\bar{\mathbf{g}}\bar{\mathbf{g}}^T + \bar{d}_-(\mathbf{e}_1\bar{\mathbf{g}}^TG + G^T\bar{\mathbf{g}}\mathbf{e}_1^T - \bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1\mathbf{e}_1^T - JZ^T\bar{\mathbf{g}}\bar{\mathbf{g}}^TZJ)$.

Proof. The matrix G can be partitioned as

$$G = \begin{pmatrix} g_0 & \mathbf{0} \\ \bar{\mathbf{t}} & G_1 \end{pmatrix}, \tag{16}$$

where $G_1 = G(2:n, 2:n)$ is a lower triangular Toeplitz matrix, so the displacement structure of GG^T can be viewed as

$$\begin{aligned}\nabla(GG^T) &= GG^T - ZGG^TZ^T \\ &= \begin{pmatrix} g_0 & \mathbf{0} \\ \bar{\mathbf{t}} & G_1 \end{pmatrix}\begin{pmatrix} g_0 & \bar{\mathbf{t}}^T \\ \mathbf{0} & G_1^T \end{pmatrix} - \begin{pmatrix} \mathbf{0} & 0 \\ G_1 & \mathbf{0} \end{pmatrix}\begin{pmatrix} \mathbf{0} & G_1^T \\ 0 & \mathbf{0} \end{pmatrix}. \end{aligned} \tag{17}$$

Now, by a straightforward computation, part (i) of the lemma can be verified. For part (ii), first we know that

$$\bar{\mathbf{g}}^T G = \begin{pmatrix} \bar{\mathbf{g}}^T\bar{\mathbf{g}} & \bar{\mathbf{t}}^T G_1 \end{pmatrix}. \tag{18}$$

Therefore,

$$\nabla(G^TG) = G^TG - ZG^TGZ^T \tag{19}$$

$$= \begin{pmatrix} g_0 & \bar{\mathbf{t}}^T \\ 0 & G_1^T \end{pmatrix}\begin{pmatrix} g_0 & 0 \\ \bar{\mathbf{t}} & G_1 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ G_1^T & J\bar{\mathbf{t}} \end{pmatrix}\begin{pmatrix} 0 & G_1 \\ 0 & (J\bar{\mathbf{t}})^T \end{pmatrix} \tag{20}$$

$$= \begin{pmatrix} \bar{\mathbf{g}}^T\bar{\mathbf{g}} & \bar{\mathbf{t}}^T G_1 \\ G_1^T\bar{\mathbf{t}} & -J\bar{\mathbf{t}}\bar{\mathbf{t}}^T J \end{pmatrix}. \tag{21}$$

It gives that

$$\begin{aligned}\nabla(G^TG) &= \begin{pmatrix} \bar{\mathbf{g}}^T\bar{\mathbf{g}} & \bar{\mathbf{t}}^T G_1 \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} \end{pmatrix} + \begin{pmatrix} \bar{\mathbf{g}}^T\bar{\mathbf{g}} & \bar{\mathbf{0}} \\ G_1^T\bar{\mathbf{t}} & \bar{\mathbf{0}} \end{pmatrix} - \begin{pmatrix} \bar{\mathbf{g}}^T\bar{\mathbf{g}} & \bar{\mathbf{0}} \\ \bar{\mathbf{0}} & \bar{\mathbf{0}} \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & J\bar{\mathbf{t}}^T\bar{\mathbf{t}}J \end{pmatrix} \\ &= \mathbf{e}_1\bar{\mathbf{g}}^T G + G^T\bar{\mathbf{g}}\mathbf{e}_1^T - \bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1\mathbf{e}_1^T - JZ^T\bar{\mathbf{g}}\bar{\mathbf{g}}^T ZJ. \end{aligned} \tag{22}$$

We note that $\nabla(I) = \mathbf{e}_1\mathbf{e}_1^T$, so

$$\begin{aligned}\nabla(P_S) &= \eta\mathbf{e}_1\mathbf{e}_1^T + \bar{d}_+\nabla(GG^T) + \bar{d}_-\nabla(G^TG) \\ &= \eta\mathbf{e}_1\mathbf{e}_1^T + \bar{d}_+\bar{\mathbf{g}}\bar{\mathbf{g}}^T + \bar{d}_-(\mathbf{e}_1\bar{\mathbf{g}}^T G + G^T\bar{\mathbf{g}}\mathbf{e}_1^T - \bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1\mathbf{e}_1^T - JZ^T\bar{\mathbf{g}}\bar{\mathbf{g}}^T ZJ), \end{aligned}$$

which completes the proof of (iii). □

Lemma 2. If we define the auxiliary vectors $\mathbf{h}_1 = \bar{d}_-(J\mathbf{g}) - c_2Z(J\mathbf{g})$ and $\mathbf{h}_2 = c_1\mathbf{g} - \bar{d}_+(Z\mathbf{g})$, then the following relations hold:

(i) $c_2\left[(J\mathbf{a})(J\mathbf{g})^T - Z(J\mathbf{a})(J\mathbf{g})^TZ^T\right] - \bar{d}_-JZ^T\bar{\mathbf{g}}\bar{\mathbf{g}}^TZJ = (c_2 - \bar{d}_-)(J\mathbf{a})(J\mathbf{g})^T + Z(J\mathbf{a})\mathbf{h}_1^T + \mathbf{e}_1(\bar{d}_-a_n + \bar{d}_-g_n)(J\mathbf{g})^T + \bar{d}_-(g_n(J\mathbf{g}) - g_n^2\mathbf{e}_1)\mathbf{e}_1^T$,

(ii) $c_1\left[(\mathbf{a})(\mathbf{g})^T - Z(\mathbf{a})(\mathbf{g})^TZ^T\right] + \bar{d}_+\bar{\mathbf{g}}\bar{\mathbf{g}}^T = (\bar{d}_+ - c_1)(Z\mathbf{a})(Z\mathbf{g})^T + \mathbf{a}\mathbf{h}_2^T + \mathbf{e}_1(a_0\bar{d}_+ + \bar{d}_+g_0)(Z\mathbf{g})^T + (\bar{d}_+g_0(Z\mathbf{g}) + \bar{d}_+g_0^2\mathbf{e}_1)\mathbf{e}_1^T$.

Proof. We see that

$$
\begin{aligned}
& c_2\left[(J\mathbf{a})(J\mathbf{g})^T - Z(J\mathbf{a})(J\mathbf{g})^T Z^T\right] - \bar{d}_- J Z^T \bar{\mathbf{g}}\bar{\mathbf{g}}^T Z J \\
&= c_2\left[(J\mathbf{a})(J\mathbf{g})^T - Z(J\mathbf{a})(J\mathbf{g})^T Z^T\right] - \bar{d}_-\left[J(\mathbf{g} - g_n\mathbf{e}_n)(\mathbf{g} - g_n\mathbf{e}_n)^T J\right] \qquad \text{By (15)} \\
&= \left[c_2(J\mathbf{a}) - \bar{d}_-(J\mathbf{g})\right](J\mathbf{g})^T - c_2 Z(J\mathbf{a})(J\mathbf{g})^T Z^T \\
&\quad + \bar{d}_- g_n\mathbf{e}_1(J\mathbf{g})^T + \bar{d}_- g_n(J\mathbf{g})\mathbf{e}_1^T - \bar{d}_- g_n^2\mathbf{e}_1\mathbf{e}_1^T \\
&= J\left[c_2\mathbf{a} - \bar{d}_-(\mathbf{a} - Z^T\mathbf{a} - a_n\mathbf{e}_n)\right](J\mathbf{g})^T - c_2 Z(J\mathbf{a})(J\mathbf{g})^T Z^T \qquad \text{By (13)} \\
&\quad + \bar{d}_- g_n\mathbf{e}_1(J\mathbf{g})^T + \bar{d}_- g_n(J\mathbf{g})\mathbf{e}_1^T - \bar{d}_- g_n^2\mathbf{e}_1\mathbf{e}_1^T \\
&= (c_2 - \bar{d}_-)(J\mathbf{a})(J\mathbf{g})^T + Z(J\mathbf{a})\mathbf{h}_1^T + \mathbf{e}_1(\bar{d}_- a_n + \bar{d}_- g_n)(J\mathbf{g})^T \\
&\quad + (\bar{d}_- g_n(J\mathbf{g}) - \bar{d}_- g_n^2\mathbf{e}_1)\mathbf{e}_1^T.
\end{aligned}
$$

In the similar way, we can prove part (ii) as follows:

$$
\begin{aligned}
& c_1\left[(\mathbf{a})(\mathbf{g})^T - Z(\mathbf{a})(\mathbf{g})^T Z^T\right] + \bar{d}_+\bar{\mathbf{g}}\bar{\mathbf{g}}^T \\
&= c_1\left[(\mathbf{a})(\mathbf{g})^T - Z(\mathbf{a})(\mathbf{g})^T Z^T\right] + \bar{d}_+(g_0\mathbf{e}_1 + Z\mathbf{g})(g_0\mathbf{e}_1 + Z\mathbf{g})^T \qquad \text{By (14)} \\
&= Z\left[\bar{d}_+\mathbf{g} - c_1\mathbf{a}\right](Z\mathbf{g})^T + c_1\mathbf{a}\mathbf{g}^T + \bar{d}_+\left[g_0^2\mathbf{e}_1\mathbf{e}_1^T + g_0\mathbf{e}_1(Z\mathbf{g})^T + g_0 Z\mathbf{g}\mathbf{e}_1^T\right] \\
&= Z\left[(\bar{d}_+ - c_1)\mathbf{a} - \bar{d}_+ Z^T\mathbf{a} - \bar{d}_+ a_n\mathbf{e}_n\right](Z\mathbf{g})^T + c_1\mathbf{a}\mathbf{g}^T \qquad \text{By (13)} \\
&\quad + \bar{d}_+\left[g_0^2\mathbf{e}_1\mathbf{e}_1^T + g_0\mathbf{e}_1(Z\mathbf{g})^T + g_0 Z\mathbf{g}\mathbf{e}_1^T\right] \\
&= (\bar{d}_+ - c_1)(Z\mathbf{a})(Z\mathbf{g})^T + \mathbf{a}\mathbf{h}_2^T + \mathbf{e}_1(a_0\bar{d}_+ + \bar{d}_+ g_0)(Z\mathbf{g})^T \\
&\quad + (\bar{d}_+ g_0(Z\mathbf{g}) + \bar{d}_+ g_0^2\mathbf{e}_1)\mathbf{e}_1^T.
\end{aligned}
$$

□

Theorem 3 indicates that P in (9) has the Toeplitz-like structure, and we can use a superfast solver to compute P^{-1}.

Theorem 3. The matrix P defined in (8) is a Toeplitz-like matrix and its displacement operator is

$$
\begin{aligned}
\nabla(P) =& (c_2 - \bar{d}_-)(J\mathbf{a})(J\mathbf{g})^T + (\bar{d}_+ - c_1)(Z\mathbf{a})(Z\mathbf{g})^T + Z(J\mathbf{a})\mathbf{h}_1^T \\
& + \mathbf{a}\mathbf{h}_2^T + \mathbf{e}_1\mathbf{h}_3^T + \mathbf{h}_4\mathbf{e}_1^T,
\end{aligned} \tag{23}
$$

where $\mathbf{h}_1$ and $\mathbf{h}_2$ are defined in Lemma 2, and

$$
\begin{aligned}
\mathbf{h}_3 &= (\bar{d}_- a_n + \bar{d}_- g_n)(J\mathbf{g}) + \bar{d}_+(a_0 + g_0)Z\mathbf{g} + \eta\mathbf{e}_1 + \bar{d}_- G^T\bar{\mathbf{g}}, \\
\mathbf{h}_4 &= \bar{d}_- g_n(J\mathbf{g}) - g_n^2\mathbf{e}_1 + \bar{d}_+ g_0(Z\mathbf{g}) + \bar{d}_+ g_0^2\mathbf{e}_1 + \bar{d}_- G^T\bar{\mathbf{g}} - \bar{d}_-\bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1.
\end{aligned}
$$

Proof. We have $P = P_S + L$. Hence

$$
\begin{aligned}
\nabla(P) =& \nabla(P_S) + \nabla(L) = \nabla(P_S) + c_1((a)(g)^T - Z(a)(Z\mathbf{g})^T) \\
& + c_2(J\mathbf{a}(J\mathbf{g}^T) - ZJ\mathbf{a}(ZJ\mathbf{g})^T).
\end{aligned} \tag{24}
$$

By part (iii) of Lemma 1, we can rewrite (24) as follows:

$$\begin{aligned}\nabla(P) &= \eta\mathbf{e}_1\mathbf{e}_1^T + \bar{d}_+\bar{\mathbf{g}}\bar{\mathbf{g}}^T + \bar{d}_-(\mathbf{e}_1\bar{\mathbf{g}}^T G + G^T\bar{\mathbf{g}}\mathbf{e}_1^T - \bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1\mathbf{e}_1^T - JZ^T\bar{\mathbf{g}}\bar{\mathbf{g}}^T ZJ)\\ &\quad + c_1((a)(g)^T - Z(a)(Z\mathbf{g})^T) + c_2(J\mathbf{a}(J\mathbf{g}^T) - ZJ\mathbf{a}(ZJ\mathbf{g})^T)\\ &= c_2\left[(J\mathbf{a})(J\mathbf{g})^T - Z(J\mathbf{a})(J\mathbf{g})^T Z^T\right] - \bar{d}_- JZ^T\bar{\mathbf{g}}\bar{\mathbf{g}}^T ZJ\\ &\quad + c_1\left[(\mathbf{a})(\mathbf{g})^T - Z(\mathbf{a})(\mathbf{g})^T Z^T\right] + \bar{d}_+\bar{\mathbf{g}}\bar{\mathbf{g}}^T\\ &\quad + \eta\mathbf{e}_1\mathbf{e}_1^T + \bar{d}_-(\mathbf{e}_1\bar{\mathbf{g}}^T G + G^T\bar{\mathbf{g}}\mathbf{e}_1^T - \bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1\mathbf{e}_1^T)\end{aligned}$$

Parts (i) and (ii) of Lemma 2 and the above equations imply that

$$\begin{aligned}\nabla(P) =& (c_2 - \bar{d}_-)(J\mathbf{a})(J\mathbf{g})^T + Z(J\mathbf{a})\mathbf{h}_1^T + \mathbf{e}_1(\bar{d}_- a_n + \bar{d}_- g_n)(J\mathbf{g})^T\\ &+ (\bar{d}_- g_n(J\mathbf{g}) - g_n^2\mathbf{e}_1)\mathbf{e}_1^T + (\bar{d}_+ - c_1)(Z\mathbf{a})(Z\mathbf{g})^T + \mathbf{a}\mathbf{h}_2^T\\ &+ \mathbf{e}_1(a_0\bar{d}_+ + \bar{d}_+ g_0)(Z\mathbf{g})^T + (\bar{d}_+ g_0(Z\mathbf{g}) + \bar{d}_+ g_0^2\mathbf{e}_1)\mathbf{e}_1^T\\ &+ \eta\mathbf{e}_1\mathbf{e}_1^T + \bar{d}_-(\mathbf{e}_1\bar{\mathbf{g}}^T G + G^T\bar{\mathbf{g}}\mathbf{e}_1^T - \bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1\mathbf{e}_1^T)\\ =& (c_2 - \bar{d}_-)(J\mathbf{a})(J\mathbf{g})^T + Z(J\mathbf{a})\mathbf{h}_1^T + (\bar{d}_+ - c_1)(Z\mathbf{a})(Z\mathbf{g})^T + \mathbf{a}\mathbf{h}_2^T\\ &+ \mathbf{e}_1\left((\bar{d}_- a_n + \bar{d}_- g_n)(J\mathbf{g}) + \bar{d}_+(a_0 + g_0)Z\mathbf{g} + \eta\mathbf{e}_1 + \bar{d}_- G^T\bar{\mathbf{g}}\right)^T\\ &+ \left(\bar{d}_- g_n(J\mathbf{g}) - g_n^2\mathbf{e}_1 + \bar{d}_+ g_0(Z\mathbf{g}) + \bar{d}_+ g_0^2\mathbf{e}_1 + \bar{d}_- G^T\bar{\mathbf{g}} - \bar{d}_-\bar{\mathbf{g}}^T\bar{\mathbf{g}}\mathbf{e}_1\right)\mathbf{e}_1^T\\ =& (c_2 - \bar{d}_-)(J\mathbf{a})(J\mathbf{g})^T + (\bar{d}_+ - c_1)(Z\mathbf{a})(Z\mathbf{g})^T\\ &+ Z(J\mathbf{a})\mathbf{h}_1^T + \mathbf{a}\mathbf{h}_2^T + \mathbf{e}_1\mathbf{h}_3^T + \mathbf{h}_4\mathbf{e}_1^T.\end{aligned}$$

□

We showed that our preconditioner P has the Toeplitz-like structure and its displacement rank of P ($rank(\nabla(P))$) is 6. So we can use fast and stable Levinson like algorithms [5] to compute $P^{-1}r$ in $O(n^2)$ operations. For Krylov subspace iteration methods such as preconditioned GMRES [10], high quality preconditioning plays a crucial role in accelerating the convergence speed of the Krylov subspace iteration methods. Instead of $O(n^2)$ operations to apply our preconditioner in each iteration, Circulant preconditioners require only $O(n\log(n))$ operations in each step. Significant reduction of total iterations (and total CPU time) by using our preconditioner in numerical experiments show the efficiency of our preconditioner.

4 Numerical experiments

The GMRES(20) (restarts every 20 inner iterations) with the proposed preconditioner is applied to solve the linear system (1). We choose the right-hand side such that $x^* = (1, 2 \ldots, n)^T$ is the exact solution, that is, $b = Ax^*$. The stopping criterion in the numerical experiments is $\|r_k\|_2/\|b\|_2 < 1e{-}7$, where r_k is the residual vector of the linear system after k iterations and b is the right-hand side. For all experiments, the initial guess is chosen as the zero

Table 1: Numerical results for Example 1

α	n	I		P_C		P_S		TL	
		Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
0.7	2^{10}	81	23.5113	6	0.6189	4	0.4012	1	0.3001
	2^{11}	†	-	7	2.1718	5	1.6312	1	0.9017
	2^{12}	†	-	7	7.9142	5	5.1305	1	2.7321
	2^{13}	†	-	8	9.8320	5	6.3483	1	5.9821
	2^{14}	†	-	11	14.8639	7	9.6407	1	8.9231
0.9	2^{10}	†	-	13	1.4921	8	1.2134	1	0.5743
	2^{11}	†	-	21	9.5901	13	5.8513	1	1.2401
	2^{12}	†	-	29	46.2163	17	26.2245	1	3.5421
	2^{13}	†	-	†	-	†	-	2	7.3142
	2^{14}	†	-	†	-	†	-	2	10.6893

vector. All the numerical experiments are run in MATLAB on a desktop with the configuration: Intel(R) Core(TM) CPU Q9450 2.66 GHz and 8.00 GB RAM. In the following tables, "I" represents the GMRES method without preconditioning technique. We use P_C to denote the circulant preconditioner [4], P_S to denote preconditioner in [2], and TL to present the proposed preconditioner defined in (9). Also the "Iter" denotes the number of iterations, "CPU" denotes the total CPU time in seconds for solving the problem, and n denotes the size of our linear system. In the tables, † indicates no convergence within 100 iterations.

Example 1. In this example, we consider

$$D_+ = \text{diag}\{0, (\frac{1}{n-1})^{1-\alpha}, (\frac{2}{n-1})^{1-\alpha}, \ldots, 1\} \tag{25}$$

$$D_- = \text{diag}\{1, (\frac{n-2}{n-1})^{1-\alpha}, (\frac{n-3}{n-1})^{1-\alpha}, \ldots, 0\}, \tag{26}$$

and $c_1 = c_2 = 2$.

Example 2. In this example, $D_\pm$ are random diagonal matrices with diagonal entries taken independently at random from $[0, 1]$ and $c_1 = c_2 = 2$.

The numerical results of Examples 1 and 2 are shown in Tables 1 and 2, respectively. Tables 1–2 show clearly the fast convergence of the TL preconditioner. We see that TL preconditioner is more effective in both CPU time and the number of iterations than P_S and P_C preconditioners.

In what follows, we compare the eigenvalues of A and $P^{-1}A$ for Example 1, the results are shown in Figures 1 and 2. As we see the eigenvalues of preconditioned matrix $P^{-1}A$ are clustered around 1.

Table 2: Numerical results for Example 2

α	n	I		P_C		P_S		TL	
		Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
0.7	2^{10}	94	0.5931	6	0.6907	4	0.4455	1	0.2517
	2^{11}	†	-	7	1.7012	5	1.3259	1	0.8993
	2^{12}	†	-	7	8.2302	5	6.4216	1	2.3263
	2^{13}	†	-	9	11.4134	6	8.1983	1	7.3129
	2^{14}	†	-	11	14.3921	7	11.0214	2	10.9386
0.9	2^{10}	†	-	30	2.3102	17	1.7492	1	0.4352
	2^{11}	†	-	41	14.0561	25	10.9318	1	1.3107
	2^{12}	†	-	75	47.5169	44	20.9312	1	4.3563
	2^{13}	†	-	†	-	†	-	2	9.8311
	2^{14}	†	-	†	-	†	-	2	14.0968

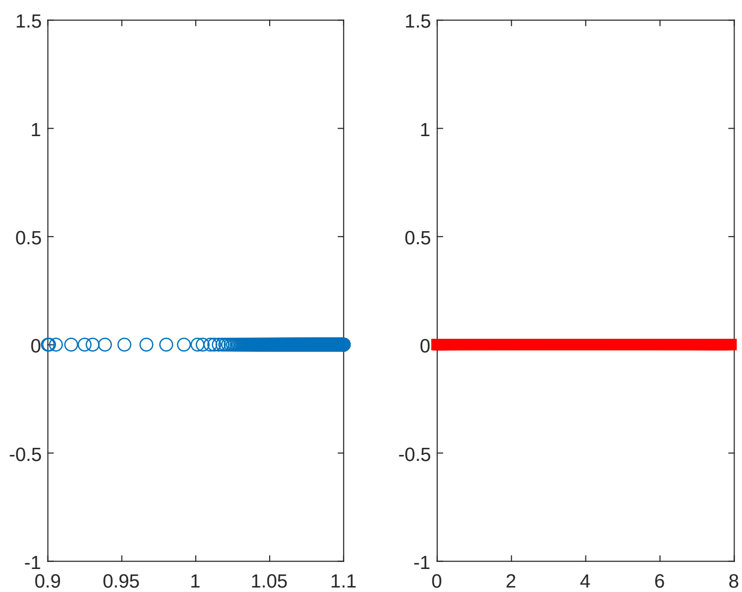


Figure 1: Eigenvalues of $P^{-1}A$ (left) and A (right) for Example 1, with $\alpha = .9$ and $n = 2^{10}$.

5 Concluding remarks

In this article, the preconditioned GMRES method with Toeplitz-like structure was employed to solve the discretized linear systems arising from FDEs. The displacement structure of the Toeplitz-like preconditioner was computed

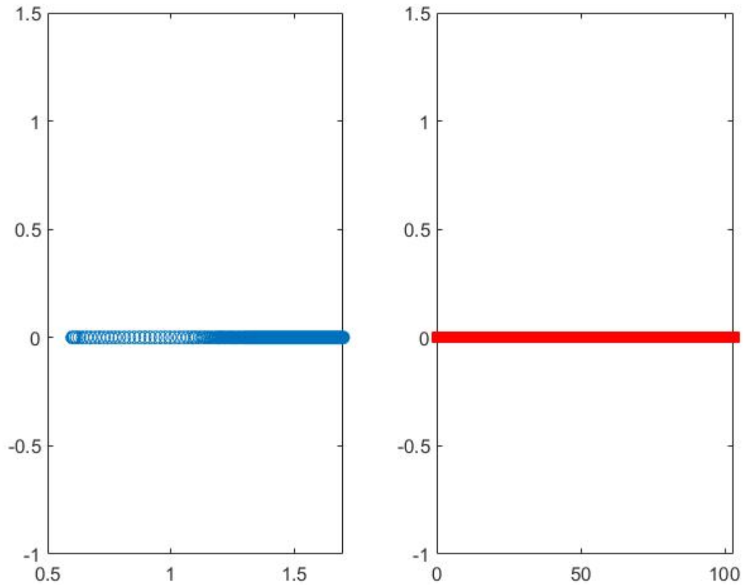


Figure 2: Eigenvalues of $P^{-1}A$ (left) and A (right) for Example 1, with $\alpha = .9$ and $n = 2^{12}$.

precisely. The efficiency of the proposed preconditioner was proved even though the diffusion coefficients are not constants. Numerical experiments have demonstrated the efficiency of the proposed preconditioner.

Nevertheless, it is interesting that to propose the possible Toeplitz-like preconditioner for two-dimensional FDEs, and then examine the efficiency of the proposed preconditioner.

References

1. Agrawal, O.P., Solution for a fractional diffusion-wave equation defined in a bounded domain, Nonlinear Dynam. 29(1-4) (2002) 145–155.

2. Akhoundi N., $2n$-by-$2n$ circulant preconditioner for a kind of spatial fractional diffusion equations, J. Math. Model. 8 (2020), 207–218.

3. Bouchaud, J.P. and Georges A., Anomalous diffusion in disordered media: statistical mechanisms, models and physical applications, Phys. Rep. 195 (1990), 127–293.

4. Fang, Z.W., Ng, M.K. and Sun, H.W., Circulant preconditioners for a kind of spatial fractional diffusion equations, Numer. Algorithms 82 (2019), 729–747.

5. Favat, P., Lotti, G. and Menchi, O., Stability of the Levinson algorithm for Toeplitz-like systems, SIAM J. Matrix Anal. Appl. 31 (2010), 2531–2552.

6. Kilbas, A.A., Srivastava, H.M. and Trujillo, J.J., Theory and applications of fractional differential equations, North-Holland Mathematics Studies, 204. Elsevier Science B.V., Amsterdam, 2006.

7. Metzler, R. and Klafter, J., The random walk's guide to anomalous diffusion: a fractional dynamics approach. Phys. Rep. 339 (2000), 77 pp.

8. Podlubny, I., Fractional differential equations: An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications, Elsevier; 1998.

9. Rudolf, H., Editor. Applications of fractional calculus in physics, World Scientific Publishing Co., Inc., River Edge, NJ, 2000.

10. Saad, Y., Preconditioned Krylov subspace methods for the numerical solution of Markov chains, In: Stewart W.J. (eds) Computations with Markov Chains. Springer, Boston, MA.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 107–115
DOI:10.22067/ijnao.2021.11329.0

Research Article

A high-order algorithm for solving nonlinear algebraic equations

A. Ghorbani* and M. Gachpazan

Abstract

A fourth-order and rapid numerical algorithm, utilizing a procedure as Runge–Kutta methods, is derived for solving nonlinear equations. The method proposed in this article has the advantage that it, requiring no calculation of higher derivatives, is faster than the other methods with the same order of convergence. The numerical results obtained using the developed approach are compared to those obtained using some existing iterative methods, and they demonstrate the efficiency of the present approach.

AMS subject classifications (2020): 65H05, 65L99.

Keywords: Order of convergence; Newton–Raphson method; Householder iteration method; Nonlinear equations.

1 Introduction

The development of numerical techniques for solving nonlinear (algebraic or transcendental) equations (NEs) is a subject of considerable interest. A vast amount of research work has been invested in the study of the NEs. Solution of these equations can be obtained using classical numerical methods as Newton–Raphson method (NRM) or the Householder iteration method (HIM); see [6]. There are many articles that deal with NEs, for example, [1, 2, 3, 4, 5]. A more extensive list of references as well as a survey on the progress made on this class of problems may be found in [8]. However, in order to establish a more accurate algorithm, one has to calculate higher

*Corresponding author
Received 28 August 2020; revised 6 October 2020; accepted 29 November 2020
Asghar Ghorbani
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: aghorbani@um.ac.ir

Mortaza Gachpazan
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: gachpazan@um.ac.ir

order derivatives, for example, like the HIM, which can be considered a serious drawback.

Here, we establish a fourth-order fast algorithm for finding an accurate solution of the NEs with the advantage that there is no calculation of higher derivatives. Moreover, the merit in this method can be observed due to high exactness, lesser number of iterations, and lesser value of errors as compared to the other existing methods. The examples analyzed here reveal that the developed algorithm is effective to solve the NEs.

2 Basic idea of the new algorithm

Consider the nonlinear algebraic equation

$$f(x) = 0. \tag{1}$$

We assume that α is a simple zero of (1), that f is a C^2 function on an interval containing α, and that λ is an initial guess sufficiently close to α. Also we suppose that $|f'(\alpha)| > 0$. Using Taylor's series around λ for (1), we have

$$f(\lambda) + (x-\lambda)f'(\lambda) + \frac{1}{2}(x-\lambda)^2 f''(\lambda) + \frac{1}{6}(x-\lambda)^3 f'''(\lambda) + \cdots = 0. \tag{2}$$

We can rewrite (2) in the following form:

$$x = c + N(x), \tag{3}$$

where

$$c = \lambda - \frac{f(\lambda)}{f'(\lambda)} \tag{4}$$

and

$$N(x) = -\frac{1}{2}(x-\lambda)^2 \frac{f''(\lambda)}{f'(\lambda)}. \tag{5}$$

The NRM is then given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{6}$$

where x_0 is the initial guess. The iteration (6) will converge to α if the starting point x_0 is close enough to α. This process has the second-order convergence.

The HIM is given by (see [6])

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n) f''(x_n)}{2f'^3(x_n)}. \tag{7}$$

The iteration (7) with the third-order convergence will converge to α if the starting point x_0 is close enough to α.

As observed in the above-mentioned methods, for obtaining a more accurate solution of α, we have to take n, order iteration, as large as possible, or go forward with the calculation of higher derivatives. In order to remove these demerits, here we look for a scheme without computing higher order iterations/derivatives that contain high accuracy. To illustrate the idea behind the developing algorithm, we construct an iterative procedure in the following form:

$$x_{n+1} = x_n + Ak_1 + Bk_2, \tag{8}$$

where

$$k_1 = \frac{f(x_n)}{f'(x_n)}, \tag{9}$$

$$k_2 = \frac{f(x_n + Ck_1)}{f'(x_n)}. \tag{10}$$

In (8)–(10), the coefficients A, B, and C are constants to be determined later. We note that the NRM (6) can be obtained from (8) for the values $A = -1$ and $B = C = 0$.

Now, we want to determine the constants of A, B, and C so that the formula (8) is similar to the formula (7) of accuracy. From (8), therefore, using Taylor's expansion, we have

$$x_{n+1} = x_n + (A + B + BC)\frac{f(x_n)}{f'(x_n)} + BC^2\frac{f^2(x_n)f''(x_n)}{2f'^3(x_n)}. \tag{11}$$

Comparing (7) and (11) and equating the coefficients with identical terms, we get

$$\begin{cases} A + B + BC = -1, \\ BC^2 = -1. \end{cases} \tag{12}$$

The solutions of (12) allow us to suggest different iterative methods (of at least third-order) for solving the NEs of (1). Two solutions for the system (12) are $A = 0$, $B = -\frac{3+\sqrt{5}}{2}$, $C = \frac{1-\sqrt{5}}{2}$, and $A = B = C = -1$. Thus, substituting the later solution into (8)–(10), we will have the following iterative algorithm:

$$\begin{cases} x_{n+1} = x_n - [k_1 + k_2], \\ k_1 = \dfrac{f(x_n)}{f'(x_n)}, \\ k_2 = \dfrac{f(x_n - k_1)}{f'(x_n)}. \end{cases} \tag{13}$$

Now, if, according to the definition $f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$, we estimate the value of first derivative by placing $h \simeq -\frac{f(x_n)}{f'(x_n)}$ as

$$f'(x) \approx \frac{f'(x_n)\left(f(x_n) - f\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)\right)}{f(x_n)}, \tag{14}$$

then we will have

$$\begin{cases} x_{n+1} = x_n - [k_1 + k_2], \\ k_1 = \dfrac{f^2(x_n)}{f'(x_n)\left(f(x_n) - f\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)\right)}, \\ k_2 = \dfrac{k_1 f(x_n - k_1)}{f(x_n)}. \end{cases} \tag{15}$$

Now the second algorithm could be gained by substituting the former solution into (8)–(10) as

$$\begin{cases} x_{n+1} = x_n - \dfrac{3+\sqrt{5}}{2} k_2, \\ k_1 = \dfrac{f(x_n)}{f'(x_n)}, \\ k_2 = \dfrac{f(x_n + \frac{1-\sqrt{5}}{2} k_1)}{f'(x_n)}. \end{cases} \tag{16}$$

Remark 1. As we know well, these kinds of algorithms need a good initial guess to work. This shortcoming can be readily removed by finding subintervals on x that contain sign changes of $f(x)$ (observe the appendix section).

Remark 2. From (15) and (16), we can comprehend that in each iteration of the algorithm (15), we need to perform four function evaluations, and for the algorithm (16), there are three function evaluations. However, the fast convergence of these methods often compensates for these defects.

3 Analysis of convergence

Before proving the convergence of the algorithms (15) and (16), here the following definition is given.

Definition 1. Let $e_n = x_n - \alpha$ be the truncation error in the nth iterate. If there exist a number $\beta \geq 1$ and a constant $c \neq 0$ such that

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^{\beta}} = c, \tag{17}$$

then β is called the order of convergence of the method.

Now, we consider the convergence of the algorithms (15) and (16).

Theorem 1. Consider the nonlinear algebraic equation $f(x) = 0$. Suppose that f is sufficiently differentiable. Then the convergence of the algorithm (16) is at least of order 3.

Proof. Let α be a simple zero of $f(x)$. Since $f(x)$ is sufficiently differentiable, by expanding $f(x)$, $f'(x)$, $f''(x)$, and $f'''(x)$ around α, we get

$$\begin{aligned}
f(x_n) &= f'(\alpha)\left[e_n + d_2e_n^2 + d_3e_n^3 + d_4e_n^4 + d_5e_n^5 + \cdots\right],\\
f'(x_n) &= f'(\alpha)\left[1 + 2d_2e_n + 3d_3e_n^2 + 4d_4e_n^3 + 5d_5e_n^4 + 6d_6e_n^5 + \cdots\right],\\
f''(x_n) &= f'(\alpha)\left[2d_2 + 6d_3e_n + 12d_4e_n^2 + 20d_5e_n^3 + 30d_6e_n^4 + 42d_7e_n^5 + \cdots\right],\\
f'''(x_n) &= f'(\alpha)\left[6d_3 + 24d_4e_n + 60d_5e_n^2 + 120d_6e_n^3 + 210d_7e_n^4 + 336d_8e_n^5 + \cdots\right],
\end{aligned} \tag{18}$$

where $d_n = \frac{1}{n!}\frac{f^{(n)}(\alpha)}{f'(\alpha)}$, $n = 2, 3, \ldots$, and $e_n = x_n - \alpha$. We can rewrite (16) as follows:

$$e_{n+1} = e_n + Bf\left(x_n + C\frac{f(x_n)}{f'(x_n)}\right), \tag{19}$$

or using Taylor's series as

$$e_{n+1} = e_n + B\left[(1+C)f(x_n) + \frac{C^2}{2}\left(\frac{f(x_n)}{f'(x_n)}\right)^2 f''(x_n) + \frac{C^3}{6}\left(\frac{f(x_n)}{f'(x_n)}\right)^3 f'''(x_n)\right], \tag{20}$$

where $B = -\frac{3+\sqrt{5}}{2}$ and $C = \frac{1-\sqrt{5}}{2}$. By a simple operation, from (18) and (20), we acquire

$$\lim_{n\to\infty}\frac{|e_{n+1}|}{|e_n|^3} = 4d_2^2 + (\sqrt{5}-3)d_3 = \frac{6f''^2(\alpha) + (\sqrt{5}-3)f'(\alpha)f'''(\alpha)}{6f'^2(\alpha)}, \tag{21}$$

which shows that the algorithm (16) is at least a third-order convergent method and this ends the proof. $\square$

Theorem 2. Under the assumptions of Theorem 1, the convergence of the algorithm (15) is at least of order 4.

Proof. Consider the above expansions of $f(x)$, $f'(x)$, $f''(x)$, and $f'''(x)$. Proceeding as before, we obtain

$$\lim_{n\to\infty}\frac{|e_{n+1}|}{|e_n|^4} = d_2^3 + d_4 = \frac{3f''^3(\alpha) + f'^2(\alpha)f^{(4)}(\alpha)}{24f'^3(\alpha)}, \tag{22}$$

which shows that the algorithm (15) is at least a fourth-order convergent method and thus the theorem is proved. $\square$

4 Numerical implementations

The test problems given in this section demonstrate the effectiveness of the developed algorithms to solve the NEs. Meanwhile, the results obtained using the algorithms proposed in this work, (15) and (16) are compared to those obtained using the HIM and a fourth-order method presented in [7]. All calculations have been carried out by using the symbolic calculus software Maple 17 to 2500 significant decimal digits.

Example 1. Solve the following NE:

$$x - \cos(x) = 0, \qquad x_0 = 2. \tag{23}$$

Example 2. Solve the following NE:

$$x - 2 - e^{-x} = 0, \qquad x_0 = 2. \tag{24}$$

Example 3. Solve the following NE:

$$\sin^2(x) - x^2 + 1 = 0, \qquad x_0 = -2. \tag{25}$$

Example 4. Solve the following NE:

$$x^2 - (1 - x)^5 = 0, \qquad x_0 = 1. \tag{26}$$

The numerical results of Examples 1–4 can be observed in Table 1. In Table 1, we list the value of absolute error $f(x_n)$ (labeled as $|f(x_n)|$) when $n = 1, 2, 3, 4, 5$ for the above examples.

From the numerical results of these cases analyzed here, it is easy to conclude that the proposed algorithms are effective and accurate for solving NEs. Moreover, unlike HIM, the developed approaches require just first-order derivative per step.

5 Conclusions

The presented algorithms in this article give rapid convergent approximations. They have the advantage of giving an accurate solution with fewer iterations than the other existing methods.

Table 1: The numerical results obtained from solving the above investigated examples using the algorithms (15) and (16), HIM, and the method of [7] for few iterations.

Algorithm	$\|f(x_1)\|$	$\|f(x_2)\|$	$\|f(x_3)\|$	$\|f(x_4)\|$	$\|f(x_5)\|$
			Example 1		
HIM	7.6_{-3}	7.7_{-6}	7.8_{-12}	8.0_{-24}	8.5_{-48}
Algorithm (16)	1.1_{-1}	5.0_{-5}	5.6_{-15}	7.7_{-45}	2.0_{-134}
Method of [7]	9.9_{-4}	1.2_{-14}	2.4_{-58}	4.1_{-233}	3.3_{-932}
Algorithm (15)	1.2_{-4}	5.1_{-19}	1.6_{-76}	1.4_{-306}	9.4_{-1227}
			Example 2		
HIM	9.2_{-4}	4.1_{-8}	8.1_{-17}	3.1_{-34}	4.5_{-69}
Algorithm (16)	1.3_{-6}	1.9_{-21}	5.8_{-66}	1.7_{-199}	4.2_{-600}
Method of [7]	7.9_{-8}	9.3_{-33}	1.9_{-132}	2.9_{-531}	1.6_{-2126}
Algorithm (15)	4.1_{-8}	3.2_{-34}	1.1_{-138}	1.5_{-556}	5.5_{-2228}
			Example 3		
HIM	3.8_{-1}	3.2_{-2}	3.0_{-4}	2.9_{-8}	2.7_{-16}
Algorithm (16)	1.6_{-1}	5.6_{-4}	3.3_{-11}	7.0_{-33}	6.7_{-98}
Method of [7]	6.9_{-2}	2.2_{-6}	2.8_{-24}	7.3_{-96}	3.5_{-382}
Algorithm (15)	3.6_{-2}	4.8_{-8}	1.7_{-31}	2.6_{-125}	1.3_{-500}
			Example 4		
HIM	2.2_{-1}	2.1_{-2}	3.0_{-4}	6.1_{-8}	2.6_{-15}
Algorithm (16)	5.1_{-2}	6.0_{-5}	1.3_{-13}	1.1_{-39}	8.7_{-118}
Method of [7]	1.6_{-2}	4.4_{-8}	2.4_{-30}	2.0_{-119}	1.0_{-475}
Algorithm (15)	2.9_{-4}	2.3_{-15}	9.9_{-60}	3.3_{-237}	4.1_{-947}

Appendix

Maple code of the algorithm (15) for finding a simple zero of $f(x) :=$ `fun` with the initial guess $x_0 :=$ `x0` $= (a, b)$ up to `dgt` digits, where `ftol` determines the tolerance.

```
fRoots := proc(fun,x0,ftol,dgt)
local m, n, p, r, i, j, k1, k2, f, Df, df, idx, num, fnrm, it;
if dgt<=abs(log10(ftol)) then
    error " Number of digits must be greater than log of ftol"
end if;
idx := [`$`(x0[1]+ii/2,ii = 0 .. 2*(x0[2]-x0[1]))];
f := unapply(fun,x);
Df := diff(f(x),x);
df := unapply(Df,x);
num := 2*(x0[2]-x0[1]);
m := 0;
for i to num-1 do
    if signum(f(idx[i])) <> signum(f(idx[i+1])) then
        m := m+1;
        p[m] := [idx[i], idx[i+1]];
```

```
end if
end do;
if not type(p,table) then error " f(x) contains no sign changes " end if;
for j from 1 to m do
    r[0] := 1/2*(p[j][1]+p[j][2]);
    fnrm : =1: it := 0:
    for n from 0 while (fnrm > ftol) do
        k1 := evalf(f(r[n])^2/(df(r[n])*(f(r[n])-f(r[n]-f(r[n])/df(r[n])))),dgt);
        k2 := evalf(k1*f(r[n]-k1)/f(r[n]),dgt);
        r[n+1] := evalf(r[n]-k1-k2,dgt);
        fnrm := abs(evalf((f(r[n+1])),dgt));
        it := it+1;
    end do;
printf(" x[0] = %g \n Itration = %d \n |f(x)| = %e
 x = %a \n\n",r[0],it,fnrm,r[it]);
end do;
end proc:
```

For example, one can use the above code to solve the NE `ln(x+1)+x-1` (with root $x = 0.557145598997611416858672000001$) by the following:

```
fRoots(ln(x+1)+x-1,[0,10],10^(-10),30);
 x[0] = 0.75
 Itration = 2
 |f(x)| = 6.839800e-26
x = .557145598997611416858671958351
```

References

1. Babolian, E. and Biazar, J. *Solution of nonlinear equations by modified Adomian decomposition method*, Appl. Math. Comput. 132 (2002), 167–172.

2. Basto, M., Semiao, V. and Calheiros, F.L. *A new iterative method to compute nonlinear equations*, Appl. Math. Comput. 173 (2006), 468–483.

3. Chun, C. *Iterative methods improving Newton's method by the decomposition method*, Comput. Math. Appl. 50 (2005), 1559–1568.

4. Golbabai, A. and Javidi, M. *A new family of iterative methods for solving system of nonlinear algebraic equations*, Appl. Math. Comput. 190 (2007), 1717–1722.

5. He, J.H. *A new iterative method for solving algebraic equations*, Appl. Math. Comput. 135 (2003), 81–84.

6. Householder, A.S. *The numerical treatment of a single nonlinear equation*, McGraw-Hill, New York, 1970.

7. Maheshwari, A.K. *A fourth-order iterative method for solving nonlinear equations*, Appl. Math. Comput. 211 (2009), 383–391.

8. Noor, M.A. *New iterative schemes for nonlinear equations*, Appl. Math. Comput. 187 (2007), 937–943.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 117–133
DOI:10.22067/ijnao.2020.11305.0

Research Article

Computation of eigenvalues of fractional Sturm–Liouville problems

E.M. Maralani, F.D. Saei*, A.A.J. Akbarfam and K. Ghanbari

Abstract

We consider the eigenvalues of the fractional-order Sturm–Liouville equation of the form

$$-{}^{c}D^{\alpha}_{0+} \circ D^{\alpha}_{0+} y(t) + q(t)y(t) = \lambda y(t), \quad 0 < \alpha \leq 1, \quad t \in [0, 1],$$

with Dirichlet boundary conditions

$$I^{1-\alpha}_{0+} y(t)|_{t=0} = 0 \quad \text{and} \quad I^{1-\alpha}_{0+} y(t)|_{t=1} = 0,$$

where $q \in L^2(0, 1)$ is a real-valued potential function. The method is used based on Picard's iteration procedure. We show that the eigenvalues are obtained from the zeros of the Mittag-Leffler function and its derivatives.

AMS subject classifications (2020): 26A33, 34A08, 34B09.

Keywords: Fractional Sturm–Liouville; Fractional calculus; Iterative methods; Eigenvalues.

1 Introduction

Fractional Sturm–Liouville Problems (FSLPs) are generalizations of the classical Sturm–Liouville Problems in which the ordinary derivatives are replaced

*Corresponding author
Received 12 January 2020; revised 2 September 2020; accepted 29 December 2020
Elnaz Massah Maralani
Department of Mathematics, Tabriz Branch, Islamic Azad University, Tabriz, Iran. e-mail: stu.massah.elnaz@iaut.ac.ir

Farhad Dastmalchi Saei
Department of Mathematics, Tabriz Branch, Islamic Azad University, Tabriz, Iran. e-mail: dastmalchi@iaut.ac.ir

Ali Asghar Jodayree Akbarfam
University of Tabriz, Tabriz, Iran. e-mail: akbarfam@tabrizu.ac.ir

Kazem Ghanbari
Sahand university of Technology, Tabriz, Iran. e-mail: kghanbari@sut.ac.ir

by fractional derivatives or derivatives of fractional order. As an introduction, the interested reader may wish to consult the great variety of works on the subject, starting with books such as [14, 20, 23]. Several authors have considered the numerical FSLP, for example Al-Mdallal [2, 24] applied the Adomian decomposition method for solving fractional Sturm–Liouville problems. Abbasbandy and Shirzadi [1] applied the homotopy analysis method for solving fractional Sturm–Liouville problems. Also in [8], the eigenvalue problems for the fractional ordinary differential equations have been investigated with different classes of boundary conditions including the Dirichlet, Neumann, and so on. They explained that choosing $\alpha = 2$ leads to the classical ones of the second-order differential equations. When the order α satisfies $1 < \alpha < 2$, the eigenvalues can be finitely many; see [9, 22]. It has been applied to many fields in science and engineering, such as viscoelasticity, anomalous diffusion, fluid mechanics, biology, chemistry, acoustics, control theory, and so on. In the applications mentioned above, a class of integro-differential equations with singularities, fractional differential equations have been involved; see [11, 16, 15, 17, 21, 18, 20, 5]. In [25], the authors have considered a regular fractional Sturm–Liouville problem (RFSLP) of two kinds RFSLP-I and RFSLP-II of order $\nu \in (0, 2)$ with the fractional differential operators both of Riemann–Liouville and Caputo type, of the same fractional-order $\mu = \nu/2 \in (0, 1)$. It was proved that the regular boundary-value problems RFSLP-I & -II are indeed asymptotic cases for the singular counterparts SFSLP-I & -II. The inverse Laplace transform method for obtaining analytical solutions of the FSLPs and eigenvalues has been investigated in [10]. The reproducing kernel method has also been used to calculate the eigenvalues of the FSLPs. Dehgan and Mingarelli [6, 7] have investigated the general solution of three- or two-term fractional differential equations of mixed Caputo/Riemann–Liouville type in the case of Dirichlet boundary conditions. From a numerical viewpoint, we also refer the reader for fractional differential equations to [2, 3, 4, 12, 24]. Particularly, the boundary value problem is of the form

$$-\,{}^{c}D^{\alpha}_{0^+} \circ D^{\alpha}_{0^+} y(t) + q(t)y(t) = \lambda y(t), \quad 0 < \alpha \le 1, \quad t \in [0,1], \tag{1}$$

$$I^{1-\alpha}_{0^+} y(t)|_{t=0} = 0, \quad \text{and} \quad I^{1-\alpha}_{0^+} y(t)|_{t=1} = 0.$$

For each $1/2 < \alpha < 1$, it is proved that there is a finite number of real eigenvalues, an infinite number of nonreal eigenvalues, that the number of such real eigenvalues grow without bound as $\alpha \to 1^-$. Also, they expressed asymptotically behavior of the eigenvalues as a function of α in the following form:

$$\lambda_n(\alpha) \sim \Big(\frac{n\pi}{\sin(\frac{\pi}{2\alpha})}\Big)^{2\alpha}, \qquad \alpha \to 1^-.$$

This corresponds exactly with the well-known classical asymptotic estimate $\lambda_n \to n^2\pi^2$ as $n \to \infty$; see [6, 7]. The remaining structure of this paper is organized as follows: In the next section, we review the Mittag-Leffler function and Laplace transform of the Riemann–Liouville and the Caputo fractional derivatives. In section 3, we discuss the zeros of the iterative method with two examples. In section 4, we discuss the analysis of the iterative method. The last section includes our conclusions.

2 Preliminaries

We recall some definitions in fractional calculus. We refer the reader to [6] for further details.

Definition 1. The left and the right Riemann–Liouville fractional integrals $I^{\alpha}_{a^+}$ and $I^{\alpha}_{b^-}$ of order $\alpha \in \mathbb{R}^+$ are defined by

$$I^{\alpha}_{a^+} f(t) := \frac{1}{\Gamma(\alpha)} \int_a^t (t-s)^{\alpha-1} f(s) ds, \quad (a, b], \tag{2}$$

and

$$I^{\alpha}_{b^-} f(t) := \frac{1}{\Gamma(\alpha)} \int_t^b (s-t)^{\alpha-1} f(s) ds, \quad [a, b),$$

respectively.

Here $\Gamma(\alpha)$ denotes Euler's gamma function. The following property is easily verified.

Lemma 1. For a constant C, we have $I^{\alpha}_{a^+} C = \frac{(t-a)^{\alpha}}{\Gamma(\alpha+1)} \cdot C$.

Definition 2. The left and the right Caputo fractional derivatives ${}^cD^{\alpha}_{a^+}$ and ${}^cD^{\alpha}_{b^-}$ are defined by

$${}^cD^{\alpha}_{a^+} f(t) := I^{n-\alpha}_{a^+} \circ D^n f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t (t-s)^{n-\alpha-1} f^{(n)}(s) ds, \quad t > a, \tag{3}$$

and for $t < b$,

$${}^cD^{\alpha}_{b^-} f(t) := (-1)^n I^{n-\alpha}_{b^-} \circ D^n f(t) = \frac{(-1)^n}{\Gamma(n-\alpha)} \int_t^b (s-t)^{\alpha-n+1} f^{(n)}(s) ds, \tag{4}$$

respectively, where f is sufficiently differentiable and $n-1 \leq \alpha < n$.

Definition 3. The left and the right Riemann–Liouville fractional derivatives $D^{\alpha}_{a^+}$ and $D^{\alpha}_{b^-}$ are defined by

$$D^{\alpha}_{a^+} f(t) := D^n \circ I^{n-\alpha}_{a^+} f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t (t-s)^{n-\alpha-1} f(s) ds, \quad t > a, \tag{5}$$

and for $t < b$,

$$D_{b^-}^{\alpha} f(t) := (-1)^n D^n \circ I_{b^-}^{n-\alpha} f(t) = \frac{(-1)^n}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_t^b (s-t)^{n-\alpha-1} f(s) ds, \quad (6)$$

respectively, where f is sufficiently differentiable and $n-1 \leq \alpha < n$.

2.1 The Mittag-Leffler function

The function $E_\alpha(z)$ is defined by

$$E_\alpha(z) := \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)} \quad (z \in \mathbb{C},\ R(\alpha) > 0), \quad (7)$$

which was introduced by Mittag-Leffler [20]. In particular, when $\alpha = 1$ and $\alpha = 2$, we have

$$E_1(z) = e^z, \quad E_2(z) = \cosh(\sqrt{z}).$$

The generalized Mittag-Leffler function $E_{\alpha,\beta}(z)$ is defined by

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)} \quad (z, \theta \in \mathbb{C},\ R(\alpha) > 0), \quad (8)$$

of course, when $\beta = 1$, $E_{\alpha,\beta}(z)$ coincides with the Mittag-Leffler function (18):

$$E_{\alpha,1}(z) = E_\alpha(z).$$

Two other particular cases of (3) are as follows:

$$E_{1,2}(z) = \frac{e^z - 1}{z}, \quad E_{2,2}(z) = \frac{\sinh(\sqrt{z})}{\sqrt{z}}.$$

Further properties of this special function may be found in [11].

Theorem 1 (see [22])**.** If $\alpha < 2$, β is an arbitrary real number, μ is such that
$\frac{\pi\alpha}{2} < \mu < \min\{\pi, \pi\alpha\}$, and c is a real constant, then

$$|E_{\alpha,\beta}(z)| \leq \frac{c}{1 + |z|} \quad (\mu \leq |arg(z)| \leq \pi), \quad |z| \geq 0, \ \ z \in \mathbb{C}.$$

2.2 Laplace transform

Definition 4 (see [22, 19])**.** The Laplace transform of a function $f(t)$ defined for all real numbers $t \geq 0$, t stands for the time, is the function $F(s)$ that is a unilateral transform defined by

$$F(s) = \mathfrak{L}\{f(t)\} := \int_0^\infty e^{-st} f(t)dt,$$

where s is the frequency parameter.

Definition 5 (see [22, 19])**.** The convolution of $f(t)$ and $g(t)$ supported on only $[0, \infty)$ is defined by

$$(f * g)(t) = \int_0^t f(s)g(t-s)ds, \qquad f, g : [0, \infty) \to \mathbb{R}.$$

Property 1 (see [22])**.** The Laplace transform of the convolution of $f(t)$ and $g(t)$ is given by following relation:

$$\mathfrak{L}\{(f * g)(t)\} = \mathfrak{L}\{f(t)\} \times \mathfrak{L}\{g(t)\}.$$

Property 2 (see [6, 22])**.** The Laplace transform of the derivatives of the Mittag-Leffler function reads as follows:

$$\int_0^\infty e^{-st} t^{\alpha k + \beta - 1} E^{(k)}_{\alpha,\beta}(\pm\lambda t^\alpha)dt = \frac{k! s^{\alpha-\beta}}{(s^\alpha \mp \lambda)^{k+1}}, \qquad (Re(p) > |a|^{\frac{1}{\alpha}}).$$

Property 3 (see [13, 22])**.** The Laplace transform of the Riemann–Liouville fractional derivative is obtained as

$$(\mathfrak{L}_0 D_t^\alpha y)(s) = s^\alpha (\mathfrak{L}y)(s) - \sum_{k=0}^{n-1} s^{n-k-1} D^k ({}_0I_t^{n-\alpha} y)(0), \; n-1 < \alpha \leq n, n \in \mathbb{N}.$$

If $0 < \alpha \leq 1$, then $(\mathfrak{L}_0 D_t^\alpha y)(s) = s^\alpha(\mathfrak{L}y)(s) - ({}_0I_t^{n-\alpha} y)(0)$.

Property 4 (see [13, 22])**.** The Laplace transform of the Caputo fractional derivative is obtained as

$$(\mathfrak{L}_0^c D_t^\alpha y)(s) = s^\alpha (\mathfrak{L}y)(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} (D^k y)(0), \quad n-1 < \alpha \leq n, n \in \mathbb{N}.$$

If $0 < \alpha \leq 1$, then $(\mathfrak{L}_0^c D_t^\alpha y)(s) = s^\alpha(\mathfrak{L}y)(s) - s^{\alpha-1} y(0)$.

3 Iterative method

If we take the Laplace transformation of equation (1), then we get the following equation:

$$\mathfrak{L}\{y(t)\} = \frac{s^{\alpha}}{\lambda + s^{2\alpha}} I_{0^+}^{1-\alpha} y(t)|_{t=0} + \frac{s^{\alpha-1}}{\lambda + s^{2\alpha}} D_{0^+}^{\alpha} y(t)|_{t=0} + \frac{1}{\lambda + s^{2\alpha}} \mathfrak{L}\{h(t)\},$$

where $h(t) := q(t)y(t)$. We use the inverse Laplace transform

$$\begin{aligned} y(t) =& c_1 t^{\alpha-1} E_{2\alpha,2}(-\lambda t^{2\alpha}) + c_2 t^{\alpha} E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) \\ &+ t^{2\alpha-1} E_{2\alpha,2\alpha}(-\lambda t^{2\alpha}) * h(t), \end{aligned} \tag{9}$$

in which, specifying the constants in (2) by setting, without loss of generality, $c_1 = 0$ and $c_2 = 1$ are given constants and $*$ is the convolution symbol. We consider the following iterative method:

$$y_m(t) = t^{\alpha} E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) + \int_0^t (t-s)^{2\alpha-1} E_{2\alpha,2\alpha}(-\lambda(t-s)^{2\alpha}) q(s) y_{m-1}(s) ds.$$

Example 1. For simplicity, first we assume that $q(s) = 1$ and that

$$y_0(t) = t^{\alpha} E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}).$$

Then we get

$$\begin{aligned} y_1(t) =& t^{\alpha} E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) \\ &+ \int_0^t (t-s)^{2\alpha-1} E_{2\alpha,2\alpha}(-\lambda(t-s)^{2\alpha}) s^{\alpha} E_{2\alpha,\alpha+1}(-\lambda s^{2\alpha}) ds. \end{aligned}$$

To calculate the integral term, we apply the following method:

$$\begin{aligned} &\mathfrak{L}^{-1}(\mathfrak{L}\{\int_0^t (t-s)^{2\alpha-1} E_{2\alpha,2\alpha}(-\lambda(t-s)^{2\alpha}) s^{\alpha} E_{2\alpha,\alpha+1}(-\lambda s^{2\alpha}) ds\}) \\ &= \frac{1}{1!} t^{3\alpha} E_{2\alpha,\alpha+1}^{(1)}(-\lambda t^{2\alpha}). \end{aligned}$$

Thus

$$y_1(t) = t^{\alpha} E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) + \frac{1}{1!} t^{3\alpha} E_{2\alpha,\alpha+1}^{(1)}(-\lambda t^{2\alpha}).$$

Now, we calculate $y_2(t)$ as follows:

$$y_2(t) = y_1(t) + \frac{1}{2!} t^{5\alpha} E_{2\alpha,\alpha+1}^{(2)}(-\lambda t^{2\alpha}),$$

as the same way, we get

$$y_3(t) = y_2(t) + \frac{1}{3!}t^{7\alpha}E^{(3)}_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}).$$

Finally, after the iteration, we have the following relation:

$$y_m(t) = \sum_{k=0}^{m+1}\frac{1}{k!}t^{(2k+1)\alpha}E^{(k)}_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}). \tag{10}$$

By choosing some terms from the approximate solution and calculating their zeros, the same results are obtained from [7] as follows:

$$\begin{aligned} y_3(t) =& t^{\alpha}E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) + \frac{1}{1!}t^{3\alpha}E^{(1)}_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) \\ &+ \frac{1}{2!}t^{5\alpha}E^{(2)}_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) + \frac{1}{3!}t^{7\alpha}E^{(3)}_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}). \end{aligned}$$

For example, if we take three terms with the boundary conditions

$$I^{1-\alpha}_{0^+}y(t)|_{t=1} = \frac{1}{\Gamma(1-\alpha)}\int_0^t (t-s)^{-\alpha}y(s)ds = 0,$$

then we get

$$\begin{aligned} I^{1-\alpha}_{0^+}y_3(t)|_{t=1} =& \mathfrak{L}^{-1}(\mathfrak{L}\{\frac{1}{\Gamma(1-\alpha)}\int_0^t (t-s)^{-\alpha}s^{\alpha}E_{2\alpha,\alpha+1}(-\lambda s^{2\alpha})ds \\ &+ \frac{1}{\Gamma(1-\alpha)}\int_0^t (t-s)^{-\alpha}\frac{1}{1!}s^{3\alpha}E^{(1)}_{2\alpha,\alpha+1}(-\lambda s^{2\alpha})ds \\ &+ \frac{1}{\Gamma(1-\alpha)}\int_0^t (t-s)^{-\alpha}\frac{1}{2!}s^{5\alpha}E^{(2)}_{2\alpha,\alpha+1}(-\lambda s^{2\alpha})ds \\ &+ \frac{1}{\Gamma(1-\alpha)}\int_0^t (t-s)^{-\alpha}\frac{1}{3!}s^{7\alpha}E^{(3)}_{2\alpha,\alpha+1}(-\lambda s^{2\alpha})ds\})|_{t=1} = 0. \end{aligned}$$

Using the inverse Laplace transform implies

$$\begin{aligned} I^{1-\alpha}_{0^+}y_3(t)|_{t=1} &= \mathfrak{L}^{-1}\{\frac{s^{2\alpha-2}}{s^{2\alpha}+\lambda}\} + \mathfrak{L}^{-1}\{\frac{s^{2\alpha-2}}{(s^{2\alpha}+\lambda)^{1+1}}\} \\ &\quad + \mathfrak{L}^{-1}\{\frac{s^{2\alpha-2}}{(s^{2\alpha}+\lambda)^{2+1}}\} + \mathfrak{L}^{-1}\{\frac{s^{2\alpha-2}}{(s^{2\alpha}+\lambda)^{3+1}}\} = 0 \\ &= \Big[tE_{2\alpha,2}(-\lambda t^{2\alpha}) + t^{2\alpha+1}E^{(1)}_{2\alpha,2}(-\lambda t^{2\alpha}) + t^{4\alpha+1}E^{(2)}_{2\alpha,2}(-\lambda t^{2\alpha}) \\ &\quad + t^{6\alpha+1}E^{(3)}_{2\alpha,2}(-\lambda t^{2\alpha})\Big]_{t=1} = 0. \end{aligned}$$

Therefore, the zeros of the above relation yield eigenvalues.

For the case of $q(s) = 0$, we refer the reader to [6, 7].

Table 1: The eigenvalues λ_n of the $FSLP$ of Example 1

k	α	0.88	0.92	0.96	0.98	0.99	1
10	λ_n	10.71568699	10.55992405	10.66410923	10.78678886	10.86333576	10.94942560
		27.42039814	31.48842150	35.71275308	38.01570185	39.23145940	40.49542871
		39.66187430	51.88983922	67.66694524	77.14409122	82.33647923	87.85647119
20	λ_n	10.71568686	10.55992405	10.66410923	10.78678886	10.86333576	10.94942560
		27.51414362	31.51192899	35.71778637	38.01801903	39.23303636	40.49650499
		60.83896648	66.90351096	76.82733019	82.94041507	86.28709241	89.83417215
		86.83800089	108.6657324	131.4598304	144.4575545	151.4897515	158.9183593
		107.4785323	147.2884927	199.9559706	231.8456925	238.3472917	249.1565859
						260.3905771	286.9178097
						275.8685836	300.6931093

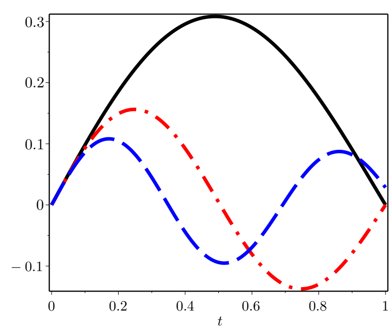


Figure 1: The curves of eigenfunctions, $k = 10$ for $n = 1$ (solid line), $n = 2$ (dash dot line), $n = 3$ (dash line), where $\alpha = 0.98$, $\lambda_1 = 10.78678886$, $\lambda_2 = 38.01570185$, and $\lambda_3 = 77.14409122$ for Example 1.

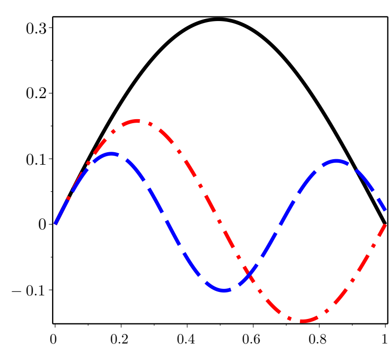


Figure 2: The curves of eigenfunctions, $k = 10$ for $n = 1$ (solid line), $n = 2$ (dash dot line), $n = 3$ (dash line), where $\alpha = 0.99$, $\lambda_1 = 10.86333576$, $\lambda_2 = 39.23145940$, and $\lambda_3 = 82.33647923$ for Example 1.

Remark 1. Note that as α approaches 1, one can see that the eigenvalues satisfy $\lambda_n = n^2\pi^2 + 1$. This shows that our results are a generalization of the classical ones.

Example 2. Now, we assume $q(s) = s^\beta$, and we have the following iterative method, analogous to the previous computation:

$$y_m(t) = t^\alpha E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) + \int_0^t (t-s)^{2\alpha-1} E_{2\alpha,2\alpha}(-\lambda(t-s)^{2\alpha}) q(s) y_{m-1}(s) ds.$$

We define

$$\begin{aligned} y_1(t) =& t^\alpha E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha}) \\ &+ \int_0^t (t-s)^{2\alpha-1} E_{2\alpha,2\alpha}(-\lambda(t-s)^{2\alpha}) \cdot s^{\alpha+\beta} E_{2\alpha,\alpha+1}(-\lambda s^{2\alpha}) ds, \end{aligned}$$

and again to calculate the integral term, we apply the following method:

$$\begin{aligned} &\mathfrak{L}^{-1}\Big(\mathfrak{L}\{t^{2\alpha-1}E_{2\alpha,2\alpha}(-\lambda t^{2\alpha})\} \cdot \mathfrak{L}\{t^{\alpha+\beta}E_{2\alpha,\alpha+1}(-\lambda t^{2\alpha})\}\Big) \\ &= \sum_{k_1=0}^{\infty} (-\lambda)^{k_1} \cdot \frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)} \cdot t^{(2k_1+3)\alpha+\beta} E_{2\alpha,(2k_1+3)\alpha+\beta+1}(-\lambda t^{2\alpha}). \end{aligned}$$

Thus

$$\begin{aligned} y_1(t) = y_0(t) + &\sum_{k_1=0}^{\infty} (-\lambda)^{k_1} \cdot \frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)} \\ &\times t^{(2k_1+3)\alpha+\beta} E_{2\alpha,(2k_1+3)\alpha+\beta+1}(-\lambda t^{2\alpha}), \end{aligned}$$

now, we calculate $y_2(t)$ as follows:

$$\begin{aligned} y_2(t) = y_1(t) + &\sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} (-\lambda)^{k_1+k_2} \\ &\times \frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)} \cdot \frac{\Gamma((2k_2+2k_1+3)\alpha+2\beta+1)}{\Gamma((2k_2+2k_1+3)\alpha+\beta+1)} \\ &\times t^{(2k_2+2k_1+5)\alpha+2\beta} E_{2\alpha,(2k_2+2k_1+5)\alpha+2\beta+1}(-\lambda t^{2\alpha}), \end{aligned} \tag{11}$$

similarly, we have

$$
\begin{aligned}
y_3(t) =& y_2(t)\\
&+\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}(-\lambda)^{k_1+k_2+k_3}\cdot\frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)}\\
&\times\frac{\Gamma((2k_2+2k_1+3)\alpha+2\beta+1)}{\Gamma((2k_2+2k_1+3)\alpha+\beta+1)}\times\frac{\Gamma((2k_3+2k_2+2k_1+5)\alpha+3\beta+1)}{\Gamma((2k_3+2k_2+2k_1+5)\alpha+2\beta+1)}\\
&\times t^{(2k_3+2k_2+2k_1+7)\alpha+3\beta}E_{2\alpha,(2k_3+2k_2+2k_1+7)\alpha+3\beta+1}(-\lambda t^{2\alpha}).
\end{aligned}
$$

Finally, we get the following relation for $y_m(t)$:

$$
\begin{aligned}
y_m(t) =& y_{m-1}(t)+\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\cdots\sum_{k_m=0}^{\infty}(-\lambda)^{k_1+k_2+\cdots+k_m}\cdot\frac{A}{B}\\
&\times\cdot t^{(2k_m+2k_{m-1}+\cdots+2k_1+(2m-1))\alpha+m\beta}\\
&\times E_{2\alpha,(2k_m+2k_{m-1}+\cdots+2k_1+(2m-1))\alpha+m\beta+1)}(-\lambda t^{2\alpha}),
\end{aligned}
\tag{12}
$$

where

$$
\begin{aligned}
A =& \Gamma((2k_1+1)\alpha+\beta+1)\Gamma((2k_2+2k_1+3)\alpha+2\beta+1)\times\cdots\\
&\times\Gamma((2k_m+2k_{m-1}+\cdots+2k_1+(2m-1))\alpha+m\beta+1),
\end{aligned}
$$

and

$$
\begin{aligned}
B =& \Gamma((2k_1+1)\alpha+1)\Gamma((2k_2+2k_1+3)\alpha+\beta+1)\times\cdots\\
&\times\Gamma((2k_m+2k_{m-1}+\cdots+2k_1+(2m-1))\alpha+(m-1)\beta+1).
\end{aligned}
$$

Now, in order to obtain eigenvalues, by choosing three terms from (37) and imposing the following boundary condition, we have

$$
\begin{aligned}
&I_{0^+}^{1-\alpha}y_3(t)|_{t=1}\\
&=\mathfrak{L}^{-1}\Bigg(\mathfrak{L}\Big\{\frac{1}{\Gamma(1-\alpha)}\int_0^t(t-s)^{-\alpha}s^{\alpha}E_{2\alpha,\alpha+1}(-\lambda s^{2\alpha})ds\\
&\quad+\sum_{k_1=0}^{\infty}(-\lambda)^{k_1}\frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)}\\
&\quad\times\frac{1}{\Gamma(1-\alpha)}\int_0^t(t-s)^{-\alpha}s^{(2k_1+3)\alpha+\beta}E_{2\alpha,(2k_1+3)\alpha+\beta+1}(-\lambda s^{2\alpha})ds\\
&\quad+\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}(-\lambda)^{k_1+k_2}\frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)}\cdot\frac{\Gamma((2k_2+2k_1+3)\alpha+2\beta+1)}{\Gamma((2k_2+2k_1+3)\alpha+\beta+1)}\\
&\quad\times\frac{1}{\Gamma(1-\alpha)}\int_0^t(t-s)^{-\alpha}s^{(2k_2+2k_1+5)\alpha+2\beta}E_{2\alpha,(2k_2+2k_1+5)\alpha+2\beta+1}(-\lambda s^{2\alpha})ds
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}(-\lambda)^{k_1+k_2+k_3}\frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)}\\
&\times\frac{\Gamma((2k_2+2k_1+3)\alpha+2\beta+1)}{\Gamma((2k_2+2k_1+3)\alpha+\beta+1)}\cdot\frac{\Gamma((2k_3+2k_2+2k_1+5)\alpha+3\beta+1)}{\Gamma((2k_3+2k_2+2k_1+5)\alpha+2\beta+1)}\\
&\times\frac{1}{\Gamma(1-\alpha)}\int_0^t(t-s)^{-\alpha}s^{(2k_3+2k_2+2k_1+7)\alpha+3\beta}\\
&\qquad\qquad\times E_{2\alpha,(2k_3+2k_2+2k_1+7)\alpha+3\beta+1}(-\lambda s^{2\alpha})ds\Big|_{t=1}\Bigg\}\Bigg)=0. \qquad (13)
\end{aligned}
$$

Now, by using the inverse Laplace transform, we have

$$
\begin{aligned}
&I_{0^+}^{1-\alpha}y_3(t)|_{t=1}\\
&\quad=tE_{2\alpha,2}(-\lambda t^{2\alpha})+\sum_{k_1=0}^{\infty}(-\lambda)^{k_1}\cdot\frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)}\\
&\quad\times t^{(2k_1+2)\alpha+\beta+1}E_{2\alpha,(2k_1+2)\alpha+\beta+2}(-\lambda t^{2\alpha})\\
&\quad+\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}(-\lambda)^{k_1+k_2}\frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)}\cdot\frac{\Gamma((2k_2+2k_1+3)\alpha+2\beta+1)}{\Gamma((2k_2+2k_1+3)\alpha+\beta+1)}\\
&\quad\times t^{(2k_2+2k_1+4)\alpha+2\beta+1}E_{2\alpha,(2k_2+2k_1+4)\alpha+2\beta+2}(-\lambda t^{2\alpha})\\
&\quad+\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}(-\lambda)^{k_1+k_2+k_3}\frac{\Gamma((2k_1+1)\alpha+\beta+1)}{\Gamma((2k_1+1)\alpha+1)}\\
&\quad\times\frac{\Gamma((2k_2+2k_1+3)\alpha+2\beta+1)}{\Gamma((2k_2+2k_1+3)\alpha+\beta+1)}\cdot\frac{\Gamma((2k_3+2k_2+2k_1+5)\alpha+3\beta+1)}{\Gamma((2k_3+2k_2+2k_1+5)\alpha+2\beta+1)}\\
&\quad\times t^{(2k_3+2k_2+2k_1+6)\alpha+3\beta+1}E_{2\alpha,(2k_3+2k_2+2k_1+6)\alpha+3\beta+2}(-\lambda t^{2\alpha})|_{t=1}=0.
\end{aligned}
$$

Therefore, solving the above relation with respect to λ yields eigenvalues.

Table 2: The eigenvalues λ_n of the $FSLP$ for Example 2

k_i	α	0.88	0.92	0.96	0.98	0.99	1.0
$k_1,k_2,k_3=5$	λ_n	10.05385926	9.933778015	10.06348862	10.19660718	10.27791426	10.36849015
		47.56007387	32.34315896	35.63420589	37.76603821	38.92583141	40.14691848
		1126.335623	44.07433976	60.04931462	68.65397481	73.15018892	77.78582381
		2383.852442	60.98311732	79.07639037	90.47082662	96.86784919	103.7750278
		4084.543198	1672.025020	2492.533591	3047.891192	3371.626995	3730.657415
		29.99913622	3669.500947	5672.277915	7063.067903	7884.473754	8803.558665
			6468.168446	10283.17818	12984.46721	14595.72336	16410.73873
			44.95131154	62.05287730	72.95773310	79.12198365	85.81612846
$k_1,k_2,k_3=10$	λ_n	10.05563487	9.934078332	10.06355202	10.19663792	10.27793590	10.36850547
		26.86376328	30.94976951	35.18620831	37.49382726	38.71166129	39.97755111
		38.80996910	51.04646579	66.84154959	76.33005616	81.52855664	87.05489991

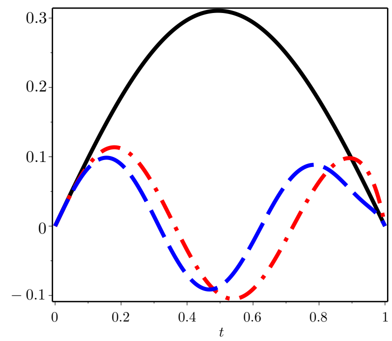


Figure 3: The curves of eigenfunctions, $k_1, k_2, k_3 = 5$ for $n = 1$ (solid line), $n = 2$ (dash dot line), $n = 3$ (dash line), where $\alpha = 0.98$, $\lambda_1 = 10.19660718$, $\lambda_2 = 68.65397481$, and $\lambda_3 = 90.47082662$ for Example 2.

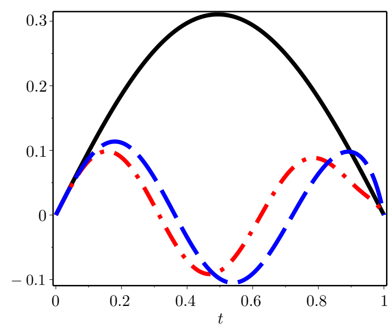


Figure 4: The curves of eigenfunctions, $k_1, k_2, k_3 = 5$ for $n = 1$ (solid line), $n = 2$ (dash dot line), $n = 3$ (dash line), where $\alpha = 0.99$, $\lambda_1 = 10.27791426$, $\lambda_2 = 73.15018892$, and $\lambda_3 = 96.86784919$ for Example 2.

4 Analysis of the iterative method

Lemma 2. The n-fold series in relation (37) is convergent for $\alpha = 1$, $\beta \in \mathbb{N}$, and $|t| < \frac{1}{\sqrt{|\lambda|}}$.

Proof. It is sufficient to show that the double series in relation (11) is convergent. By Theorem 1, we have

$$\begin{aligned} |y_2(t) - y_1(t)| \leq \tfrac{c}{1+|\lambda t^2|} \textstyle\sum_{k_1=0}^{\infty} |\lambda|^{k_1} \tfrac{(2k_1+1+\beta)!}{[(2k_1+1)]!} \\ \times \sum_{k_2=0}^{\infty} |\lambda|^{k_2} \frac{(2k_2+2k_1+3+2\beta)!}{([(2k_2+2k_1+3)]+\beta)!} t^{2k_2+2k_1+5+2\beta}. \end{aligned} \tag{14}$$

Since $\limsup_{k_2\to\infty} \sqrt[k_2]{\frac{(2k_2+2k_1+3+2\beta)!|\lambda|^{k_2}}{([(2k_2+2k_1+3)]+\beta)!}} = |\lambda|$, by the basic root test of convergence of nonnegative series, we conclude that for $|t| < \frac{1}{\sqrt{|\lambda|}}$, the series

$$\sum_{k_2=0}^{\infty} |\lambda|^{k_2} \frac{(2k_2+2k_1+3+2\beta)!}{([2k_2+2k_1+3]+\beta)!} t^{2k_2+2k_1+5+2\beta}$$

is absolutely convergent. Moreover, it is uniformly convergent on any compact subset of the interval $(-\frac{1}{\sqrt{|\lambda|}}, \frac{1}{\sqrt{|\lambda|}})$.

On the other hand, the series $\sum_{k_2=0}^{\infty} |\lambda|^{k_2}.t^{2k_2+2k_1+5+2\beta}$ is absolutely convergent on the interval $(-\frac{1}{\sqrt{|\lambda|}}, \frac{1}{\sqrt{|\lambda|}})$ and is uniformly convergent on any compact subset of the interval $(-\frac{1}{\sqrt{|\lambda|}}, \frac{1}{\sqrt{|\lambda|}})$. So we can take derivative of it on this interval, term by term, for β times and get

$$\begin{aligned}
&\sum_{k_2=0}^{\infty} ([2k_2+2k_1+3]+\beta+1)([2k_2+2k_1+3]+\beta+2) \\
&\quad \ldots([2k_2+2k_1+3]+2\beta)|\lambda|^{k_2} t^{2k_2+2k_1+5+2\beta} \\
&\quad = t^{\beta}\Big(\sum_{k_2=0}^{\infty} |\lambda|^{k_2}.t^{2k_2+2k_1+5+2\beta}\Big)^{(\beta)} \\
&\quad = t^{\beta}\Big(t^{2k_1+5+2\beta} \sum_{k_2=0}^{\infty} (|\lambda| t^2)^{k_2}\Big)^{(\beta)} \\
&\quad = t^{\beta}\Big(\frac{t^{2k_1+5+2\beta}}{1-|\lambda|t^2}\Big)^{(\beta)}.
\end{aligned}$$

Now, using (14), we obtain

$$\begin{aligned}
&|y_2(t)-y_1(t)| \\
&\le \frac{c}{1+|\lambda|t^2} \sum_{k_1=0}^{\infty} |\lambda|^{k_1} \frac{(2k_1+1+\beta)!}{[2k_1+1]!} \cdot t^{\beta}\Big(\frac{t^{2k_1+5+2\beta}}{1-|\lambda|t^2}\Big)^{(\beta)} \\
&= \frac{ct^{\beta}}{1+|\lambda|t^2}\Big(\sum_{k_1=0}^{\infty} |\lambda|^{k_1} \frac{(2k_1+1+\beta)!}{[2k_1+1]!} \cdot \frac{t^{2k_1+5+2\beta}}{1-|\lambda|t^2}\Big)^{(\beta)} \\
&= \frac{ct^{\beta}}{1+|\lambda|t^2}\Big(\frac{t^{2\beta+4}}{1-|\lambda|t^2} \sum_{k_1=0}^{\infty} |\lambda|^{k_1} \frac{(2k_1+1+\beta)!}{[2k_1+1]!} t^{2k_1+1}\Big)^{(\beta)} \\
&= \frac{ct^{\beta}}{1+|\lambda|t^2}\Big(\frac{t^{2\beta+4}}{1-|\lambda|t^2} \sum_{k_1=0}^{\infty} |\lambda|^{k_1} (2k_1+2)\cdots(2k_1+1+\beta) t^{2k_1+1}\Big)^{(\beta)}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{ct^{\beta}}{1+|\lambda|t^2}\left(\frac{t^{2\beta+4}}{1-|\lambda|t^2}\sum_{k_1=0}^{\infty}|\lambda|^{k_1}\big(t^{2k_1+1+\beta}\big)^{(\beta)}\right)^{(\beta)}\\
&= \frac{ct^{\beta}}{1+|\lambda|t^2}\left(\frac{t^{2\beta+4}}{1-|\lambda|t^2}\Big(\sum_{k_1=0}^{\infty}|\lambda|^{k_1}t^{2k_1+1+\beta}\Big)^{(\beta)}\right)^{(\beta)}\\
&= \frac{ct^{\beta}}{1+|\lambda|t^2}\left(\frac{t^{2\beta+4}}{1-|\lambda|t^2}\Big(t^{\beta+1}\sum_{k_1=0}^{\infty}|\lambda|^{k_1}t^{2k_1}\Big)^{(\beta)}\right)^{(\beta)}\\
&= \frac{ct^{\beta}}{1+|\lambda|t^2}\left(\frac{t^{2\beta+4}}{1-|\lambda|t^2}\Big(\frac{t^{\beta+1}}{1-|\lambda|t^2}\Big)^{(\beta)}\right)^{(\beta)}.
\end{aligned}
$$

Therefore, (14) is appeared to be the derivative of a function and hence is convergent.

It follows from the root test that $\frac{1}{R} = \limsup_{k_2\to\infty} \sqrt[k_2]{\frac{(2k_2+2k_1+3+2\beta)!|\lambda|^{k_2}}{([(2k_2+2k_1+3)]+\beta)!}} = |\lambda|$. Hence

$$
\begin{aligned}
C = &\sum_{k_2=0}^{\infty}([(2k_2+2k_1+3)]+\beta+1)([(2k_2+2k_1+3)]+\beta+2)\\
&\times\cdots\times([(2k_2+2k_1+3)]+2\beta)|\lambda|^{k_2}.t^{2k_2+2k_1+5+2\beta}\\
=&\Big(\sum_{k_2=0}^{\infty}(|\lambda|^{k_2}.t^{2k_2+2k_1+5+2\beta}\Big)^{(k)},
\end{aligned}
$$

where $k = (2k_2+2k_1+5) - [(2k_2+2k_1+5)+\beta-2]$. Then

$$
C = \Big(t^{2k_1+5+2\beta}\sum_{k_2=0}^{\infty}(|\lambda|t^2)^{k_2}\Big)^{(k)} = \Big(\frac{t^{2k_1+5+2\beta}}{1-|\lambda|t^2}\Big)^{(k)}.
$$

Noting that $(\frac{t^{2k_1+5+2\beta}}{1-|\lambda|t^2})^{(k)}$ is convergent for $\alpha = 1$ and $\beta \in \mathbb{N}$, then

$$
\begin{aligned}
&\le \frac{c}{1+|\lambda|t^2}\Big(\sum_{k_1=0}^{\infty}|\lambda|^{k_1}\frac{(2k_1+1+\beta)!}{[(2k_1+1)+1]!}\cdot\frac{t^{2k_1+5+2\beta}}{1-|\lambda|t^2}\Big)^{(k)}\\
&\le \frac{c}{1+|\lambda|t^2}\Big(\frac{t^{\beta+2}}{1-|\lambda|t^2}\sum_{k_1=0}^{\infty}([(2k_1+1)+1]+1)\times\cdots\\
&\times(2k_1+1+\beta)t^{2k_1+\beta+3}\Big)^{(k)}\\
&\le \frac{c}{1+|\lambda|t^2}\left(\frac{t^{\beta+2}}{1-|\lambda|t^2}(t^{\beta+3}\sum_{k_1=0}^{\infty}t^{2k_1})^{(l)}\right)^{(k)},
\end{aligned}
$$

where $l = (2k_1+\beta) - [(2k_1+3)+1]$. We have

$$\leq \frac{c}{1+|\lambda|t^2}[\frac{t^{\beta+2}}{1-|\lambda|t^2}(t^{\beta+3}\frac{1}{1-t^2})^{(l)}]^{(k)} < \infty.$$

the proof is completed. □

The reader can prove the convergence of triple series and so on in the same manner.

5 Conclusion

In this paper, we have considered the analytical and numerical solutions of the FSLPs and eigenvalue problems for the fractional differential equations with Dirichlet boundary conditions by using an iterative method. Moreover, we proved that the resulting series is convergent.

References

1. Abbasbandy, S. and Shirzadi, A. *Homotopy analysis method for multiple solutions of the fractional Sturm-Liouville problems*, Numer. Algorithms, 54(4) (2010) 521–532.

2. Al-Mdallal, Q.M. *An efficient method for solving fractional Sturm–Liouville problems*, Chaos, Solitons Fractals, 40(1) (2009) 183–189.

3. Al-Mdallal, Q.M. *On the numerical solution of fractional Sturm–Liouville problems*, Int. J. Comput. Math., 87(12) (2010) 2837–2845.

4. Ansari, A. *On finite fractional Sturm–Liouville transforms*, Integral Transform Spec. Funct. 26(1) (2015) 51–64.

5. Dastmalchi Saei, F., Abbasi, S. and Mirzay, Z. *Inverse Laplace transform method for multiple solutions of the fractional Sturm-Liouville problems*, Computational Methods for Differential Equations, 2(1) (2014) 56–61.

6. Dehghan, M. and Mingarelli, A.B. *Fractional Sturm-Liouville eigenvalue problems, I*, RACSAM 114, 46 (2020).

7. Dehghan, M. and Mingarelli, A.B. *Fractional Sturm-Liouville eigenvalue problems, II*, arXiv preprint arXiv:1712.09894, 2017.

8. Duan, J.-S. *Time-and space-fractional partial differential equations*, J. Math. Phys., 46(1) (2005) 013504.

9. Duan, J.-S., Rach, R., Baleanu, D. and Wazwaz, A.-M. *A review of the Adomian decomposition method and its applications to fractional differential equations*, Communications in Fractional Calculus, 3(2) (2012) 73–99.

10. Duan, J.-S.,Wang, Z., Liu, Y.-L. and Qiu, X. *Eigenvalue problems for fractional ordinary differential equations*, Chaos Solitons Fractals 46 (2013), 46–53.

11. Erdélyi, A., Magnus, W., Oberhettinger, F. and Tricomi, F.G. *Higher transcendental functions*,Vol. III. Based, in part, on notes left by Harry Bateman. McGraw-Hill Book Company, Inc., New York-Toronto-London, 1955.

12. Eshaghi, S. and Ansari, A. *Finite fractional Sturm–Liouville transforms for generalized fractional derivatives*, Iran J. Sci. Technol. Trans. Sci. 41(4) (2017) 931–937.

13. Gorenflo, R., Kilbas, A.A., Mainardi, F., and Rogosin, S.V. a *Mittag-Leffler functions, related topics and applications*, volume 2. Springer, 2014.

14. Gorenflo, R. and Mainardi, F. *Fractional oscillations and Mittag-Leffler functions*, Citeseer, 1996.

15. Joseph, K., Ralf, M. and Cheng, L.S. *Fractional dynamics: Recent advances*, World Scientific, 2011.

16. Kilbas, A.A. and Srivastava, H.M. *Theory and applications of fractional differential equations*, North-Holland Mathematics Studies, 204. Elsevier Science B.V., Amsterdam, 2006.

17. Mainardi, F. *Fractional calculus and waves in linear viscoelasticity: An introduction to mathematical models*, World Scientific, 2010.

18. Mainardi, F. *Fractional calculus: some basic problems in continuum and statistical mechanics*, arXiv:1201.0863, 2012.

19. Miller, K.S. and Ross, B. *An introduction to the fractional calculus and fractional differential equations*, John-Wiley and sons. Inc. New York, 1993.

20. Mittag-Leffler, G.M. *Sur la nouvelle fonction* $E\alpha(x)$, CR Acad. Sci. Paris, 137(2) (1903) 554–558.

21. Oldham, K.B. and Spanier, J. *The fractional calculus. Theory and applications of differentiation and integration to arbitrary order.* With an annotated chronological bibliography by Bertram Ross. Mathematics in Science and Engineering, Vol. 111. Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1974.

22. Podlubny, I. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, 1999. vol. 198 of Mathematics in Science and Engineering.

23. Rossikhin, Y.A. and Shitikova, M.V. *Application of fractional calculus for dynamic problems of solid mechanics: novel trends and recent results*, ASME. Appl. Mech. Rev. , 63(1) (2010) 010801.

24. Syam, M.I., Al-Mdallal, Q.M. and Al-Refai, M. *A numerical method for solving a class of fractional Sturm-Liouville eigenvalue problems*, Commun. Num. Anal., (2017) 217–232.

25. Zayernouri, M. and Karniadakis, G.E. *Fractional Sturm–Liouville eigenproblems: Theory and numerical approximation*, J. Comput. Phys., 252 (2013) 495–517.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 135–157
DOI:10.22067/ijnao.2021.11300.0

Research Article

Using approximate endpoint property on existing solutions for two inclusion problems of the fractional q-differential†

G.K. Ranjbar* and M.E. Samei

Abstract

Using the approximate endpoint property, we describe a technique for existing solutions of the fractional q-differential inclusion with boundary value conditions on multifunctions. For this, we use an approximate endpoint result on multifunctions. Also, we give an example to elaborate on our results and to present the obtained results by fractional calculus.

AMS subject classifications (2020): 34A08, 39A13, 34A60.

Keywords: Approximate endpoint; Fractional q-differential inclusion; Boundary value conditions.

1 Introduction

Fractional calculus and q-calculus are some of the significant branches in mathematical analysis. The field of fractional calculus has countless applications (for instance, consider [2, 10, 32]). Similarly, the subject of fractional differential equations ranges from the theoretical views of the existence and uniqueness of solutions to analytical methods (for more details, see [4, 5]). There has been intensive development in fractional differential equations and

*Corresponding author
Received 1 January 2020; revised 16 November 2020; accepted 15 January 2021
Ghorban Kalilzadeh Ranjbar
Department of Mathematics, Faculty of Science, Bu-Ali Sina University, Hamedan, Iran.
e-mail: ghkhalilzadeg@yahoo.com

Mohammad Esmael Samei
Department of Mathematics, Faculty of Science, Bu-Ali Sina University, Hamedan, Iran.
e-mail: mesamei@basu.ac.ir; mesamei@gmail.com

† This article was suggested by the Scientific Committee of the "Third Iranian Seminar on Control and Optimization" for publication in IJNAO, which was accepted after independent review.

inclusion (for example, see [9, 13, 38, 36, 35, 37, 29, 20, 34]). Also, great attention was devoted to the fractional differential inclusions (for more information, see [4, 7, 12, 30, 40]). For finding more details about elementary notions and definitions of fractional differential equations one can study, for instance, [6, 26, 31].

In this article, motivated by [8, 33] and among these achievements, we try to stretch out the problem in a sense for the fractional q-differential inclusions with integral boundary conditions, in conformity with the definition of the fractional Caputo type q-derivative of order α and the fractional Riemann–Liouville type q-integral. We state the basic definitions and some properties of fractional q-differential inclusion from [18]. For this purpose, we consider and discuss the inclusion problem

$$ {}^{c}\mathbb{D}_q^{\alpha} u(t) \in \mathcal{W}\left(t, u(t), u'(t), u''(t)\right), \tag{1}$$

under conditions $u(0) + u(p) + u(1) = \int_0^1 f_0(s, u(s))\,\mathrm{d}s$ and

$$\begin{cases} {}^{c}\mathbb{D}_q^{\beta} u(0) + {}^{c}\mathbb{D}_q^{\beta} u(p) + {}^{c}\mathbb{D}_q^{\beta} u(1) = \int_0^1 f_1(s, u(s))\,\mathrm{d}s, \\ {}^{c}\mathbb{D}_q^{\gamma} u(0) + {}^{c}\mathbb{D}_q^{\gamma} u(p) + {}^{c}\mathbb{D}_q^{\gamma} u(1) = \int_0^1 f_2(s, u(s))\,\mathrm{d}s, \end{cases} \tag{2}$$

where $\alpha \in (2,3]$, $0 < q, p, \beta < 1$, $\gamma \in (1,2)$, $f_i : J \times \mathbb{R} \to \mathbb{R}$ for $i = 1,2,3$, are continuous functions, $\mathcal{W} : J \times \mathbb{R}^3 \to P_{cp}(\mathbb{R})$ is a multifunction, and ${}^{c}\mathbb{D}_q^{\beta}$ is the fractional Caputo type q-derivative for $t \in J = [0,1]$. The set of all compact subsets of $\mathbb{R}$ is denoted by $P_{cp}(\mathbb{R})$. Also, we look into the existence of solutions for the fractional q-differential inclusion problem on the multifunction $\mathcal{W} : J \times \mathbb{R}^{n+1} \to P(\mathbb{R})$,

$$ {}^{c}\mathbb{D}_q^{\alpha} u(t) \in \mathcal{W}\left(t, u(t), {}^{c}\mathbb{D}_q^{\gamma_1} u(t), \dots, {}^{c}\mathbb{D}_q^{\gamma_n} u(t)\right), \tag{3}$$

with conditions

$$\begin{cases} u'(0) + a_1 u'(1) = \sum_{i=1}^{n} {}^{c}\mathbb{D}_q^{\gamma_i} u(p), \\ u(0) + a_2 u(1) = \sum_{i=1}^{n} \mathbb{I}_q^{\gamma_i} u(p), \end{cases} \tag{4}$$

where $\alpha \in (1,2]$, $0 < q, p, \gamma_i < 1$, $\alpha - \gamma_i \in [1, \infty)$ for all $1 \le i \le n$,

$$ a_1 > \sum_{i=1}^{n} \frac{p^{1-\gamma_i}}{\Gamma_q(2-\gamma_i)}, \quad a_2 > \sum_{i=1}^{n} \frac{p^{\gamma_i+1}}{\Gamma_q(\gamma_i+2)}, $$

here $n \ge 1$ and $t \in J = [0,1]$.

As before, we remind some of the previous works briefly. In 1910, the subject of q-difference equations was introduced by Jackson [22, 24, 23]. After that, at the beginning of the last century, studies on q-difference equations have been appeared in many works, especially in [1, 3, 15, 28, 39]. An excellent account in the study of fractional differential and q-differential equations can be found in [21, 25, 26].

2 Preliminaries

We recall from [14, 25] some basic definitions, notation, and results of q-fractional calculus, which are needed throughout this article. In fact, we consider the fractional q-calculus on the specific time scale $\mathbb{T} = \mathbb{R}$, where $\mathbb{T}_{s_0} = \{0\} \cup \{s : s = s_0 q^n\}$, for a nonnegative integer n, $s_0 \in \mathbb{R}$, and $q \in J_0$. Let $a \in \mathbb{R}$. Define $[a]_q = (1-q^a)/(1-q)$ [24]. The power function $(y-z)_q^n$ with $n \in \mathbb{N}_0$ is defined by $(y-z)_q^{(n)} = \prod_{k=0}^{n-1}(y-zq^k)$, for $n \geq 1$ and $(y-z)_q^{(0)} = 1$, where y and z are real numbers and $\mathbb{N}_0 := \{0\} \cup \mathbb{N}$ [1]. Also, for $\sigma \in \mathbb{R}$ and $q \neq 0$, we have $(y-z)_q^{(\sigma)} = v^\sigma \prod_{k=0}^{\infty}(y-zq^k)/(y-zq^{\sigma+k})$. If $z = 0$, then it is clear that $y^{(\sigma)} = y^\sigma$ [11] (Algorithm 1). The q-Gamma function is given by $\Gamma_q(\sigma) = (1-q)^{(\sigma-1)}/(1-q)^{\sigma-1}$, where $\sigma \in \mathbb{R} \backslash \{0, -1, -2, \ldots\}$ [24] (Algorithm 2). Note that, $\Gamma_q(\sigma+1) = [\sigma]_q \Gamma_q(\sigma)$. The q-derivative of function $\mathfrak{g}$ is defined by

$$\mathbb{D}_q[\mathfrak{g}](\tau) = \frac{\mathfrak{g}(\tau) - \mathfrak{g}(q\tau)}{(1-q)\tau},$$

and $\mathbb{D}_q[\mathfrak{g}](0) = \lim_{\tau \to 0} \mathbb{D}_q[\mathfrak{g}](\tau)$, which is shown in Algorithm 3 [1]. Furthermore, the higher order q-derivative of a function $\mathfrak{g}$ is defined by $\mathbb{D}_q^n[\mathfrak{g}](\tau) = \mathbb{D}_q[\mathbb{D}_q^{n-1}[\mathfrak{g}]](\tau)$, for $n \geq 1$, where $\mathbb{D}_q^0[\mathfrak{g}](\tau) = \mathfrak{g}(\tau)$ [1]. The q-integral of a function $\mathfrak{g}$ is defined on $[0, b]$ by

$$\mathbb{I}_q[\mathfrak{g}](\tau) = \int_0^\tau \mathfrak{g}(\xi)\, \mathrm{d}_q\xi = \tau(1-q)\sum_{k=0}^{\infty} q^k \mathfrak{g}(\tau q^k),$$

for $0 \leq \tau \leq b$, provided the series is absolutely converges [1]. If τ in $[0, T]$, then

$$\int_\tau^T \mathfrak{g}(\xi)\, \mathrm{d}_q\xi = \mathbb{I}_q[\mathfrak{g}](T) - \mathbb{I}_q[\mathfrak{g}](\tau) = (1-q)\sum_{k=0}^{\infty} q^k \left[T\mathfrak{g}(Tq^k) - \tau\mathfrak{g}(\tau q^k)\right],$$

whenever the series exists. The operator $\mathbb{I}_q^n$ is given by $\mathbb{I}_q^0[\mathfrak{g}](\tau) = \mathfrak{g}(\tau)$ and $\mathbb{I}_q^n[\mathfrak{g}](\tau) = \mathbb{I}_q[\mathbb{I}_q^{n-1}[\mathfrak{g}]](\tau)$ for $n \geq 1$ and $\mathfrak{g} \in C([0, T])$ [1]. It has been proved that $\mathbb{D}_q[\mathbb{I}_q[\mathfrak{g}]](\tau) = \mathfrak{g}(\tau)$ and $\mathbb{I}_q[\mathbb{D}_q[\mathfrak{g}]](\tau) = \mathfrak{g}(\tau) - \mathfrak{g}(0)$ whenever $\mathfrak{g}$ is continuous at $\tau = 0$ [1]. The fractional Riemann–Liouville type q-integral of the function $\mathfrak{g}$ on $J_0 = (0, 1)$ for $\sigma \geq 0$ is defined by $\mathbb{I}_q^0[\mathfrak{g}](\tau) = \mathfrak{g}(\tau)$ and

$$\begin{aligned}
\mathbb{I}_q^\sigma[\mathfrak{g}](\tau) &= \frac{1}{\Gamma_q(\sigma)} \int_0^\tau (\tau - q\xi)^{(\sigma-1)} \mathfrak{g}(\xi)\, \mathrm{d}_q\xi \\
&= \tau^\sigma (1-q)^\sigma \sum_{k=0}^{\infty} q^k \frac{\prod_{i=1}^{k-1}\left(1 - q^{\sigma+i}\right)}{\prod_{i=1}^{k-1}\left(1 - q^{i+1}\right)} \mathfrak{g}(\tau q^k),
\end{aligned} \tag{5}$$

for $t \in \overline{J}_0$ [17, 10](Algorithm 4). The Caputo fractional q-derivative of a function $\mathfrak{g}$ is defined by

$$
\begin{aligned}
{}^{c}\mathbb{D}_q^{\sigma}[\mathfrak{g}](\tau) &= \mathbb{I}_q^{[\sigma]-\sigma}[\mathbb{D}_q^{[\sigma]}[\mathfrak{g}]](\tau) \\
&= \frac{1}{\Gamma_q\left([\sigma]-\sigma\right)} \int_0^{\tau} (\tau - q\xi)^{([\sigma]-\sigma-1)} \mathbb{D}_q^{[\sigma]}[\mathfrak{g}](\xi)\, \mathrm{d}_q q\xi,
\end{aligned} \tag{6}
$$

where $t \in \overline{J}_0$ and $\sigma > 0$ [17]. It has been proved that $\mathbb{I}_q^{\nu}[\mathbb{I}_q^{\sigma}[\mathfrak{g}]](\tau) = \mathbb{I}_q^{\sigma+\nu}[\mathfrak{g}](\tau)$, and $\mathcal{D}_q^{\sigma}[\mathbb{I}_q^{\sigma}[\mathfrak{g}]](\tau) = \mathfrak{g}(\tau)$, where $\sigma, \nu \geq 0$ [17]. Algorithm 4 shows pesudo-code $\mathbb{I}_q^{\sigma}[\mathfrak{g}](\tau)$.

We say a multifunction $G : J \to P_{cl}(\mathbb{R})$ is measurable whenever for each real number y, the function $t \mapsto d(y, G(t))$ is measurable [16]. The Pompeiu–Hausdorff metric $H_d \colon 2^X \times 2^X \to [0, \infty)$ on a metric space (X, ρ) is defined by

$$
H_\rho(A, B) = \max\Big\{ \sup_{a \in A} \rho(a, B), \sup_{b \in B} \rho(A, b) \Big\},
$$

where $\rho(A, b) = \inf_{a \in A} \rho(a, b)$ [19]. The set of closed and bounded of X, and the set of closed subsets of X are denoted by $CB(X)$ and $C(X)$, respectively. In this case, $(CB(X), H_\rho)$ and $(C(X), H_\rho)$ are a metric space and a generalized metric space, respectively [27]. An element $z \in X$ is called an endpoint of multifunction $\mathcal{W} : X \to 2^X$ whenever $\mathcal{W}z = \{z\}$ [8]. Also, the multifunction $\mathcal{W}$ has the approximate endpoint property whenever $\inf_{x \in X} \sup_{y \in \mathcal{W}x} \rho(x, y) = 0$ [8]. A function $\theta : \mathbb{R} \to \mathbb{R}$ is called upper semi-continuous whenever $\limsup_{n\to\infty} \theta(\lambda_n) \leq \theta(\lambda)$ for all sequence $\{\lambda_n\}_{n\geq 1}$ with $\lambda_n \to \lambda$ [8].

Lemma 1. [8] Consider an upper semi-continuous function $\theta : [0, \infty) \to [0, \infty)$ such that $\theta(t) < t$ and $\liminf_{t\to\infty}(t - \theta(t)) > 0$, for all $t > 0$. Also, Assume that (X, ρ) a complete metric space and that $\mathcal{W} : X \to CB(X)$ is a multifunction such that $H_d(\mathcal{W}(x), \mathcal{W}() \leq \theta(\rho(x, y))$, for all $x, y \in X$. Then $\mathcal{W}$ has a unique endpoint if and only if $\mathcal{W}$ has the approximate endpoint property.

3 Main results

Right away, we are ready to state and prove our main results. Foremost, we make the adjacent one.

Lemma 2. Suppose that $v \in C(J, \mathbb{R})$, that $\alpha \in (2, 3]$, that $0 < \beta, q, p < 1$, that $\gamma \in (1, 2)$, and that $f_i : J \times \mathbb{R} \to \mathbb{R}$, for $i = 0, 1, 2$, are continuous functions. The unique solution of the fractional q-differential problem

$$
{}^{c}\mathbb{D}_q^{\alpha} u(t) = v(t), \tag{7}
$$

with conditions (2) is given by

$$
\begin{aligned}
u(t) &= \mathbb{I}_q^{\alpha} v(t) + \frac{1}{3}\int_0^1 f_0(s,u(s))\,\mathrm{d}s - \frac{1}{3}\left[\mathbb{I}_q^{\alpha} v(1) + \mathbb{I}_q^{\alpha} v(p)\right] \\
&\quad + a_1(t)\int_0^1 f_1(s,u(s))\,\mathrm{d}s + a_2(t)\left[\mathbb{I}_q^{\alpha-\beta} v(1) + \mathbb{I}_q^{\alpha-\beta} v(p)\right] \\
&\quad + (b_1 + a_3(t))\int_0^1 g_2(s,u(s))\,\mathrm{d}s \\
&\quad + (b_2 + a_4(t))\left[\mathbb{I}_q^{\alpha-\gamma} v(1) + \mathbb{I}_q^{\alpha-\gamma} v(p)\right],
\end{aligned}
$$

where

$$
\begin{aligned}
a_1(t) &= \frac{3t\Gamma_q(2-\beta) - (p+1)\Gamma_q(2-\beta)}{3(p^{1-\beta}+1)}, \\
a_2(t) &= \frac{(p+1)\Gamma_q(2-\beta) - 3\Gamma_q(2-\beta)t}{3(p^{1-\beta}+1)}, \\
a_3(t) &= \frac{-6(p^{2-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(2-\beta)t}{6(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)} \\
&\quad + \frac{3(p^{1-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(3-\beta)t^2}{6(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)}, \\
a_4(t) &= \frac{6(p^{2-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(2-\beta)t}{6(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)} \\
&\quad - \frac{3\Gamma_q(3-\gamma)\Gamma_q(3-\beta)(p^{1-\beta}+1)t^2}{6(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)}, \\
b_1 &= \frac{2(p+1)(p^{2-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(2-\beta)}{6(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)} \\
&\quad - \frac{(p^2+1)\Gamma_q(3-\gamma)(p^{1-\beta}+1)\Gamma_q(3-\beta)}{6(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)}, \\
b_2 &= \frac{(p^2+1)\Gamma_q(3-\gamma)(p^{1-\beta}+1)\Gamma_q(3-\beta)}{6(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)} \\
&\quad - \frac{2(p+1)(p^{2-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(2-\beta)}{6(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)}.
\end{aligned}
\tag{8}
$$

Proof. As we have known that the general solution of (7) is

$$
u(t) = \frac{1}{\Gamma_q(\alpha)}\int_0^t (t-qs)^{(\alpha-1)} v(s)\,\mathrm{d}_q s + c_0 + c_1 t + c_2 t^2, \tag{9}
$$

where c_i $(i = 0,1,2)$ are arbitrary real constants [26, 31, 38]. Thus,

$$\begin{aligned}
{}^{c}\mathbb{D}_q^{\beta}u(t) &= \mathbb{I}_q^{\alpha-\beta}v(t) + \frac{c_1 t^{1-\beta}}{\Gamma_q(2-\beta)} + \frac{2c_2 t^{2-\beta}}{\Gamma_q(3-\beta)},\\
{}^{c}\mathbb{D}_q^{\gamma}u(t) &= \mathbb{I}_q^{\alpha-\gamma}v(t) + \frac{2c_2 t^{2-\gamma}}{\Gamma_q(3-\gamma)}.
\end{aligned}$$

Hence, we get

$$u(0) + u(p) + u(1) = 3c_0 + (1+p)c_1 + (1+p^2)c_2 + \mathbb{I}_q^{\alpha}v(1) + \mathbb{I}_q^{\alpha}v(p)$$

and

$$\begin{aligned}
{}^{c}\mathbb{D}_q^{\beta}u(0) + {}^{c}\mathbb{D}_q^{\beta}u(p) + {}^{c}\mathbb{D}_q^{\beta}u(1) &= c_1\frac{p^{1-\beta}+1}{\Gamma_q(2-\beta)} + c_2\frac{2(p^{2-\beta}+1)}{\Gamma_q(3-\beta)}\\
&\quad + \mathbb{I}_q^{\alpha-\beta}v(1) + \mathbb{I}_q^{\alpha-\beta}v(p),\\
{}^{c}\mathbb{D}_q^{\gamma}u(0) + {}^{c}\mathbb{D}_q^{\gamma}u(p) + {}^{c}\mathbb{D}_q^{\gamma}x(1) &= c_2\frac{2(p^{2-\gamma}+1)}{\Gamma_q(3-\gamma)}\\
&\quad + \mathbb{I}_q^{\alpha-\gamma}v(1) + \mathbb{I}_q^{\alpha-\gamma}v(p).
\end{aligned}$$

By employing the boundary conditions, we have

$$\begin{aligned}
3c_0 + (1+p)c_1 + (1+p^2)c_2 &= \int_0^1 f_0(s,u(s))\,\mathrm{d}s\\
&\quad - \mathbb{I}_q^{\alpha}v(1) - \mathbb{I}_q^{\alpha}v(p),\\
c_1\frac{p^{1-\beta}+1}{\Gamma_q(2-\beta)} + c_2\frac{2(p^{2-\beta}+1)}{\Gamma_q(3-\beta)} &= \int_0^1 f_1(s,x(s))\,\mathrm{d}s\\
&\quad - \mathbb{I}_q^{\alpha-\beta}v(1) - \mathbb{I}_q^{\alpha-\beta}v(p),\\
c_2\frac{2(p^{2-\gamma}+1)}{\Gamma_q(3-\gamma)} &= \int_0^1 f_2(s,x(s))\,\mathrm{d}s\\
&\quad - \mathbb{I}_q^{\alpha-\gamma}v(1) - \mathbb{I}_q^{\alpha-\gamma}v(p).
\end{aligned}$$

This is a linear system of equations of triangular shape, having c_0, c_1, and c_2 as unknowns. By a back substitution, we obtain

$$
\begin{aligned}
c_0 = {} & \frac{1}{3}\int_0^1 f_0(s,u(s))\,\mathrm{d}s - \frac{1}{3}\left[\mathbb{I}_q^{\alpha} v(1) + \mathbb{I}_q^{\alpha} v(p)\right] \\
& - \frac{\Gamma_q(2-\beta)(p+1)}{3(p^{1-\beta}+1)}\int_0^1 f_1(s,u(s))\,\mathrm{d}s \\
& + \frac{(p+1)\Gamma_q(2-\beta)}{3(p^{1-\beta}+1)}\left[\mathbb{I}_q^{\alpha-\beta} v(1) + \mathbb{I}_q^{\alpha-\beta} v(p)\right] \\
& + b_1\int_0^1 f_2(s,u(s))\,\mathrm{d}s + b_2\left[\mathbb{I}_q^{\alpha-\gamma} v(1) + \mathbb{I}_q^{\alpha-\gamma} v(p)\right], \\
c_1 = {} & \frac{\Gamma_q(2-\beta)}{(p^{1-\beta}+1)}\int_0^1 f_1(s,u(s))\,\mathrm{d}s \\
& - \frac{\Gamma_q(2-\beta)}{(p^{1-\beta}+1)}\left[\mathbb{I}_q^{\alpha-\beta} v(1) + \mathbb{I}_q^{\alpha-\beta} v(p)\right] \\
& - \frac{(p^{2-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(2-\beta)}{(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)}\int_0^1 f_2(s,u(s))\,\mathrm{d}s \\
& + \frac{(p^{2-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(2-\beta)}{(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)}\left[\mathbb{I}_q^{\alpha-\gamma} v(1) + \mathbb{I}_q^{\alpha-\gamma} v(p)\right], \\
c_2 = {} & \frac{\Gamma_q(3-\gamma)}{2(p^{2-\gamma}+1)}\int_0^1 f_2(s,u(s))\,\mathrm{d}s \\
& - \frac{\Gamma_q(3-\gamma)}{2(p^{2-\gamma}+1)}\left[\mathbb{I}_q^{\alpha-\gamma} v(1) + \mathbb{I}_q^{\alpha-\gamma} v(p)\right].
\end{aligned}
$$

At once, we replace c_0, c_1, and c_2 in (9) and find the solution $u(t)$ as we stated. □

Assume that $\mathcal{X} = C^2(J)$ endowed with the norm

$$\|u\| = \sup_{t\in J}|u(t)| + \sup_{t\in J}|u'(t)| + \sup_{t\in J}|u''(t)|.$$

Then $(\mathcal{X}, \|\cdot\|)$ is a Banach space (see [16]). For $u \in \mathcal{X}$, we define the selection set $S_{\mathcal{W},u}$ by the set of all $v \in L^1(J)$ such that $v(t) \in \mathcal{W}(t, u(t), u'(t), u''(t))$ for all $t \in J$. For the study of problem (1) and (2), we shall consider the following conditions:

(C1) The multifunction $\mathcal{W} : J \times \mathbb{R}^3 \to P_{cp}(\mathbb{R})$ be is an integrable and bounded such that $\mathcal{W}(\cdot, x_1, x_2, x_3) : J \to P_{cp}(\mathbb{R})$ is measurable for all $x_i \in \mathbb{R}$;

(C2) The functions $f_i : J \times \mathbb{R} \to \mathbb{R}$ be are continuous and map $\theta : [0,\infty) \to [0,\infty)$ be a nondecreasing upper semi-continuous such that

$$\liminf_{t\to\infty}(t - \theta(t)) > 0$$

and $\theta(t) < t$ for all $t > 0$;

(C3) There exist $m, m_0, m_1, m_2 \in C(J, [0, \infty))$ such that

$$\begin{aligned} &H_d\left(\mathcal{W}(t, x_1, x_2, x_3), \mathcal{W}(t, x_1', x_2', x_3')\right) \\ &\leq \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3} m(t) \theta\left(\sum_{k=1}^{3} |x_i - x_i'|\right) \end{aligned}$$

and

$$|f_j(t, x) - f_j(t, x')| \leq \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3} m_j(t) \psi(|x - x'|),$$

for all $t \in J$, $x, x', x_i, x_i' \in \mathbb{R}$, where

$$\begin{aligned} \Lambda_1 = \Bigg[&\frac{\|m\|_\infty}{\Gamma_q(\alpha+1)} + \frac{\|m_0\|_\infty}{3} + \frac{2\|m\|_\infty}{3\Gamma_q(\alpha+1)} \\ &+ \frac{5\Gamma_q(2-\beta)\|m_1\|_\infty}{3} + \frac{10\Gamma_q(2-\beta)\|m\|_\infty}{3\Gamma_q(\alpha-\beta+1)} \\ &+ 10\big(2\Gamma_q(2-\beta) + \Gamma_q(3-\beta)\big) \\ &\times \left(\frac{\Gamma_q(3-\gamma)\left(\|m_2\|_\infty \Gamma_q(\alpha-\gamma+1) + 2\|m\|_\infty\right)}{3\Gamma_q(3-\beta)\Gamma_q(\alpha-\gamma+1)} \right) \Bigg], \\ \Lambda_2 = \Bigg[&\frac{\|m\|_\infty}{\Gamma_q(\alpha)} + \frac{2\Gamma_q(2-\beta)\|m\|_\infty}{\Gamma_q(\alpha-\beta+1)} + \big(2\Gamma_q(2-\beta) + \Gamma_q(3-\beta)\big) \\ &\times \left(\frac{\Gamma_q(3-\gamma)\left(\|m_2\|_\infty \Gamma_q(\alpha-\gamma+1) + 2\|m\|_\infty\right)}{\Gamma(3-\beta)\Gamma_q(\alpha-\gamma+1)} \right) \Bigg], \\ \Lambda_3 = \Bigg[&\frac{\|m\|_\infty}{\Gamma_q(\alpha-1)} + \frac{\Gamma_q(3-\gamma)\left(\|m_2\|_\infty \Gamma_q(\alpha-\gamma+1) + 2\|m\|_\infty\right)}{\Gamma_q(\alpha-\gamma+1)} \Bigg]; \end{aligned}$$

(C4) Multifunction $N : \mathcal{X} \to 2^{\mathcal{X}}$ is given by

$$N(u) = \{h \in \mathcal{X} \mid \exists\, v \in S_{\mathcal{W},u} \ : \ h(t) = w(t)\},$$

for each $t \in J$, where by applying the notation in (8), we have

$$\begin{aligned} w(t) = \mathbb{I}_q^\alpha v(t) &+ \frac{1}{3}\int_0^1 f_0(s, u(s))\,\mathrm{d}s - \frac{1}{3}\left[\mathbb{I}_q^\alpha v(1) + \mathbb{I}_q^\alpha v(p)\right] \\ &+ a_1(t)\int_0^1 f_1(s, u(s))\,\mathrm{d}s + a_2(t)\left[\mathbb{I}_q^{\alpha-\beta} v(1) + \mathbb{I}_q^{\alpha-\beta} v(p)\right] \\ &+ (b_1 + a_3(t))\int_0^1 f_2(s, u(s))\,\mathrm{d}s \\ &+ (b_2 + a_4(t))\left[\mathbb{I}_q^{\alpha-\gamma} v(1) + \mathbb{I}_q^{\alpha-\gamma} v(p)\right]. \end{aligned}$$

Theorem 1. The boundary value q-differential inclusion problem (1) and (2) has a solution, whenever the multifunction $N : \mathcal{X} \to P(\mathcal{X})$ has the approximate endpoint property and conditions (C1)–(C4) are hold.

Proof. We demonstrate that the multifunction N has an endpoint, which is a solution of the problem (1) and (2). Because the multivalued map $t \vdash \mathcal{W}(t, u(t), u'(t), u''(t))$ is measurable and so has closed values, it has measurable selection and so $S_{\mathcal{W},u}$ is nonempty for all $u \in \mathcal{X}$. At present, we show that $N(u) \subset \mathcal{X}$ is closed for $u \in \mathcal{X}$. Let $u \in \mathcal{X}$ and let $\{x_n\}_{n\geq 1}$ be a sequence in $N(u)$ with $u_n \to x$. For each $n \in \mathbb{N}$, choose $v_n \in S_{\mathcal{W},u}$ such that

$$\begin{aligned}x_n(t) = \mathbb{I}_q^{\alpha} v_n(t) &+ \frac{1}{3}\int_0^1 f_0(s, u(s))\,\mathrm{d}s - \frac{1}{3}\left[\mathbb{I}_q^{\alpha} v_n(1) + \mathbb{I}_q^{\alpha} v_n(p)\right]\\
&+ a_1(t)\int_0^1 f_1(s, u(s))\,\mathrm{d}s + a_2(t)\left[\mathbb{I}_q^{\alpha-\beta} v_n(1) + \mathbb{I}_q^{\alpha-\beta} v_n(p)\right]\\
&+ (b_1 + a_3(t))\int_0^1 f_2(s, u(s))\,\mathrm{d}s\\
&+ (b_2 + a_4(t))\left[\mathbb{I}_q^{\alpha-\gamma} v_n(1) + \mathbb{I}_q^{\alpha-\gamma} v_n(p)\right].\end{aligned}$$

It is noteworthy that $\{v_n\}_{n\geq 1}$ has a subsequence that converges to some $v \in L^1(J)$, because $\mathcal{W}$ has compact values. Again, we denote this subsequence by $\{v_n\}_{n\geq 1}$. It is easy to go over that $v \in S_{\mathcal{W},u}$ and $x_n(t)$ tends to $x(t)$, where

$$\begin{aligned}x(t) = \mathbb{I}_q^{\alpha} v(t) &+ \frac{1}{3}\int_0^1 f_0(s, u(s))\,\mathrm{d}s - \frac{1}{3}\left[\mathbb{I}_q^{\alpha} v(1) + \mathbb{I}_q^{\alpha} v(p)\right]\\
&+ a_1(t)\int_0^1 f_1(s, u(s))\,\mathrm{d}s + a_2(t)\left[\mathbb{I}_q^{\alpha-\beta} v(1) + \mathbb{I}_q^{\alpha-\beta} v(p)\right]\\
&+ (b_1 + a_3(t))\int_0^1 f_2(s, u(s))\,\mathrm{d}s\\
&+ (b_2 + a_4(t))\left[\mathbb{I}_q^{\alpha-\gamma} v(1)\mathbb{I}_q^{\alpha-\gamma} v(p)\right],\end{aligned}$$

for each $t \in J$. This implies that $x \in N(u)$ and so N has closed values. Since $\mathcal{W}$ is a compact multivalued map, it is easy to check that $N(u)$ is a bounded set for all $u \in \mathcal{X}$. Now, we show that

$$H_d(N(u), N(v)) \leq \theta(\|u - v\|).$$

Let $u, v \in \mathcal{X}$ and let $h_1 \in N(v)$. Choose $w_1 \in S_{\mathcal{W},v}$ such that

$$\begin{aligned}h_1(t) = \mathbb{I}_q^{\alpha} w_1(t) &+ \frac{1}{3}\int_0^1 f_0(s, v(s))\,\mathrm{d}s - \frac{1}{3}\left[\mathbb{I}_q^{\alpha} w_1(1) + \mathbb{I}_q^{\alpha} w_1(p)\right]\\
&+ a_1(t)\int_0^1 f_1(s, v(s))\,\mathrm{d}s + a_2(t)\left[\mathbb{I}_q^{\alpha-\beta} w_1(1) + \mathbb{I}_q^{\alpha-\beta} w_1(p)\right]\end{aligned}$$

$$+ (b_1 + a_3(t)) \int_0^1 f_2(s, v(s))\, \mathrm{d}s$$
$$+ (b_2 + a_4(t)) \left[\mathbb{I}_q^{\alpha-\gamma} w_1(1) + \mathbb{I}_q^{\alpha-\gamma} w_1(p)\right],$$

for almost all $t \in J$. Put

$$\tilde{\mathcal{W}}_{u(t)} = \mathcal{W}\left(t, u(t), u'(t), u''(t)\right), \quad \tilde{\mathcal{W}}_{v(t)} = \mathcal{W}\left(t, v(t), v'(t), v''(t)\right).$$

Since

$$H_d\left(\tilde{\mathcal{W}}_{u(t)}, \tilde{\mathcal{W}}_{v(t)}\right) \leq \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3} m(t) \times \theta\left(|u(t) - n(t)| + |u'(t) - v'(t)| + |u''(t) - v''(t)|\right),$$

for all $t \in J$, there exists $w \in \tilde{\mathcal{W}}_{u(t)}$ such that

$$|w_1(t) - w| \leq \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3} m(t) \times \theta\left(|u(t) - v(t)| + |u'(t) - v'(t)| + |u''(t) - v''(t)|\right), \tag{10}$$

for all $t \in J$. Consider the multivalued map $G : J \to P(\mathbb{R})$, which defines the set of all $w \in \mathbb{R}$ such that w satisfies in (10). Since w_1 and

$$\varphi = m\theta\big(|u - v| + |u' - v'| + |u'' - v''|\big)\left[\frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3}\right],$$

are measurable, the multifunction

$$G(.) \bigcap \mathcal{W}\left(\cdot, u(\cdot), u'(\cdot), u''(\cdot)\right),$$

is measurable. Choose $w_2(t) \in \mathcal{W}\left(t, u(t), u'(t), u''(t)\right)$ such that

$$|w_1(t) - w_2(t)| \leq \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3} m(t) \times \psi\left(|u(t) - v(t)| + |u'(t) - v'(t)| + |u''(t) - v''(t)|\right),$$

for all $t \in J$. Now, Consider the element $h_2 \in N(u)$, which is defined by

$$
\begin{aligned}
h_2(t) = {} & \mathbb{I}_q^{\alpha} w_2(t) + \frac{1}{3}\int_0^1 f_0(s,u(s))\,\mathrm{d}s - \frac{1}{3}\left[\mathbb{I}_q^{\alpha} w_2(1) + \mathbb{I}_q^{\alpha} w_2(p)\right] \\
& + a_1(t)\int_0^1 f_1(s,u(s))\,\mathrm{d}s + a_2(t)\left[\mathbb{I}_q^{\alpha-\beta} w_2(1) + \mathbb{I}_q^{\alpha-\beta} w_2(p)\right] \\
& + (b_1 + a_3(t))\int_0^1 f_2(s,u(s))\,\mathrm{d}s \\
& + (b_2 + a_4(t))\left[\mathbb{I}_q^{\alpha-\gamma} w_2(1) + \mathbb{I}_q^{\alpha-\gamma} w_2(p)\right],
\end{aligned}
$$

for all $t \in J$. Thus,

$$
\begin{aligned}
|h_1(t) - h_2(t)| \le {} & \mathbb{I}_q^{\alpha}|w_1(t) - w_2(t)| \\
& + \frac{1}{3}\int_0^1 |f_0(s,v(s)) - f_0(s,u(s))|\,\mathrm{d}s \\
& + \frac{1}{3}\left[\mathbb{I}_q^{\alpha}|w_1(1) - w_2(1)|\mathbb{I}_q^{\alpha-1}|w_1(p) - w_2(p)|\,\mathrm{d}s\right] \\
& + |a_1(t)|\int_0^1 |f_1(s,v(s)) - f_1(s,u(s))|\,\mathrm{d}s \\
& + |a_2(t)|\left[\mathbb{I}_q^{\alpha-\beta}|w_1(1) - w_2(1)|\right. \\
& \left. + \mathbb{I}_q^{\alpha-\beta}|w_1(p) - w_2(p)|\right] \\
& + |b_1 + a_3(t)|\int_0^1 |f_2(s,v(s)) - f_2(s,u(s))|\,\mathrm{d}s \\
& + |b_2 + a_4(t)|\left[\mathbb{I}_q^{\alpha-\gamma}|w_1(1) - w_2(1)|\,\mathrm{d}s\right. \\
& \left. + \mathbb{I}_q^{\alpha-\gamma}|w_1(p) - w_2(p)|\,\mathrm{d}s\right] \\
\le {} & \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3}\theta\left(\|u - v\|\right)\left[\frac{\|m\|_\infty}{\Gamma_q(\alpha+1)} + \frac{\|m_0\|_\infty}{3}\right. \\
& + \frac{2\|m\|_\infty}{3\Gamma_q(\alpha+1)} + \frac{5\Gamma_q(2-\beta)\|m_1\|_\infty}{3} \\
& + \frac{10\Gamma_q(2-\beta)\|m\|_\infty}{3\Gamma_q(\alpha-\beta+1)} \\
& + 10\big(2\Gamma_q(2-\beta) + \Gamma_q(3-\beta)\big) \\
& \left. \times\left(\frac{\Gamma_q(3-\gamma)(\|m_2\|_\infty\Gamma_q(\alpha-\gamma+1) + 2\|m\|_\infty)}{3\Gamma_q(3-\beta)\Gamma_q(\alpha-\gamma+1)}\right)\right] \\
= {} & \frac{\Lambda_1}{\Lambda_1 + \Lambda_2 + \Lambda_3}\psi(\|u - v\|),
\end{aligned}
$$

$$
\begin{aligned}
|h_1'(t) - h_2'(t)| \le {} & \mathbb{I}_q^{\alpha-1}|w_1(t) - w_2(t)| \\
& + \frac{\Gamma(2-\beta)}{(p^{1-\beta}+1)}\left[\mathbb{I}_q^{\alpha-\beta}|w_1(1) - w_2(1)|\right.
\end{aligned}
$$

$$
\begin{aligned}
&+\mathbb{I}_q^{\alpha-\beta}|w_1(p)-w_2(p)|\big] \\
&+|a_5(t)|\int_0^1 |f_2(s,v(s))-f_2(s,u(s))|\,\mathrm{d}s \\
&+|a_6(t)|\left[\mathbb{I}_q^{\alpha-\gamma}|w_1(1)-w_2(1)|\right. \\
&\left.+\mathbb{I}_q^{\alpha-\gamma-1}|w_1(p)-w_2(p)|\right] \\
\leq{}& \frac{1}{\Lambda_1+\Lambda_2+\Lambda_3}\theta(\|u-v\|)\left[\frac{\|m\|_\infty}{\Gamma_q(\alpha)}+\frac{2\Gamma_q(2-\beta)\|m\|_\infty}{\Gamma_q(\alpha-\beta+1)}\right. \\
&+(2\Gamma_q(2-\beta)+\Gamma_q(3-\beta)) \\
&\left.\times\left(\frac{\Gamma_q(3-\gamma)\left(\|m_2\|_\infty\Gamma_q(\alpha-\gamma+1)+2\|m\|_\infty\right)}{\Gamma_q(3-\beta)\Gamma_q(\alpha-\gamma+1)}\right)\right] \\
={}& \frac{\Lambda_2}{\Lambda_1+\Lambda_2+\Lambda_3}\theta(\|u-v\|),
\end{aligned}
$$

and

$$
\begin{aligned}
|h_1''(t)-h_2''(t)| \leq{}& \mathbb{I}_q^{\alpha-2}|w_1(t)-w_2(t)| \\
&+\frac{\Gamma_q(3-\gamma)}{(p^{2-\gamma}+1)}\int_0^1 |f_2(s,v(s))-f_2(s,u(s))|\,\mathrm{d}s \\
&+\frac{\Gamma_q(3-\gamma)}{(p^{2-\gamma}+1)}\left[\mathbb{I}_q^{\alpha-\gamma}|w_1(1)-w_2(1)|\right. \\
&\left.+\mathbb{I}_q^{\alpha-\gamma}|w_1(p)-w_2(p)|\right] \\
\leq{}& \frac{1}{\Lambda_1+\Lambda_2+\Lambda_3}\psi(\|u-v\|)\left[\frac{\|m\|_\infty}{\Gamma_q(\alpha-1)}\right. \\
&\left.+\frac{\Gamma_q(3-\gamma)\left(\|m_2\|_\infty\Gamma_q(\alpha-\gamma+1)+2\|m\|_\infty\right)}{\Gamma_q(\alpha-\gamma+1)}\right] \\
={}& \frac{\Lambda_3}{\Lambda_1+\Lambda_2+\Lambda_3}\psi(\|u-v\|),
\end{aligned}
$$

where

$$
\begin{aligned}
a_5(t) ={}& \frac{(p^{2-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(2-\beta)}{(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)} \\
&+\frac{\Gamma_q(3-\gamma)(p^{1-\beta}+1)\Gamma_q(3-\beta)t}{(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)}, \\
a_6(t) ={}& \frac{(p^{2-\beta}+1)\Gamma_q(3-\gamma)\Gamma_q(2-\beta)}{(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)\Gamma_q(\alpha-\gamma)} \\
&-\frac{\Gamma_q(3-\gamma)\Gamma_q(3-\beta)(p^{1-\beta}+1)t}{(p^{1-\beta}+1)(p^{2-\gamma}+1)\Gamma_q(3-\beta)\Gamma_q(\alpha-\gamma)}.
\end{aligned}
$$

Hence,

$$\begin{aligned}\|h_1 - h_2\| &= \sup_{t\in J} |h_1(t) - h_2(t)| + \sup_{t\in J} |h_1'(t) - h_2'(t)| \\ &\quad + \sup_{t\in J} |h_1''(t) - h_2''(t)| \\ &\leq \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3} \theta(\|x-y\|)(\Lambda_1 + \Lambda_2 + \Lambda_3) = \theta(\|x-y\|).\end{aligned}$$

Therefore, it is easy to get that

$$H_d(N(u), N(v)) \leq \theta(\|u - v\|),$$

for all $u, v \in \mathcal{X}$. On the other hand, the multifunction N has the approximate endpoint property. By using Lemma 1, there exists $u^* \in \mathcal{X}$ such that $N(u^*) = \{u^*\}$. Thus, by employing Lemma 2, u^* is a solution of problem (1) and (2). □

At present, we investigate the existence of a solution for the fractional q-differential inclusion problem with integral boundary value conditions

$$\begin{cases} {}^c\mathbb{D}_q^{\alpha} u(t) \in \mathcal{W}\left(t, u(t), {}^c\mathbb{D}_q^{\gamma_1} u(t), \dots, {}^c\mathbb{D}_q^{\gamma_n} u(t)\right), \\ u'(0) + a_1 u'(1) = \sum_{i=1}^{n} {}^c\mathbb{D}_q^{\gamma_i} u(p), \\ u(0) + a_2 u(1) = \sum_{i=1}^{n} \mathbb{I}_q^{\gamma_i} u(p), \end{cases} \tag{11}$$

for the multifunction $\mathcal{W} : J \times \mathbb{R}^{n+1} \to P(\mathbb{R})$, where $t \in J$, $\alpha \in (1, 2]$, $n \geq 2$, $0 < q, p, \gamma_i < 1$, $\alpha - \gamma_i \geq 1$ for all $1 \leq i \leq n$, and

$$a_1 > \sum_{i=1}^{n} \frac{p^{1-\gamma_i}}{\Gamma_q(2-\gamma_i)}, \quad a_2 > \sum_{i=1}^{n} \frac{p^{\gamma_i+1}}{\Gamma_q(\gamma_i + 2)}.$$

4 Examples and algorithms for the problems

In this part, we give a complete computational technique for solving problems (1) and (2), such that it covers all the problems and presents numerical examples solving perfect. To this aim, we consider a pseudo-code description of the method for the calculated q-Gamma function of order n in Algorithm 2 (for more details, see the link https://en.wikipedia.org/wiki/Q-gamma_function).

Table 1 shows that when q is constant, the q-Gamma function is an increasing function. Also, for smaller values of x, an approximate result is obtained with a fewer values of n. It has been shown by underlined rows. Table 2 shows that the q-Gamma function for values q near one is obtained with more values of n in comparison with other columns. They have been underlined in line 8 of the first column, line 17 of the second column, and line 29 of the third column of Table 2. Also, Table 3 is the same as Table 2, but x values increase in Table 3. Similarly, the q-Gamma function for values q near to one is obtained with more values of n in comparison with other columns. Furthermore, we provide Algorithm 3 that calculates $(\mathbb{D}_q^{\alpha} f)(x)$. Here, we give an example to illustrate our first main result, which applies to the different values q in Theorem 1.

Example 1. Consider the fractional q-differential inclusion problem

$$\begin{aligned} {}^{c}\mathbb{D}_q^{\frac{9}{4}} u(t) \in \Bigg[0, \frac{t^2}{100} \sin u(t) + \frac{1}{100} \cos u'(t) \\ + \frac{1}{100} \left(\frac{|u''(t)|}{1+|u''(t)|} \right) \Bigg], \end{aligned} \tag{12}$$

under the integral boundary conditions

$$u(0) + u(\frac{3}{4}) + u(1) = \int_0^1 \frac{s^2}{20} \cos u(s) \,\mathrm{d}s$$

and

$$\begin{cases} {}^{c}\mathbb{D}_q^{\frac{2}{3}} u(0) + {}^{c}\mathbb{D}_q^{\frac{2}{3}} u(\frac{3}{4}) + {}^{c}\mathbb{D}_q^{\frac{2}{3}} u(1) = \int_0^1 \frac{e^{s^2-1}}{20} \cos u(s) \,\mathrm{d}s, \\ {}^{c}\mathbb{D}_q^{\frac{5}{3}} u(0) + {}^{c}\mathbb{D}_q^{\frac{5}{3}} u(\frac{3}{4}) + {}^{c}\mathbb{D}_q^{\frac{5}{3}} u(1) = \int_0^1 \frac{2s^3+1}{20\pi} \cos u(s) \,\mathrm{d}s, \end{cases} \tag{13}$$

where $t \in J = [0, 1]$, $\alpha = \frac{9}{4}$, $\beta = \frac{2}{3}$, $\gamma = \frac{5}{3}$, and $p = \frac{3}{4}$ in (1) and (2). Consider the map $\mathcal{W} : J \times \mathbb{R}^3 \to P(\mathbb{R})$ defined by

$$\mathcal{W}(t, x_1, x_2, x_3) = \left[0, \frac{t^2}{100} \sin x_1 + \frac{1}{100} \cos x_2 + \frac{1}{100} \left(\frac{|x_3|}{1+|x_3|} \right) \right].$$

Also, $f_i : J \times \mathbb{R} \to \mathbb{R}$ are define by

$$f_0(t, x) = \frac{t^2}{20} \cos x, \quad f_1(t, x) = \frac{e^{t^2-1}}{20} \cos x, \quad f_2(t, x) = \frac{2t^3+1}{300\pi} \cos x,$$

and $N : C^2(J) \to 2^{C^2(J)}$ is defined by

$$N(u) = \Big\{ h \in C^2(J) \,\Big|\, \exists v \in S_{\mathcal{W},u} : h(t) = w(t) \Big\},$$

for all $t \in J$ such that

$$\begin{aligned} w(t) = \mathbb{I}_q^{\frac{9}{4}} v(t) + \frac{1}{3} \int_0^1 \frac{s^2}{20} \cos u(s) \,\mathrm{d}s - \frac{1}{3} \left[\mathbb{I}_q^{\frac{9}{4}} v(1) + \mathbb{I}_q^{\frac{9}{4}} v(\frac{3}{4}) \right] \\ + a_1(t) \int_0^1 \frac{e^{s^2-1}}{20} \cos u(s) \,\mathrm{d}s + a_2(t) \left[\mathbb{I}_q^{\frac{19}{12}} v(1) + \mathbb{I}_q^{\frac{19}{12}} v(\frac{3}{4}) \right] \\ + (b_1 + a_3(t)) \int_0^1 \frac{2s^3+1}{300\pi} \cos u(s) \,\mathrm{d}s \end{aligned}$$

$$+ (b_2 + a_4(t)) \left[\mathbb{I}_q^{\frac{7}{12}} v(1) + \mathbb{I}_q^{\frac{7}{12}} v(\frac{3}{4}) \right],$$

where

$$\begin{aligned}
a_1(t) &= \frac{3\Gamma_q(\frac{4}{3})t - \frac{7}{4}\Gamma_q(\frac{4}{3})}{3((\frac{3}{4})^{\frac{1}{3}} + 1)}, \\
a_2(t) &= \frac{\frac{7}{4}\Gamma_q(\frac{4}{3}) - 3\Gamma_q(\frac{4}{3})t}{3((\frac{3}{4})^{\frac{1}{3}} + 1)}, \\
a_3(t) &= \frac{-6((\frac{3}{4})^{\frac{4}{3}} + 1)\Gamma_q(\frac{4}{3})\Gamma_q(\frac{4}{3})t + 3((\frac{3}{4})^{\frac{1}{3}} + 1)\Gamma_q(\frac{4}{3})\Gamma_q(\frac{7}{3})t^2}{6((\frac{3}{4})^{\frac{1}{3}} + 1)((\frac{3}{4})^{\frac{1}{3}} + 1)\Gamma_q(\frac{7}{3})}, \\
a_4(t) &= \frac{6((\frac{3}{4})^{\frac{4}{3}} + 1)\Gamma_q(\frac{4}{3})\Gamma_q(\frac{4}{3})t - 3\Gamma_q(\frac{4}{3})\Gamma_q(\frac{7}{3})((\frac{3}{4})^{\frac{1}{3}} + 1)t^2}{6((\frac{1}{3})^{\frac{1}{3}} + 1)((\frac{3}{4})^{\frac{1}{3}} + 1)\Gamma_q(\frac{7}{3})}, \\
b_1 &= \frac{\frac{7}{2}((\frac{3}{4})^{\frac{4}{3}} + 1)\Gamma_q(\frac{4}{3})\Gamma_q(\frac{4}{3}) - ((\frac{3}{4})^2 + 1)\Gamma_q(\frac{4}{3})((\frac{3}{4})^{\frac{1}{3}} + 1)\Gamma_q(\frac{7}{3})}{6((\frac{3}{4})^{\frac{1}{3}} + 1)((\frac{3}{4})^{\frac{1}{3}} + 1)\Gamma_q(\frac{7}{3})}, \\
b_2 &= \frac{((\frac{3}{4})^2 + 1)\Gamma_q(\frac{4}{3})((\frac{3}{4})^{\frac{1}{3}} + 1)\Gamma_q(\frac{7}{3}) - \frac{8}{3}((\frac{3}{4})^{\frac{4}{3}} + 1)\Gamma_q(\frac{4}{3})\Gamma_q(\frac{4}{3})}{6((\frac{3}{4})^{\frac{1}{3}} + 1)((\frac{3}{4})^{\frac{1}{3}} + 1)\Gamma_q(\frac{7}{3})}.
\end{aligned} \tag{14}$$

Put $m(t) = \frac{3t}{20}$, $m_0(t) = \frac{t^2}{20}$, $m_1(t) = \frac{e^{t^2-1}}{20}$, $m_2(t) = \frac{2t^3+1}{300\pi}$, and $\psi(t) = \frac{t}{5}$. Then, we have

$$\begin{aligned}
\Lambda_1 &= \left[\frac{\|m\|_\infty}{\Gamma_q(\frac{13}{4})} + \frac{\|m_0\|_\infty}{3} + \frac{2\|m\|_\infty}{3\Gamma_q(\frac{13}{4})} + \frac{5\Gamma_q(\frac{4}{3})\|m_1\|_\infty}{3} + \frac{10\Gamma_q(\frac{4}{3})\|m\|_\infty}{3\Gamma_q(\frac{31}{12})}. \right. \\
&\quad \left. + \frac{10(2\Gamma_q(\frac{4}{3}) + \Gamma_q(\frac{7}{3}))\Gamma_q(\frac{4}{3}) \left(\|m_2\|_\infty \Gamma_q(\frac{19}{12}) + 2\|m\|_\infty \right)}{3\Gamma_q(\frac{7}{3})\Gamma_q(\frac{19}{12})} \right], \\
\Lambda_2 &= \left[\frac{\|m\|_\infty}{\Gamma_q(\frac{9}{4})} + \frac{2\Gamma_q(\frac{4}{3})\|m\|_\infty}{\Gamma_q(\frac{31}{12})} \right. \\
&\quad \left. + \frac{(2\Gamma_q(\frac{4}{3}) + \Gamma_q(\frac{7}{3}))\Gamma_q(\frac{4}{3}) \left(\|m_2\|_\infty \Gamma_q(\frac{19}{12}) + 2\|m\|_\infty \right)}{\Gamma(\frac{7}{3})\Gamma_q(\frac{19}{12})} \right], \\
\Lambda_3 &= \left[\frac{\|m\|_\infty}{\Gamma_q(\frac{5}{4})} + \frac{\Gamma_q(\frac{4}{3}) \left(\|m_2\|_\infty \Gamma_q(\frac{19}{12}) + 2\|m\|_\infty \right)}{\Gamma_q(\frac{19}{12})} \right].
\end{aligned}$$

In following data of Tables 4 and 5, it is easy to check that

$$H_d(\mathcal{W}(t, u_1, u_2, u_3), F(t, v_1, v_2, v_3)) \leq \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3} m(t)\theta(\sum_{k=1}^{3} |u_k - v_k|)$$

and that

$$|f_j(t,u) - f_j(t,v)| \leq \frac{1}{\Lambda_1 + \Lambda_2 + \Lambda_3} m_j(t)\psi(|u-v|),$$

for $t \in J$, $j = 0, 1, 2$. Because $\sup_{u \in N(0)} \|u\| = 0$, we have

$$\inf_{u \in C^2(J)} \left[\sup_{v \in N(u)} \|u - v\| \right] = 0.$$

Thus, N has the approximate endpoint property. At present, by applying Theorem 1, the system of fractional q-differential inclusions (12) and (13) has at least one solution.

5 Conclusion

The q-differential boundary equations and their applications represent a matter of high interest in the area of fractional q-calculus and its applications in various areas of science and technology. q-differential boundary value problems occur in the mathematical modeling of a variety of physical operations. The end of this article was to investigate a complicated case by utilizing an appropriate basic theory. In this manner, we proved the existence of a solution for familiar problems of q-differential equations under three boundary conditions (1)–(2) and (3)–(4) on a time scale and showed the perfect numerical effects for the problem, which confirm our results.

Acknowledgements

Authors are grateful to the anonymous referees and editor for their constructive comments. The authors were supported by Bu-Ali Sina University.

References

1. Adams, C.R. *The general theory of a class of linear partial q- difference equations*, Trans. Am. Math. Soc. 26 (1924), 283–312.

2. Adams, C.R. *Note on the integro-q-difference equations*, Trans. Am. Math. Soc. 31 (4) (1929), 861–867.

3. Agarwal, R.P. *Certain fractional q-integrals and q-derivatives*, Proceedings of the Cambridge Philosophical Society 66 (1969), 365–370.

4. Agarwal, R.P., Belmekki, M. and Benchohra, M. *A survey on semi- linear differential equations and inclusions involving Riemann-Liouville fractional derivative*, Adv. Differ. Equ. 2009, 981728 (2009).

5. Ahmad, B. and Nieto, J.J. *Riemann-Liouville fractional integro- differential equations with fractional nonlocal integral boundary conditions*, Bound. Value Probl. 2011, 2011:36, 9 pp.

6. Ahmad B. and Ntouyas, S.K. *Existence of solutions for nonlinear fractional q-difference inclusions with nonlocal robin (separated) conditions*, Mediterr. J. Math. 10 (2013), 1333–1351.

7. Ahmad B. and Ntouyas, S.K. *Boundary value problem for fractional differential inclusions with four-point integral boundary conditions*, Surv. Math. Appl. 6 (2011), 175–193.

8. Amini-Harandi, A. *Endpoints of set-valued contractions in metric spaces*, Nonlinear Anal. 72 (2010), 132–134.

9. Anastassiou, G.A. *Principles of delta fractional calculus on time scales and inequalities*, Math. Comput. Modelling 52 (2010), 556–566.

10. Annaby, M.H. and Mansour, Z.S. *q-fractional calculus and equations*, With a foreword by Mourad Ismail. Lecture Notes in Mathematics, 2056. Springer, Heidelberg, 2012.

11. Atici, F. and Eloe, P.W. *Fractional q-calculus on a time scale*, J. Nonlinear Math. Phys. 14 (2007), 333–344.

12. Aubin, J. and Ceuina, A. *Differential inclusions: set-valued maps and viability theory*, Springer-Verlag, 1984.

13. Baleanu, D., Agarwal, R. P., Mohammadi, H. and Rezapour, S. *Some existence results for a nonlinear fractional differential equation on partially ordered banach spaces*, Bound. Value Probl. 2013 (2013), 112 pp.

14. Bohner M. and Peteson, A. *Dynamic equations on time scales*, Birkhauser, Boston, 2001.

15. Carmichael, R.D. *The general theory of linear q-difference equations*, Amer. J. Math. 34 (1912), 147–168.

16. Deimling, K. *Multi-valued differential equations*, Walter de Gruyter, Berlin, 1992.

17. Ferreira, R.A.C. *Nontrivials solutions for fractional q-difference boundary value problems*, Electron. J. Qual. Theory Differ. Equ. 2010, No. 70, 10 pp.

18. Gasper, G.and Rahman, M. *Basi hypergeometric series*, University Press, Cambridge, 1990.

19. Granas, A. and Dugundji, J. *Fixed point theory*, Springer-Verlag, 2005.

20. Hedayati, V. and Samei, M.E. *Positive solutions of fractional differential equation with two pieces in chain interval and simultaneous Dirichlet boundary conditions*, Bound. Value Probl. 2019 (2019), 163 pp.

21. Hilfer, R. *Applications of fractional calculus in physics*, World Scientific, 2000.

22. Jackson, F.H. *On q-functions and a certain difference operator*, Transactions of the Royal Society of Edinburgh 46 (1909), 253–281.

23. Jackson, F.H. *On q-definite integrals*, Pure Appl. Math. Q. 41 (1910), 193–203.

24. Jackson, F.H. *q-difference equations*, Am. J. Math. 32 (1910), 305–314.

25. Kac, V. and Cheung, P. *Quantum calculus*, Universitext, Springer, New York, 2002.

26. Kilbas, A.A., Srivastava, H. M. and Trujillo, J.J. *Theory and applications of fractional differential equations*, North-Holland Mathematics Studies, Elsevier Science, North-Holland, 2006.

27. Kisielewicz, M. *Differential inclusions and optimal control*, Kluwer, Dordrecht, 1991.

28. Mason, T.E. *On properties of the solution of linear q-difference equations with entire function coefficients*, Am. J. Math. 37 (1915), 439–444.

29. Ntouyas, S.K. and Samei, M.E. *Existence and uniqueness of solutions for multi-term fractional q-integro-differential equations via quantum calculus* Adv. Difference Equ. 2019, Paper No. 475, 20 pp.

30. Phung, P. D. and Truong, L. X. *On a fractional differential inclusion with integral boundary conditions in Banach space*, Fract. Calc. Appl. Anal. 16 (2013), 538–558.

31. Podlubny, I. *Fractional differential equations*, Academic Press, 1999.

32. Rajković, P. M., Marinković, S. D. and Stanković, M. S. *Fractional integrals and derivatives in q-calculus*,Appl. Anal. Discrete Math. 1 (2007), 311–323.

33. Rezapour, Sh. and Hedayati, V. *On a Caputo fractional differential inclusion with integral boundary condition for convex-compact and nonconvex- compact valued multifunctions*,Kragujev. J. Math. 41 (1) (2017), 143–158.

34. Samei, M.E. *Existence of solutions for a system of singular sum fractional q-differential equations via quantum calculus*, Adv. Differ. Equ. 2020 (2020), 23.

35. Samei, M.E., Hedayati, V. and Rezapour, Sh. *Existence results for a fraction hybrid differential inclusion with Caputo–Hadamard type fractional derivative*, Adv. Difference Equ. 2019, Paper No. 163, 15 pp.

36. Samei, M.E. and Khalilzadeh Ranjbar, G. *Some theorems of existence of solutions for fractional hybrid q-difference inclusion*, J. Adv. Math. Stud. 12 (2019), 63–76.

37. Samei, M.E., Ranjbar,G.K. and Hedayati, V. *Existence of solutions for equations and inclusions of multi-term fractional q-integro-differential with non-separated and initial boundary conditions*, J. Inequal. Appl. 2019 (2019), 273.

38. Samko, S.G. ,Kilbas, A.A. and Marichev, O.I. *Fractional integrals and derivatives: Theory and applications*, Gordon and Breach Science Publishers, Switzerland; Philadelphia, Pa., USA, 1993.

39. Trjitzinsky, W.J. *Analytic theory of linear q-difference equations,* Acta Math. 61 (1933), 1–38.

40. Wang, J. and Ibrahim, A.G. *Existence and controllability results for nonlocal fractional impulsive differential inclusions in Banach spaces,* Journal of Function Spaces and Applications, 2013, Article ID 518306, 16 pp.

Supporting informations

Table 1: Some numerical results for calculation of $\Gamma_q(x)$ with $q = \frac{1}{3}$ that is constant, $x = 4.5$, 8.4, 12.7 and $n = 1, 2, \ldots, 15$ of Algorithm 2.

n	$x = 4.5$	$x = 8.4$	$x = 12.7$	n	$x = 4.5$	$x = 8.4$	$x = 12.7$
1	2.472950	11.909360	68.080769	9	2.340263	11.257158	64.351366
2	2.383247	11.468397	65.559266	10	2.340250	11.257095	64.351003
3	2.354446	11.326853	64.749894	11	2.340245	11.257074	64.350881
4	2.344963	11.280255	64.483434	12	2.340244	11.257066	64.350841
5	2.341815	11.264786	64.394980	13	2.340243	11.257064	64.350828
6	2.340767	11.259636	64.365536	14	2.340243	11.257063	64.350823
7	2.340418	11.257921	64.355725	15	2.340243	11.257063	64.350822
8	2.340301	11.257349	64.352456				

Table 2: Some numerical results for calculation of $\Gamma_q(x)$ with $q = \frac{1}{3}$, $\frac{1}{2}$, $\frac{2}{3}$, $x = 5$ and $n = 1, 2, \ldots, 35$ of Algorithm 2.

n	$q = \frac{1}{3}$	$q = \frac{1}{2}$	$q = \frac{2}{3}$	n	$q = \frac{1}{3}$	$q = \frac{1}{2}$	$q = \frac{2}{3}$
1	3.016535	6.291859	18.937427	18	2.853224	4.921884	8.476643
2	2.906140	5.548726	14.154784	19	2.853224	4.921879	8.474597
3	2.870699	5.222330	11.819974	20	2.853224	4.921877	8.473234
4	2.859031	5.069033	10.537540	21	2.853224	4.921876	8.472325
5	2.855157	4.994707	9.782069	22	2.853224	4.921876	8.471719
6	2.853868	4.958107	9.317265	23	2.853224	4.921875	8.471315
7	2.853438	4.939945	9.023265	24	2.853224	4.921875	8.471046
8	2.853295	4.930899	8.833940	25	2.853224	4.921875	8.470866
9	2.853247	4.926384	8.710584	26	2.853224	4.921875	8.470747
10	2.853232	4.924129	8.629588	27	2.853224	4.921875	8.470667
11	2.853226	4.923002	8.576133	28	2.853224	4.921875	8.470614
12	2.853224	4.922438	8.540736	29	2.853224	4.921875	8.470578
13	2.853224	4.922157	8.517243	30	2.853224	4.921875	8.470555
14	2.853224	4.922016	8.501627	31	2.853224	4.921875	8.470539
15	2.853224	4.921945	8.491237	32	2.853224	4.921875	8.470529
16	2.853224	4.921910	8.484320	33	2.853224	4.921875	8.470522
17	2.853224	4.921893	8.479713	34	2.853224	4.921875	8.470517

Table 3: Some numerical results for calculation of $\Gamma_q(x)$ with $x = 8.4$, $q = \frac{1}{3}$, $\frac{1}{2}$, $\frac{2}{3}$ and $n = 1, 2, \ldots, 40$ of Algorithm 2.

n	$q = \frac{1}{3}$	$q = \frac{1}{2}$	$q = \frac{2}{3}$	n	$q = \frac{1}{3}$	$q = \frac{1}{2}$	$q = \frac{2}{3}$
1	11.909360	63.618604	664.767669	21	11.257063	49.065390	260.033372
2	11.468397	55.707508	474.800503	22	11.257063	49.065384	260.011354
3	11.326853	52.245122	384.795341	23	11.257063	49.065381	259.996678
4	11.280255	50.621828	336.326796	24	11.257063	49.065380	259.986893
5	11.264786	49.835472	308.146441	25	11.257063	49.065379	259.980371
6	11.259636	49.448420	290.958806	26	11.257063	49.065379	259.976023
7	11.257921	49.256401	280.150029	27	11.257063	49.065379	259.973124
8	11.257349	49.160766	273.216364	28	11.257063	49.065378	259.971192
9	11.257158	49.113041	268.710272	29	11.257063	49.065378	259.969903
10	11.257095	49.089202	265.756606	30	11.257063	49.065378	259.969044
11	11.257074	49.077288	263.809514	31	11.257063	49.065378	259.968472
12	11.257066	49.071333	262.521127	32	11.257063	49.065378	259.968090
13	11.257064	49.068355	261.666471	33	11.257063	49.065378	259.967836
14	11.257063	49.066867	261.098587	34	11.257063	49.065378	259.967666
15	11.257063	49.066123	260.720833	35	11.257063	49.065378	259.967553
16	11.257063	49.065751	260.469369	36	11.257063	49.065378	259.967478
17	11.257063	49.065564	260.301890	37	11.257063	49.065378	259.967427
18	11.257063	49.065471	260.190310	38	11.257063	49.065378	259.967394
19	11.257063	49.065425	260.115957	39	11.257063	49.065378	259.967371
20	11.257063	49.065402	260.066402	40	11.257063	49.065378	259.967357

Table 4: Some numerical results of $\Gamma_q(\alpha)$ in Example 1 with different values of q by Algorithm 2.

n	$\alpha=\frac{3}{4}$	$\alpha=\frac{5}{4}$	$\alpha=\frac{4}{3}$	$\alpha=\frac{19}{12}$	$\alpha=\frac{9}{4}$	$\alpha=\frac{7}{3}$	$\alpha=\frac{31}{12}$	$\alpha=\frac{13}{4}$
	$q=\frac{1}{3}$							
1	1.1174	0.9592	0.9559	0.9673	1.1055	1.1315	1.2199	1.5327
2	1.1311	0.9505	0.9448	0.9500	1.0747	1.0990	1.1824	1.4805
3	1.1356	0.9476	0.9411	0.9444	1.0647	1.0886	1.1703	1.4638
4	1.1371	0.9467	0.9400	0.9425	1.0615	1.0851	1.1664	1.4583
5	1.1376	0.9464	0.9396	0.9419	1.0604	1.0840	1.1651	1.4564
6	1.1377	0.9463	0.9394	0.9417	1.0600	1.0836	1.1646	1.4558
7	1.1378	0.9462	0.9394	0.9417	1.0599	1.0835	1.1645	1.4556
8	1.1378	0.9462	0.9394	0.9416	1.0599	1.0834	1.1644	1.4556
9	1.1378	0.9462	0.9394	0.9416	1.0598	1.0834	1.1644	1.4555
10	1.1378	0.9462	0.9394	0.9416	1.0598	1.0834	1.1644	1.4555
	$q=\frac{1}{2}$							
1	1.1069	0.9743	0.9772	1.0122	1.2620	1.3087	1.4715	2.1039
2	1.1377	0.9526	0.9493	0.9662	1.1655	1.2049	1.3437	1.8906
3	1.1522	0.9426	0.9364	0.9453	1.1221	1.1583	1.2865	1.7960
4	1.1593	0.9378	0.9302	0.9353	1.1015	1.1362	1.2594	1.7514
5	1.1628	0.9355	0.9272	0.9303	1.0915	1.1254	1.2463	1.7297
6	1.1645	0.9343	0.9257	0.9279	1.0865	1.1201	1.2398	1.7190
7	1.1654	0.9337	0.9249	0.9267	1.0841	1.1175	1.2365	1.7137
8	1.1658	0.9334	0.9245	0.9261	1.0828	1.1161	1.2349	1.7111
9	1.1660	0.9333	0.9244	0.9258	1.0822	1.1155	1.2341	1.7098
10	1.1662	0.9332	0.9243	0.9257	1.0819	1.1152	1.2337	1.7091
	$q=\frac{4}{5}$							
1	0.9665	1.1206	1.1787	1.4118	2.6441	2.8906	3.8168	8.5184
2	1.0284	1.0602	1.0963	1.2516	2.1063	2.2761	2.9085	6.0237
3	1.0710	1.0218	1.0443	1.1539	1.8020	1.9312	2.4107	4.7288
4	1.1018	0.9954	1.0091	1.0891	1.6109	1.7160	2.1053	3.9658
5	1.1248	0.9766	0.9840	1.0438	1.4826	1.5723	1.9040	3.4780
6	1.1421	0.9628	0.9657	1.0111	1.3927	1.4718	1.7646	3.1483
7	1.1555	0.9523	0.9519	0.9868	1.3275	1.3992	1.6647	2.9162
8	1.1659	0.9444	0.9414	0.9685	1.2793	1.3455	1.5913	2.7481
9	1.1740	0.9383	0.9334	0.9545	1.2429	1.3051	1.5363	2.6235
10	1.1803	0.9335	0.9271	0.9436	1.2151	1.2743	1.4945	2.5297

Table 5: Some numerical results of Λ_1, Λ_2, and Λ_3 in Example 1 with different values of q.

n	Λ_1	Λ_2	Λ_3	$\frac{1}{\Lambda_1+\Lambda_2+\Lambda_3}$
		$q=\frac{1}{3}$		
1	3.3364	1.1763	0.4559	0.2013
2	3.3955	1.1988	0.4592	0.1979
3	3.4147	1.2061	0.4602	0.1968
4	3.4219	1.2088	0.4606	0.1964
5	3.4240	1.2096	0.4608	0.1963
6	3.4244	1.2098	0.4608	0.1963
7	3.4246	1.2099	0.4608	0.1963
8	3.4251	1.2101	0.4608	0.1962
9	3.4251	1.2101	0.4608	0.1962
10	3.4251	1.2101	0.4608	0.1962
		$q=\frac{1}{2}$		
1	2.9820	1.0480	0.4467	0.2234
2	3.1379	1.1076	0.4552	0.2127
3	3.2160	1.1375	0.4593	0.2078
4	3.2553	1.1525	0.4613	0.2054
5	3.2754	1.1601	0.4623	0.2042
6	3.2852	1.1639	0.4628	0.2036
7	3.2898	1.1656	0.4630	0.2033
8	3.2923	1.1666	0.4631	0.2032
9	3.2939	1.1671	0.4632	0.2031
10	3.2943	1.1673	0.4632	0.2031
		$q=\frac{4}{5}$		
1	1.8371	0.6109	0.3881	0.3526
2	2.0805	0.7071	0.4077	0.3130
3	2.2800	0.7853	0.4216	0.2868
4	2.4430	0.8488	0.4319	0.2686
5	2.5751	0.9001	0.4395	0.2554
6	2.6823	0.9415	0.4454	0.2457
7	2.7686	0.9748	0.4499	0.2385
8	2.8380	1.0016	0.4534	0.2329
9	2.8943	1.0232	0.4562	0.2286
10	2.9392	1.0404	0.4584	0.2253

Algorithm 1 The proposed method for calculated $(a-b)_q^{(\alpha)}$

Input: a, b, α, n, q

$s \leftarrow 1$
if $n = 0$ **then**
  $p \leftarrow 1$
else
  for $k = 0$ to n **do**
    $s \leftarrow s * (a - b * a^k)/(a - b * q^{\alpha+k})$
  end for
  $p \leftarrow a^{\alpha} * s$
end if

Output: $(a-b)^{(\alpha)}$

Algorithm 2 The proposed method for calculated $\Gamma_q(x)$

Input: n, $q \in (0,1)$, $x \in \mathbb{R}\backslash\{0,-1,2,\dots\}$

$p \leftarrow 1$
for $k = 0$ to n **do**
  $p \leftarrow p(1 - q^{k+1})(1 - q^{x+k})$
end for
$\Gamma_q(x) \leftarrow p/(1-q)^{x-1}$

Output: $\Gamma_q(x)$

Algorithm 3 The proposed method for calculated $(\mathbb{D}_q f)(x)$

Input: $q \in (0,1)$, $f(x)$, x

syms z
if $x = 0$ **then**
  $g \leftarrow \lim((f(z) - f(q * z))/((1-q)z), z, 0)$
else
  $g \leftarrow (f(x) - f(q * x))/((1-q)x)$
end if

Output: $(\mathbb{D}_q f)(x)$

Algorithm 4 The proposed method for calculated $(\mathbb{I}_q^{\alpha} f)(x)$

Input: $q \in (0,1)$, α, n, $f(x)$, x

$s \leftarrow 0$
for $i = 0$ to n **do**
  $pf \leftarrow (1 - q^{i+1})^{\alpha-1}$
  $s \leftarrow s + pf * q^i * f(x * q^i)$
end for
$g \leftarrow (x^{\alpha} * (1-q) * s)/(\Gamma_q(x))$

Output: $(\mathbb{I}_q^{\alpha} f)(x)$

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 159–194
DOI:10.22067/ijnao.2021.11319.0

Research Article

Hopf bifurcation analysis in a delayed model of tumor therapy with oncolytic viruses

N. Akbari and R. Asheghi*

Abstract

The stability and Hopf bifurcation of a nonlinear mathematical model are described by the delay differential equation proposed by Wodarz for interaction between uninfected tumor cells and infected tumor cells with the virus. By choosing τ as a bifurcation parameter, we show that the Hopf bifurcation can occur for a critical value τ. Using the normal form theory and the center manifold theory, formulas are given to determine the stability and the direction of bifurcation and other properties of bifurcating periodic solutions. Then, by changing the infection rate to two nonlinear infection rates, we investigate the stability and existence of a limit cycle for the appropriate value of τ, numerically. Lastly, we present some numerical simulations to justify our theoretical results.

AMS subject classifications (2020): 92C50, 34C23, 92D25, 34K05.

Keywords: Hopf bifurcation; Delay model; Oncolytic viruses; Tumor cells.

1 Introduction

Cancer is a significant cause of death in the world. Thus, it is essential to discover some practical ways to prevail over it. Many studies have been made on cancer treatments, tumor cells behavior, clinical care, and so on. The primary purpose of cancer treatment is to reduce the destructive effects of cell behaviors [7]. The routine therapeutic substances for cancer are surgery,

*Corresponding author
Received 21 April 2020; revised 29 December 2020; accepted 19 January 2021
Najmeh Akbari
Department of Mathematical Sciences, Isfahan University of Techonology, Isfahan, Iran, 84156-83111. e-mail: najmeh.akbari@math.iut.ac.ir,

Rasoul Asheghi
Department of Mathematical Sciences, Isfahan University of Techonology, Isfahan, Iran, 84156-83111. e-mail: r.asheghi@iut.ac.ir

radiation, and chemotherapies. The first two treatments are excellent choices for tumor cells when they have metastatic behavior [16]. The conventional therapies are not only efficient but also highly toxic, so most efforts focus on establishing tumor cells targeted treatments. For this reason, the biological encountering is the most up to date practice [11]. An Oncolytic virus is a type of virus that infects a cell and then explodes it. In this process, the cell dies, and new virus particles are produced. The main problem with Oncolytic virus therapy is that the virus stays in the blood for a short time due to the secretion of the immune system [6, 15]. Oncolytic viruses that specifically target tumor cells are promising anti-cancer therapeutic agents. Due to our lack of understanding of the Oncolytic virus's dynamic spread in cancer cells, it is challenging to continuously control or eradicate cancer cells. The interaction between an Oncolytic virus and tumor cells, a type of virus-cell interaction, can be described by the mathematical models. The interplay between populations of uninfected tumor cells and infected tumor cells with the virus is complex and nonlinear. In this context, some mathematical models for the virus therapies of cancer cells can be seen as a tool for perceiving cancer-virus dynamics and finding better strategies for treatment; see [3, 4, 9].

In the proposed mathematical models, some parameters play a crucial role in the model's qualitative analysis. For example, an average and optimal rate of virus-infected cell death may optimize the treatment success, or the lower the number of uninfected cancer cells in the stable state can better predict the treatment process; see [11, 12]. Here, we recall a set of mathematical models that describe the virus's spread through the tumor cells in different ways [1]. First, we consider a general model as follows:

$$\begin{aligned}\frac{dx}{dt} &= xF(x,y) - \beta y G(x,y), \\ \frac{dy}{dt} &= \beta y G(x,y) - ay.\end{aligned} \tag{1}$$

This model consists of two populations: the population of uninfected tumor cells x and the infected tumor cells population by the virus y. The function $F(x,y)$ denotes the growth rate of noninfected cells, and the function $G(x,y)$ represents which the uninfected cells become infected with the virus. The coefficient β represents the infectivity of the virus. Virus-infected cells die with a rate of ay. Assuming that the growth of tumor cells approaches the carrying capacity and after a while, slows down and that the cell growth rate reaches zero, $F(x,y) = rx(1-(x+y))$ can be expressed by the logistic function. The $G(x,y)$ function plays a crucial role in determining the system's stability and helps us to make long-term predictions about treatment outcomes. Also, $G(x,y)$ can be divided into two different classes. In the class I, if the number of noninfected cells is greater than the number of infected cells and the tumor cells are not solid, then the virus replicates and increases the number of infected cells. Biologically, the growth of the virus is exponential and is called "fast virus spread." In class II, when the number of infected cells increases, the virus's growth rate decreases. The situation occurs in solid tumors because, as the number of infected cells increases, internal cells are surrounded by outer cells. Therefore, they cannot spread the virus. This type of infection is known as the "slow virus spread" [14].

We choose the following rates of infection given by [14]:

$$\begin{aligned} G_1(x,y) &= x, \\ G_2(x,y) &= \frac{(1+\varepsilon)x}{(x+y+\varepsilon)}, \\ G_3(x,y) &= \frac{x}{(xy^{\frac{1}{3}}+\varepsilon)}. \end{aligned}$$

Here G_1 and G_2 belong to class of "fast virus spread" and G_3 belongs to class of "slow virus spread". By replacing $F(x,y) = rx(1-\frac{x+y}{\omega})$ and $G = G_1$ in (1), we get

$$\begin{aligned} \frac{dx}{dt} &= rx(1-\frac{x+y}{\omega}) - \beta xy, \\ \frac{dy}{dt} &= \beta xy - ay. \end{aligned} \tag{2}$$

In 2016, the following model was developed by Wodarz, similar to (2) in which tumor cells death rate was considered (see [13]) as

$$\begin{aligned} \frac{dx}{dt} &= rx(1-\frac{x+y}{\omega}) - \delta x - \beta xy, \\ \frac{dy}{dt} &= \beta xy - ay, \end{aligned} \tag{3}$$

so that δ is defined the death rate for the uninfected tumor cells. Since (3) is an extension of (2), we introduce a delay time τ for entering virus in (3), such that it changes into

$$\begin{aligned} \frac{dx}{dt} &= rx(1-\frac{x+y}{\omega}) - \delta x - \beta xy, \\ \frac{dy}{dt} &= \beta x(t-\tau)y(t-\tau) - ay. \end{aligned} \tag{4}$$

Because of the simplicity of the delay calculations in system (4), first, we perform the calculations the delay equation of (4) analytically, and we find the appropriate parameter of τ to produce the limit cycle in (4). Now, by changing the linear infection rate of (4) to G_2 and G_3 and entering τ, we obtain, respectively,

$$\frac{dx}{dt} = rx(1 - \frac{x+y}{\omega}) - \delta x - \frac{\beta(1+\varepsilon)xy}{(x+y+\varepsilon)}, \qquad (5)$$
$$\frac{dy}{dt} = \frac{\beta(1+\varepsilon)x(t-\tau)y(t-\tau)}{(x(t-\tau)+y(t-\tau)+\varepsilon)} - ay,$$

and

$$\frac{dx}{dt} = rx(1 - \frac{x+y}{\omega}) - \delta x - \frac{\beta xy}{(xy^{\frac{1}{3}}+\varepsilon)}, \qquad (6)$$
$$\frac{dy}{dt} = \frac{\beta x(t-\tau)y(t-\tau)}{(x(t-\tau)y(t-\tau)^{\frac{1}{3}}+\varepsilon)} - ay.$$

By placing the parameter τ obtained from (4) and changing ε, we simulate the existence of a limit cycle in (5) and (6), numerically. The parameters r, ω, ε, β, and a are all positive and independent of the time. The state variables $x(t)$ and $y(t)$ are nonnegative. The parameter r is the reproduction rate of tumor cells, ω is the maximum carrying capacity of tumor cells, and the coefficient β denotes the virus's replication rate. Moreover, a is a death rate for the population y that are killed by the virus and ε is a positive parameter that is expressed only to fit the data to the model better.

In [10], the mathematical model

$$\frac{dx}{dt} = rx(1 - \frac{x+y}{K}) - bxy,$$
$$\frac{dy}{dt} = bx(t-\tau)y(t-\tau)e^{-n\tau} - ay,$$

for tumor virotherapy with the viral life-cycle is considered. In this model, the delay parameter displays the period of the viral life-cycle. The two parameters, b and a, are dominant factors in virotherapy. When $b < a$, the tumor cells reach their maximum size and the equilibrium solution $(1, 0)$, for $\tau \geq 0$, is stable. So, the therapy fails, but for $b > a$, the model's dynamic is much complicated, where the viral life-cycle comes to play an important role. As $b > a$, the viruses cannot exterminate the tumor cells without the viral life-cycle period. When the delay parameter value is small, the tumor cells and the viruses coexist, and their equilibrium solution is locally asymptotical stable. If the value of the delay parameter is longer than the period of viral life-cycle, the coexisting solution will be unstable, then the population of the tumor cells and the viruses will not rest on a fixed level. The model has a stable period solution for the value of the delay parameter in the middle range.

In appearance, by supposing $\delta = 0$, the model of (4) is similar to [10] for $n = 0$, but they are different in some ways. Firstly, in (4), there is the mortality rate of uninfected cells, which results in different reproduction rates with [10]. In this article, we investigate the stability of equilibrium points of (3) for different values of the reproduction rate by using the Lyapunov function. Then, with the Dulak theorem, we prove the absence of a limit cycle in (4) for $\tau = 0$. Another difference between this work and [10] is the use of the center manifold theory in the study of the stability, direction, and period of periodic solutions of bifurcation at (4) for critical values of τ, analytically.

In [2], another mathematical model

$$\frac{dx}{dt} = rx(1 - \frac{x+y}{k}) - dx - \beta xy,$$
$$\frac{dy}{dt} = \beta xy + sy(1 - \frac{x+y}{k}) - ay(t-\tau),$$

for tumor virotherapy is proposed. Wodarz in his article has stated that the infection rate and death rate of virus-infected cells play a key role in model dynamics. In that article, there is the time delay in death of infected cells and it entered in the linear sentence. Also, the Hopf bifurcation for linear infection rate is investigated. Our article's innovation is that the time delay in the arrival of the virus to tumor cells, and it entered in the nonlinear sentence. Also we have used three models with different Wodarz infection rates, two of them belong to the class of nonsolid tumors, and the other belongs to solid tumors. Using the obtained value τ suitable for the existence of a limit cycle in the initial model and its placement in the next two models, we have numerically shown that this dynamic depicts the general biological conditions of solid and nonsolid tumor cells and the effect of viral therapy on them.

In this article, we study the stability and Hopf bifurcation of system (4); then we compare it with (5) and (6). At first, in Theorem 1, we investigate the stability of the equilibrium points for system (4) for $\tau = 0$. Then in Lemma 2 and Theorem 3, we consider the existence of periodic solutions and the Hopf bifurcation for system (4) by inserting τ into the second equation (3) as a delayed time for importing Oncolytic viruses to the body and countering it with cancer cells. In Section 4, we describe the stability, direction, and period of the bifurcating periodic solutions at critical values of τ, by using the center manifold theory introduced in [5]. In Section 5, we substitute the appropriate value of τ obtained from (4) into (5) and (6). Then we simulate (5) and (6) for different values of ε. Finally, we discuss about them.

2 Stability of equilibrium points

In this section, we obtain equilibrium points of system (4) for $\tau = 0$ and study the conditions for the existence of positive equilibrium points. It is obvious that system (4) has equilibrium points $E_0 = (0, 0)$, $E_1 = \left(\frac{\omega(r-\delta)}{r}, 0\right)$, and $E_2 = \left(\frac{a}{\beta}, \frac{\beta\omega(r-\delta)-ra}{\beta(r+\beta\omega)}\right)$.
The Jacobian matrix of system (4) is denoted by

$$J(x,y) = \begin{pmatrix} (r-\delta) - \frac{2rx}{\omega} - (\frac{r}{\omega}+\beta)y & -(\frac{r}{\omega}+\beta)x \\ \beta y & \beta x - a \end{pmatrix}.$$

Theorem 1. Suppose $\mathcal{R}_0 = \frac{r}{\delta}$; then for $\tau = 0$, we have the following properties:

(a) If $\mathcal{R}_0 \leqslant 1$, then E_0 is a unique equilibrium point that is a stable node.

(b) If $1 - \frac{1}{\mathcal{R}_0} < \frac{a}{\beta\omega}$ and $\mathcal{R}_0 > 1$, then there exist two equilibrium points E_0 and E_1 such that E_0 is a saddle point and E_1 is a stable focus.

(c) If $1 - \frac{1}{\mathcal{R}_0} > \frac{a}{\beta\omega}$ and $\mathcal{R}_0 > 1$, then there exist three equilibrium points E_0, E_1, and E_2 such that E_0 and E_1 are saddle points and E_2 is a stable focus.

(d) If $1 - \frac{1}{\mathcal{R}_0} = \frac{a}{\beta\omega}$, then $E_1 = E_2$ and we have two equilibrium points such that E_0 is a saddle point and E_1 is locally asymptotically stable.

Proof.

(a) The Jacobian matrix of system (4) at the origin is

$$J(0,0) = \begin{pmatrix} r-\delta & 0 \\ 0 & -a \end{pmatrix}, \tag{7}$$

which has the eigenvalues $\lambda_1 = r-\delta < 0$ and $\lambda_1 = -a < 0$, when $\mathcal{R}_0 = \frac{r}{\delta} < 1$. This shows that E_0 is a stable node.

(b) When $\mathcal{R}_0 > 1$, for $\tau = 0$, there exists the equilibrium point E_1 and the Jacobian matrix of system (4) at E_1 is

$$J(E_1) = \begin{pmatrix} -r+\delta & -(r-\delta)(1+\frac{\beta\omega}{r}) \\ 0 & \frac{\beta\omega(r-\delta)-ra}{r} \end{pmatrix}. \tag{8}$$

It has the characteristic polynomial

$$\lambda^2 + m_1\lambda + m_2 = 0,$$

where

$$m_1 = a - \beta\omega + \beta\omega\frac{\delta}{r} + (r-\delta),$$

$$m_2 = (r-\delta)(a - \beta\omega + \beta\omega\frac{\delta}{r}).$$

For $a > \beta\omega - \beta\omega\frac{\delta}{r}$ or $1 - \frac{1}{\mathcal{R}_0} < \frac{a}{\beta\omega}$, we have that $m_k > 0$ $(k = 1,2)$ and this implies that E_1 is a stable focus.

(c) For $\mathcal{R}_0 > 1$ and $a < \beta\omega - \beta\omega\frac{\delta}{r}$, there exist two positive equilibrium points E_1 and E_2 such that the Jacobian matrix of system (4) at E_2 is

$$J(E_2) = \begin{pmatrix} -\frac{ra}{\beta\omega} & -a(\frac{r+\beta\omega}{\beta\omega}) \\ \frac{\beta\omega(r-\delta)-ra}{r+\beta\omega} & 0 \end{pmatrix}.$$

Since $tr(J(E_2)) = \frac{-ra}{\beta\omega} < 0$ and $det(J(E_2)) = \frac{a(\beta\omega(r-\delta)-ra)}{\beta\omega} > 0$, then E_2 is a stable focus.

(d) By substituting $1 - \frac{1}{\mathcal{R}_0} = \frac{a}{\beta\omega}$ in (8), we obtain $\lambda_1 = -(r-\delta) < 0$ and $\lambda_2 = 0$. Hence, E_1 is a nonhyperbolic fixed point with a zero eigenvalue. To show its stability, we use a suitable Lyapunove function. In the set $\Delta = \{(x(t), y(t)) | x(t) \geq 0, y(t) \geq 0\}$, we define a Lyapunov function

$$V(x,y) = x - \frac{\omega(r-\delta)}{r} - \frac{\omega(r-\delta)}{r} \ln \frac{rx}{\omega(r-\delta)} + \frac{\beta\omega + r}{\beta\omega} y,$$

such that $V(E_1) = 0$ and $V(x,y) > 0$ for all $(x,y) \in \Delta \setminus \{E_1\}$. We now show $\frac{dV}{dt} \leq 0$ for all $(x,y) \in \Delta$. By differentiating $V(x,y)$ along the solutions $(x(t), y(t))$ of (4), we have

$$\begin{aligned}
\frac{dV}{dt} &= \frac{\partial V}{\partial x}\frac{dx}{dt} + \frac{\partial V}{\partial y}\frac{dy}{dt} = \left(1 - \frac{\omega(r-\delta)}{rx}\right)\dot{x} + \frac{r+\beta\omega}{\beta\omega}\dot{y} \\
&= \left(1 - \frac{\omega(r-\delta)}{rx}\right)\left((r-\delta)x - \frac{r}{\omega}x^2 - (\frac{r+\beta\omega}{\omega})xy\right) + \left(\frac{r+\beta\omega}{\beta\omega}\right)(\beta xy - ay) \\
&= 2(r-\delta) - \frac{r}{\omega}x^2 - \frac{\omega(r-\delta)^2}{r} + (r+\beta\omega)\left(\frac{(r-\delta)}{r} - \frac{a}{\beta\omega}\right)y \\
&= -\frac{\omega}{r}\left(\frac{r^2}{\omega^2}x^2 - 2\frac{r}{\omega}(r-\delta)x + (r-\delta)^2\right) + \left(\frac{r+\beta\omega}{\beta\omega}\right)\left(\frac{\beta\omega(r-\delta)-ra}{r}\right)y \\
&= -\frac{\omega}{r}\left(\frac{r}{\omega}x - (r-\delta)\right)^2 + \left(\frac{r+\beta\omega}{\beta\omega}\right)\left(\frac{\beta\omega(r-\delta)-ra}{r}\right)y.
\end{aligned}$$

As $1 - \frac{1}{\mathcal{R}_0} = \frac{a}{\beta\omega}$, then $\frac{\beta\omega(r-\delta)-ra}{r} = 0$ and $\frac{dV}{dt} = -\frac{\omega}{r}\left(\frac{r}{\omega}x - (r-\delta)\right)^2 \leq 0$. If we set $\frac{dV}{dt} = 0$, then $x = \frac{\omega(r-\delta)}{r}$, and by substituting in system (4), we get $y = 0$. Therefore, E_1 is asymptotically stable. □

Theorem 2. System (3) in the set $\Delta = \{(x(t), y(t)) | \; x(t) \geq 0, y(t) \geq 0\}$ does not have any periodic solutions.

Proof. We define $F_1 = (r-\delta)x - \frac{r}{\omega}x^2 - (\frac{r+\beta\omega}{\omega})xy$ and $F_2 = \beta xy - ay$. By considering the Dulac function $L(x,y) = \frac{1}{xy}$, we have

$$\frac{\partial(LF_1)}{\partial x} + \frac{\partial(LF_2)}{\partial y} = -\frac{r}{\omega y} < 0.$$

Thus, by the Bendixson's criterion, system (3) has no closed orbits lying entirely in Δ. □

We have now shown that system (3) has no periodic solutions. We consider system (4) and show that the Hopf bifurcation exists under special conditions for $\tau > 0$. First, we suppose that $E^* = (x^*, y^*)$ is a positive equilibrium point for the system (4). The characteristic equation of any equilibrium point $E^* = (x^*, y^*)$ of system (4) for $\tau > 0$, is given by

$$D(\lambda, \tau) = det(\lambda I - Q_1 - Q_2 e^{-\lambda\tau}) = 0,$$

where

$$Q_1 = \begin{pmatrix} r - \delta - \frac{2rx}{\omega} - (\frac{r}{\omega} + \beta)y & -(\frac{r}{\omega} + \beta)x \\ 0 & -a \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 0 & 0 \\ \beta y & \beta x \end{pmatrix}.$$

Then

$$\begin{aligned} D(\lambda, \tau) &= det \begin{pmatrix} \lambda - r + \delta + \frac{2rx^*}{\omega} + (\frac{r}{\omega} + \beta)y^* & (\frac{r}{\omega} + \beta)x^* \\ -\beta y^* e^{-\lambda\tau} & \lambda - \beta x^* e^{-\lambda\tau} + a \end{pmatrix} \\ &= \lambda^2 + A_1\lambda + A_2 - (\lambda + B_1)B_2 e^{-\lambda\tau}, \end{aligned}$$

where

$$\begin{aligned} A_1 &= a - (r - \delta) + \frac{2rx^*}{\omega} + \frac{ry^*}{\omega} + \beta y^*, \\ A_2 &= a((r - \delta) + \frac{2rx^*}{\omega} + \frac{ry^*}{\omega} + \beta y^*), \\ B_1 &= (r - \delta) + \frac{2rx^*}{\omega}, \\ B_2 &= \beta x^* + a. \end{aligned}$$

Lemma 1. Let $x(\theta) = \phi_1(\theta) \geq 0$ and let $y(\theta) = \phi_2(\theta) \geq 0$, for $\theta \in [-\tau, 0]$, where $\phi = (\phi_1, \phi_2) \in C([-\tau, 0]; \mathbb{R}^2)$. Then the solution $(x(t), y(t))$ of system (4), defined on the interval $\Lambda = [0, T]$ for some $0 < T < \infty$, is positive and uniformly bounded on Λ.

Proof. The first equation of system (4) is a Bernouli equation, which can be written as

$$\dot{x}(t) - x(t)\left\{(r - \delta) - (\frac{r}{\omega} + \beta)y(t)\right\} = -\frac{r}{\omega}x^2(t).$$

Its solution with the initial data $x(0) = x_0$ is

$$x(t) = x_0 e^{\int_0^t \left\{(r-\delta)-(\frac{r}{\omega}+\beta)y(\gamma)\right\}d\gamma} \left\{x_0 \int_0^t \frac{r}{\omega} e^{\int_0^s \left\{(r-\delta)-(\frac{r}{\omega}+\beta)y(\gamma)\right\}d\gamma} ds + 1\right\}^{-1}.$$

Thus, $x(t) \geq 0$ when $x(0) = x_0 \geq 0$. We know $y(\theta) \geq 0$ for $\theta \in [-\tau, 0]$; then $y(t - \tau) \geq 0$ for $t \in [0, \tau]$. Thus, by the second equation of system (4), we have $y(t) \geq y(0)e^{-at} \geq 0$, for $t \in [0, \tau]$. In a similar way, we obtain $y(t) \geq 0$ for $t \in [\tau, 2\tau]$. By repeating this process, we conclude that $y(t)$ is nonnegative for all $t \geq 0$. To prove the boundedness property of the solutions, we will first show that $x(t)$ is bounded. By the first equation of system (4), we have

$$\dot{x}(t) \leq (r-\delta)x(t) - \frac{r}{\omega}x^2(t),$$

and hence

$$x(t) \leq \frac{\omega(r-\delta)}{r + \left(\frac{\omega(r-\delta)}{x_0} - r\right)e^{-(r-\delta)t}}.$$

Putting $\beta_0 = \frac{\omega(r-\delta)}{x_0} - r$, we have

$$x(t) \leq \frac{\omega(r-\delta)}{r + \beta_0 e^{-(r-\delta)t}},$$

and therefore $\limsup_{t\to\infty} x(t) \leq \frac{\omega(r-\delta)}{r}$. Now by supposing $\Omega(t) = x(t) + y(t+\tau)$ and $\gamma = r - \delta > 0$, we have $\limsup_{t\to\infty} x(t) \leq \frac{\omega\gamma}{r} =: \beta_1$ and

$$\begin{aligned}
\dot{\Omega}(t) &= \dot{x}(t) + \dot{y}(t+\tau)\\
&= \gamma x(t) - \frac{r}{\omega}x^2(t) - \frac{r}{\omega}x(t)y(t) - \beta x(t)y(t) + \beta x(t)y(t) - ay(t+\tau)\\
&= \gamma x(t) - \frac{r}{\omega}x^2(t) - \frac{r}{\omega}x(t)y(t) - ay(t+\tau) - ax(t) + ax(t)\\
&\leq \beta_1(\gamma + a) - \frac{r}{\omega}x^2(t) - \frac{r}{\omega}x(t)y(t) - ay(t+\tau) - ax(t)\\
&\leq \beta_1(\gamma + a) - a(y(t+\tau) + x(t))\\
&\leq \beta_1(\gamma + a) - a\Omega(t).
\end{aligned}$$

Hence, $\dot{\Omega}(t) + a\Omega(t) \leq \beta_1(\gamma + a)$, which gives $\Omega(t) \leq \frac{\beta_1(\gamma+a)}{a} - (\frac{\beta_1(\gamma+a)}{a} + \Omega(0))e^{-at}$. This implies that $\limsup_{t\to\infty} \Omega(t) \leq \frac{\beta_1(\gamma+a)}{a}$, for $r > \delta$. As a consequence, the functions $x(t)$ and $y(t)$ are uniformly bounded. □

3 Hopf bifurcation

We claim that for some $\tau > 0$, a Hopf bifurcation occurs. The interior equilibrium point E_2 exists when $\beta\omega(r-\delta) - ra > 0$. We move this point to the point $(1, 1)$ by setting

$$\begin{aligned}
x &= \frac{a}{\beta}X,\\
y &= \frac{\beta\omega(r-\delta) - ra}{\beta(r+\beta\omega)}Y, \qquad (9)\\
t &= T.
\end{aligned}$$

Applying the transformation (9) to (4), we get

$$\begin{aligned} \frac{dX}{dT} &= X(\gamma - mX) - nXY, \\ \frac{dY}{dT} &= -aY + aX(T-\tau)Y(T-\tau), \end{aligned} \tag{10}$$

where

$$\gamma = r - \delta, \qquad m = \frac{ra}{\omega\beta}, \qquad n = \frac{\beta\omega(r-\delta) - ra}{\beta\omega}. \tag{11}$$

Therefore, $(X, Y) = (1, 1)$ is an equilibrium point for system (10). Writing t instead of T and considering $u_1(t) = X(T) - 1$ and $u_2(t) = Y(T) - 1$ in system (10), we find that

$$\begin{aligned} \dot{u}_1(t) &= (u_1(t)+1)(\gamma - m(u_1(t)+1) - n(u_2(t)+1)), \\ \dot{u}_2(t) &= -a(u_2(t)+1) + a(u_1(t-\tau)+1)(u_2(t-\tau)+1). \end{aligned} \tag{12}$$

From (11), we know $\gamma - m - n = 0$, and by simplifying the right side of (12), the linear part of system (12) around $u = 0$ is given by

$$\begin{aligned} \dot{u}_1(t) &= -mu_1(t) - nu_2(t), \\ \dot{u}_2(t) &= -au_2(t) + au_1(t-\tau) + au_2(t-\tau). \end{aligned} \tag{13}$$

We know that the characteristic equation for system (13) is given by

$$D(\lambda, \tau) = det(\lambda I - Q_1 - Q_2 e^{-\lambda\tau}) = 0,$$

where

$$Q_1 = \begin{pmatrix} -m & -n \\ 0 & -a \end{pmatrix}, \qquad Q_2 = \begin{pmatrix} 0 & 0 \\ a & a \end{pmatrix}.$$

By the above assumptions, we can write

$$D(\lambda, \tau) = \lambda^2 + p_1\lambda + p_2 - (\lambda + q_1)q_2 e^{-\lambda\tau}, \tag{14}$$

in which

$$p_1 = m + a, \qquad p_2 = ma, \qquad q_1 = m - n, \qquad q_2 = a.$$

For $\tau > 0$, we show that under some conditions on parameters, a Hopf bifurcation happens. We suppose that $i\xi$ for some $\xi > 0$ is a root of (14) so that

$$
\begin{aligned}
D(i\xi,\tau) &= (i\xi)^2 + (i\xi)p_1 + p_2 - (i\xi + q_1)q_2 e^{-(i\xi)\tau} \\
&= -\xi^2 + i\xi p_1 + p_2 - (i\xi + q_1)q_2(\cos(\xi\tau) - i\sin(\xi\tau)) \\
&= -\xi^2 + i\xi p_1 + p_2 - i\xi q_2 \cos(\xi\tau) \\
&\quad - q_1 q_2 \cos(\xi\tau) - \xi q_2 \sin(\xi\tau) + i q_1 q_2 \sin(\xi\tau) \\
&= 0. \qquad (15)
\end{aligned}
$$

Separating the imaginary and real parts of (15) gives that

$$p_2 - \xi^2 = q_1 q_2 \cos(\xi\tau) + \xi q_2 \sin(\xi\tau), \qquad (16)$$

$$\xi p_1 = \xi q_2 \cos(\xi\tau) - q_1 q_2 \sin(\xi\tau). \qquad (17)$$

Eliminating $\sin(\xi\tau)$ and $\cos(\xi\tau)$ from (16) and (17) implies that

$$\xi^4 + A\xi^2 + B = 0, \qquad (18)$$

where

$$A = p_1^2 - 2p_2 - q_2^2, \qquad B = p_2^2 - q_1^2 q_2^2.$$

By assuming $Z = \xi^2$, equation (18) changes into

$$Z^2 + AZ + B = 0, \qquad (19)$$

which has the solutions

$$Z_{1,2} = \frac{-(p_1^2 - 2p_2 - q_2^2) \pm \sqrt{(p_1^2 - 2p_2 - q_2^2)^2 - 4(p_2^2 - q_1^2 q_2^2)}}{2}.$$

We suppose that $h(Z) = Z^2 + AZ + B$ has two positive roots Z_k, $k = 0, 1$ such that

$$0 < Z_0 < Z_1,$$

and

$$\xi_0 = \sqrt{Z_0}, \qquad \xi_1 = \sqrt{Z_1}.$$

Then we have the following result.

Lemma 2. Suppose that $h'(Z_k) \neq 0$ and $Z_k = \xi_k^2$. Then $\frac{dr}{d\tau}(\tau_k^{(j)}) \neq 0$ and its sign is given by the sign of $h'(Z_k)$. In particular, we have

$$h'(Z_0) < 0, \qquad h'(Z_1) > 0,$$

such that

$$\frac{dr(\tau_0^{(j)})}{d\tau} < 0, \qquad \frac{dr(\tau_1^{(j)})}{d\tau} > 0, \qquad j = 0, 1, 2, \ldots. \qquad (20)$$

Proof. Multiplying (16) by q_1 and (17) by ξ, we have

$$p_2q_1 - \xi^2 q_1 = q_1^2 q_2 \cos(\xi\tau) + \xi q_1 q_2 \sin(\xi\tau),$$
$$p_1\xi^2 = \xi^2 q_2 \cos(\xi\tau) - \xi q_1 q_2 \sin(\xi\tau),$$

and hence

$$\cos(\xi\tau) = \frac{p_2q_1 - \xi^2q_1 + p_1\xi^2}{q_1^2q_2 + \xi^2 q_2} = \frac{p_2q_1 - \xi^2q_1 + p_1\xi^2}{q_2(q_1^2 + \xi^2)}.$$

Consequently,

$$\tau_k^{(j)} = \frac{1}{\xi_k}\left[\arccos\left(\frac{p_2q_1 - \xi^2q_1 + p_1\xi^2}{q_2(q_1^2 + \xi^2)}\right)\right] + 2j\pi, \quad for \quad j \in \mathbb{Z}.$$

Therefore, $\pm i\xi_k$ for $k = 0, 1$ are two pairs of imaginary eigenvalues for $\tau = \tau_k^{(j)}$.
Let $\lambda(\tau) = r(\tau) + i\xi(\tau)$ be a root of (14). Then for $\tau = \tau_k^{(j)}$, we have

$$\lambda(\tau_k^{(j)}) = r(\tau_k^{(j)}) + i\xi(\tau_k^{(j)}) \qquad \textit{such that} \qquad r(\tau_k^{(j)}) = 0, \qquad \xi(\tau_k^{(j)}) = \xi_k > 0.$$

We substitute $\lambda(\tau)$ into (14), and we get

$$\lambda^2(\tau) + p_1\lambda(\tau) + p_2 - (\lambda(\tau) + q_1)q_2 e^{-\lambda(\tau)\tau} = 0.$$

Differentiating this equality with respect to τ leads to the identity

$$(2\lambda(\tau) + p_1)\frac{d\lambda(\tau)}{d\tau} - (1 - \tau(\lambda(\tau) + q_1))q_2 e^{-\lambda(\tau)\tau}\frac{d\lambda(\tau)}{d\tau} + \lambda(\tau)(\lambda(\tau) + q_1)q_2 e^{-\lambda(\tau)\tau} = 0.$$

This gives

$$\begin{aligned}\left(\frac{d\lambda(\tau)}{d\tau}\right)^{-1} &= \frac{1}{\lambda(\tau)(\lambda(\tau) + q_1)} - \frac{2\lambda(\tau) + p_1}{\lambda(\tau)(\lambda(\tau) + q_1)q_2 e^{-\lambda(\tau)\tau}} - \frac{\tau}{\lambda(\tau)} \\ &= \frac{\lambda(\tau)}{\lambda^2(\tau)(\lambda(\tau) + q_1)} - \frac{2\lambda^2(\tau) + p_1\lambda(\tau)}{\lambda^2(\tau)(\lambda(\tau) + q_1)q_2 e^{-\lambda(\tau)\tau}} - \frac{\tau}{\lambda(\tau)}.\end{aligned}$$

Now, by using (14), we obtain that

$$\left(\frac{d\lambda(\tau)}{d\tau}\right)^{-1} = \frac{-q_1}{\lambda^2(\tau)(\lambda(\tau) + q_1)} - \frac{\lambda^2(\tau) - p_2}{\lambda^2(\tau)(\lambda^2(\tau) + p_1\lambda(\tau) + p_2)} - \frac{\tau}{\lambda(\tau)},$$

and therefore,

$$\begin{aligned}
&\left(\frac{dRe\lambda(\tau)}{d\tau}\right)^{-1}\Big|_{\tau=\tau_k^{(j)}} \\
&\quad= \left(Re\left[\frac{-q_1}{\lambda^2(\tau)(\lambda(\tau)+q_1)}\right] + Re\left[\frac{p_2-\lambda^2(\tau)}{\lambda^2(\tau)(\lambda^2(\tau)+p_1\lambda(\tau)+p_2)}\right]\right. \\
&\qquad \left. + Re\left[\frac{-\tau}{\lambda(\tau)}\right]\right)\Big|_{\tau=\tau_k^{(j)}}.
\end{aligned}$$

Since, $\lambda(\tau_k^{(j)}) = r(\tau_k^{(j)}) + i\xi(\tau_k^{(j)}) = i\xi_k$, hence

$$\begin{aligned}
Re\left(\frac{-q_1}{\lambda^2(\tau)(\lambda(\tau)+q_1)}\right)\Big|_{\tau=\tau_k^{(j)}} &= Re\left[\frac{q_1(-i\xi_k+q_1)}{\xi_k^2(i\xi_k+q_1)(-i\xi_k+q_1)}\right] \\
&= \frac{q_1^2}{\xi_k^2(\xi_k^2+q_1^2)}, \\
Re\left(\frac{p_2-\lambda^2(\tau)}{\lambda^2(\tau)(\lambda^2(\tau)+p_1\lambda(\tau)+p_2)}\right)\Big|_{\tau=\tau_k^{(j)}} &= Re\left[\frac{p_2+\xi_k^2}{-\xi_k^2(-\xi_k^2+i\xi_k p_1+p_2)}\right] \\
&= \frac{(p_2^2-\xi_k^4)}{-\xi_k^2((-\xi_k^2+p_2)^2+p_1^2\xi_k^2)}, \\
Re\left(\frac{-\tau}{\lambda(\tau)}\right)\Big|_{\tau=\tau_k^{(j)}} &= Re\left[\frac{-\tau}{i\xi_k}\right] = 0.
\end{aligned}$$

By the above relations, we get

$$\left(\frac{dRe\lambda(\tau)}{d\tau}\right)^{-1}\Big|_{\tau=\tau_k^{(j)}} = \frac{q_1^2}{\xi_k^2(\xi_k^2+q_1^2)} - \frac{p_2^2-\xi_k^4}{\xi_k^2((-\xi_k^2+p_2)^2+\xi_k^2 p_1^2)}.$$

Let $\xi_k^2 = Z_k$. Then

$$\left(\frac{dRe\lambda(\tau)}{d\tau}\right)^{-1}\Big|_{\tau=\tau_k^{(j)}} = \frac{q_1^2}{Z_k(Z_k+q_1^2)} - \frac{p_2^2-Z_k^2}{Z_k((-Z_k+p_2)^2+Z_k p_1^2)}.$$

Since Z_k is a root of (19), we have

$$\begin{aligned}
&(-Z_k+p_2)^2 - Z_k p_1^2 = Z_k^2 + (p_1^2-2p_2)Z_k + p_2^2 = q_2^2 Z_k + q_1^2 q_2^2 = q_2^2(Z_k+q_1^2), \\
&Z_k^2 + (p_1^2-2p_2-q_2^2)Z_k = -p_2^2 + q_1^2 q_2^2.
\end{aligned}$$

Consequently,

$$\begin{aligned}
\left(\frac{dRe\lambda(\tau)}{d\tau}\right)^{-1}\Big|_{\tau=\tau_k^{(j)}} &= \frac{Z_k^2-p_2^2+q_1^2q_2^2}{Z_k q_2^2(Z_k+q_1^2)} = \frac{2Z_k^2+(p_1^2-2p_2-q_2^2)Z_k}{Z_k q_2^2(Z_k+q_1^2)} \\
&= \frac{2Z_k+p_1^2-2p_2-q_2^2}{q_2^2(Z_k+q_1^2)} = \frac{h'(Z_k)}{\Gamma_k},
\end{aligned}$$

where

$$\Gamma_k = q_2^2(Z_k + q_1^2) > 0,$$
$$h'(Z_k) = 2Z_k + p_1^2 - 2p_2 - q_2^2.$$

Therefore,

$$sign\left[\frac{dr(\tau)}{d\tau}\right]^{-1}_{\tau=\tau_k^{(j)}} = sign\left[\frac{dRe\lambda(\tau)}{d\tau}\right]^{-1}_{\tau=\tau_k^{(j)}} = sign\left[\frac{h'(Z_k)}{\Gamma_k}\right] = sign\left[h'(Z_k)\right].$$

□

Theorem 3. Let (19) have at least a positive root. Then system (4) has a Hopf bifurcation at $\tau = \tau_*^{(j)}$ and has one periodic solution surrounding E^*, where

$$\tau_*^{(j)} = \frac{1}{\sqrt{Z^*}}\left[\cos^{-1}\left(\frac{p_2q_1 - Z^*q_1 + p_1Z^*}{q_2(q_1^2 + Z^*)}\right)\right] + 2j\pi.$$

Proof. This result easily follows from Lemma 2. □

We recall the results stated in [8] on the roots of (14) for $\tau > 0$.

Lemma 3. ([8]) Consider the following assumptions:

(A1) $p_1 - q_2 > 0$;

(A2) $p_2 - q_1q_2 > 0$;

(A3) $p_1^2 - 2p_2 - q_2^2 > 0$ and $p_2^2 - q_1^2q_2^2 > 0$ or $(p_1^2 - 2p_2 - q_2^2)^2 < 4(p_2^2 - q_1^2q_2^2)$;

(A4) $p_1^2 - 2p_2 - q_2^2 < 0$ or $p_2^2 - q_1^2q_2^2 < 0$ and $(p_1^2 - 2p_2 - q_2^2)^2 = 4(p_2^2 - q_1^2q_2^2)$;

(A5) $p_1^2 - 2p_2 - q_2^2 < 0$, $p_2^2 - q_1^2q_2^2 > 0$ or $(p_1^2 - 2p_2 - q_2^2)^2 > 4(p_2^2 - q_1^2q_2^2)$.

Then the following properties hold:

(i) If $(A1) - (A3)$ are satisfied, then the real parts of all roots of (14) are negative for $\tau \geq 0$.

(ii) If $(A1)$, $(A2)$, and $(A4)$ are satisfied, then (14) at $\tau = \tau_k^{(j)}$, has a pair of imaginary roots $\pm i\xi_k$, such that the real parts of all roots except $\pm i\xi_k$ are negative.

(iii) If $(A1)$, $(A2)$, and $(A5)$ are satisfied, then (14) at $\tau = \tau_k^{(j)}$, has a pair of imaginary roots $\pm i\xi_k$, such that the real parts of all roots except $\pm i\xi_k$ are negative.

Theorem 4. If there are no positive roots for $h(Z) = Z^2 + AZ + B$ and $\mathcal{R}_0 > 1$, then E_2 is locally asymptotically stable for any $\tau \geq 0$.

Proof. Lemma 3 states that the real parts all of roots of (14) are negative, and this shows that E_2 is locally asymptotically stable for any $\tau \geq 0$. □

4 Stability of the Hopf bifurcation

In the previous section, we derived some conditions on Hopf bifurcation at the equilibrium point $E^* = (1,1)$, when $\tau = \tau_k^{(j)}, (j = 0, 1, 2 \dots)$. In this section, we obtain stability, direction, and period of periodic solution bifurcating from certain values of τ. We use the center manifold and normal form theory recognized in [5]. Let system (4) have a pair of imaginary roots when $\tau = \tau^*$ and let the system have a Hopf bifurcation from E^*. We set

$$u_1(t) = X(\tau t) - 1, \qquad u_2(t) = Y(\tau t) - 1, \qquad \tau = \tau^* + \nu, \qquad \nu \in \mathbb{R}.$$

Then, from (10), we get

$$\begin{aligned} \dot{u}_1(t) &= (\tau^* + \nu)(u_1(t) + 1)(-mu_1(t) - nu_2(t)), \\ \dot{u}_2(t) &= a(\tau^* + \nu)(-u_2(t) + u_1(t-1)u_2(t-1) + u_1(t-1) + u_2(t-1)). \end{aligned} \tag{21}$$

Equation (21) can be written in the form

$$\dot{u}(t) = L_\nu(u_t) + f(\nu, u_t), \qquad u(t) = (u_1(t), u_2(t)) \in \mathbb{R}^2.$$

Let $C = C([-1, 0]; \mathbb{R}^2)$ be the Banach space of continuous functions on $[-1, 0]$ with values in $\mathbb{R}^2$. Then for $\phi = (\phi_1, \phi_2) \in C$, we define

$$L_\nu : C \longrightarrow \mathbb{R}^2, \qquad f : \mathbb{R} \times C \longrightarrow \mathbb{R}^2,$$

through

$$\begin{aligned} L_\nu \phi &= (\tau^* + \nu) \begin{pmatrix} -m & -n \\ 0 & -a \end{pmatrix} \begin{pmatrix} \phi_1(0) \\ \phi_2(0) \end{pmatrix} + (\tau^* + \nu) \begin{pmatrix} 0 & 0 \\ a & a \end{pmatrix} \begin{pmatrix} \phi_1(-1) \\ \phi_2(-1) \end{pmatrix}, \\ f(\nu, \phi) &= (\tau^* + \nu) \left[-m \begin{pmatrix} \phi_1^2(0) \\ 0 \end{pmatrix} - n \begin{pmatrix} \phi_1(0)\phi_2(0) \\ 0 \end{pmatrix} + a \begin{pmatrix} 0 \\ \phi_1(-1)\phi_2(-1) \end{pmatrix} \right] \\ &= (\tau^* + \nu) \begin{pmatrix} -m\phi_1^2(0) - n\phi_1(0)\phi_2(0) \\ a\phi_1(-1)\phi_2(-1) \end{pmatrix}. \end{aligned} \tag{22}$$

By the Rise representation theorem, there exists a bounded variation function $\eta(\theta, \nu)$ for $\theta \in [-1, 0]$, such that

$$L_\nu(\phi) = \int_{-1}^{0} d\eta(\theta, \nu)\phi(\theta),$$

with

$$\eta(\theta, \nu) = (\tau^* + \nu) \begin{pmatrix} -m & -n \\ 0 & -a \end{pmatrix} \delta(\theta) - (\tau^* + \nu) \begin{pmatrix} 0 & 0 \\ a & a \end{pmatrix} \delta(\theta + 1),$$

where δ is Dirac delta function. For $\phi \in C^1([-1, 0]; \mathbb{R}^2)$, we define

$$A(\nu)\phi = \begin{cases} \frac{d\phi(\theta)}{d\theta}, & \theta \in [-1,0), \\ \int_{-1}^{0} d\eta(\theta,s)\phi(s), & \theta = 0, \end{cases} \tag{23}$$

and

$$R(\nu)\phi = \begin{cases} 0, & \theta \in [-1,0), \\ f(\nu,\phi), & \theta = 0. \end{cases} \tag{24}$$

By the above notations, system (21) can be written as

$$\dot{u}_t = A(\nu)u_t + R(\nu)u_t, \tag{25}$$

in which $u_t(\theta) = u(t+\theta)$ for $\theta \in [-1,0]$. For $\psi \in C^1([0,1];(\mathbb{R}^2)^*)$, define

$$(A^*\psi)(s) = \begin{cases} -\frac{d\psi(s)}{ds}, & s \in (0,1], \\ \int_{-1}^{0} d\eta(t,0)\psi(-t), & s = 0, \end{cases} \tag{26}$$

and denote the bilinear inner product by

$$< \psi(s), \phi(\theta) >= \overline{\psi(0)}\phi(0) - \int_{-1}^{0}\int_{\xi=0}^{\theta} \overline{\psi}(\xi-\theta)d\eta(\theta)\phi(\xi)d\xi. \tag{27}$$

We suppose that $\eta(\theta) = \eta(\theta,0)$ and that A^* is the adjoint operator of the linear map $A(0)$. From Section 3, we have that $\pm i\xi^*\tau^*$ are the eigenvalues of $A(0)$ and that also the eigenvalues of A^*. Assume that $q(\theta)$ is an eigenvector of $A(0)$ for $i\xi^*\tau^*$ and that $q^*(s)$ is an eigenvector of A^* for $-i\xi^*\tau^*$. Then by the definition of eigenvalue and using (23), we have

$$A(0)q(\theta) = i\xi^*\tau^*q(\theta) \qquad \Longrightarrow \qquad q'(\theta) = i\xi^*\tau^*q(\theta) \quad for \qquad \theta \in [-1,0),$$

and hence

$$\begin{aligned} q(\theta) &= q(0)e^{i\xi^*\tau^*\theta}, \qquad -1 \le \theta < 0, \\ q(-1) &= q(0)e^{-i\xi^*\tau^*}, \\ A_0q(0) + B_0q(-1) &= i\xi^*\tau^*q(0). \end{aligned}$$

Thus,

$$\begin{pmatrix} -m - i\xi^* & -n \\ ae^{-i\xi^*\tau^*} & -a - i\xi^* + ae^{-i\xi^*\tau^*} \end{pmatrix} \begin{pmatrix} q_1(0) \\ q_2(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{28}$$

It is easily seen that a nontrivial solution of (28) is as follows:

$$q_2(0) = \frac{-m - i\xi^*}{n}q_1(0), \qquad q_1(0) = 1 \neq 0,$$

so that

$$q(\theta) = (q_1(0), q_2(0))\, e^{i\xi^*\tau^*\theta} = \left(1, \frac{-m - i\xi^*}{n}\right) e^{i\xi^*\tau^*\theta}.$$

Similarly, we obtain $q^*(s) = E\left(q_1^*(0), q_2^*(0)\right)^T e^{i\xi^*\tau^* s}$ as an eigenvector of A^* corresponding to the eigenvalue $-i\xi^*\tau^*$. Indeed, by using (26), we have

$$A^* q^*(s) = -i\xi^*\tau^* q^*(s) \qquad \Longrightarrow \qquad -q^*\prime(s) = -i\xi^*\tau^* q(s) \quad for \qquad s \in (0,1],$$

and therefore

$$\begin{aligned} q^*(s) &= q^*(0)e^{i\xi^*\tau^* s}, \qquad 0 < s \leq 1, \\ q^*(1) &= q^*(0)e^{i\xi^*\tau^*}, \\ A_0^* q^*(0) + B_0^* q^*(1) &= -i\xi^*\theta^* q^*(0). \end{aligned}$$

Hence
$$\begin{pmatrix} -m + i\xi^* & ae^{i\xi^*\tau^*} \\ -n & -a + i\xi^* + ae^{i\xi^*\tau^*} \end{pmatrix} \begin{pmatrix} q_1^*(0) \\ q_2^*(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{29}$$

Equation (29) shows that

$$q_1^*(0) = \frac{i\xi^* - a + ae^{i\xi^*\tau^*}}{n} q_2^*(0), \qquad\qquad q_2^*(0) = 1 \neq 0.$$

Thus, we find that

$$q^*(s) = E\left(q_1^*(0), q_2^*(0)\right) e^{i\xi^*\tau^* s} = E\left(\frac{i\xi^* - a + ae^{i\xi^*\tau^*}}{n}, 1\right) e^{i\xi^*\tau^* s}.$$

For normalizing q and q^*, we assume that $< q^*(s), q(\theta) >= 1$ and by using (27), we get

$$\begin{aligned} < q^*(s), q(\theta) >=& \overline{E}(\overline{q^*}_1(0), 1)(1, q_2(0))^T \\ & - \int_{-1}^{0}\int_{\xi=0}^{\theta} \overline{E}(\overline{q^*}_1(0), 1)e^{-i\xi^*\tau^*(\xi-\theta)} d\eta(\theta)(1, q_2(0))d\xi \\ =& \overline{E}\Big\{ \left(\frac{-i\xi^* - a + ae^{-i\xi^*\tau^*}}{n}, 1\right)\left(1, \frac{-m - i\xi^*}{n}\right) \\ & - \int_{-1}^{0}\int_{\xi=0}^{\theta} \left(\frac{-i\xi^* - a + ae^{-i\xi^*\tau^*}}{n}, 1\right) e^{-i\xi^*\tau^*(\xi-\theta)} d\eta(\theta) \\ & \quad \times \left(1, \frac{-m - i\xi^*}{n}\right) e^{i\xi^*\tau^*\theta} d\xi \Big\} \\ =& \overline{E}\Big\{ \frac{-2i\xi^* - a + ae^{-i\xi^*\tau^*} - m}{n} \\ & - \int_{-1}^{0} \left(\frac{-i\xi^* - a + ae^{-i\xi^*\tau^*}}{n}, 1\right) \theta e^{i\xi^*\tau^*\theta} d\eta(\theta) \begin{pmatrix} 1 \\ \frac{-m - i\xi^*}{n} \end{pmatrix} \Big\} \end{aligned}$$

$$=\overline{E}\Big\{\frac{-2i\xi^*-a+ae^{-i\xi^*\tau^*}-m}{n}-\tau^*\frac{a(m-n)+ia\xi^*}{n}e^{-i\xi^*\tau^*}\Big\}$$
$$=1,$$

which gives

$$\overline{E}=\Big\{\frac{-2i\xi^*-a+ae^{-i\xi^*\tau^*}-m-\tau^*(a(m-n)+ia\xi^*)e^{-i\xi^*\tau^*}}{n}\Big\}^{-1}.$$

We now describe the solutions of (21) on the center manifold C_0 at $\nu=0$. Let $u(t)$ be the solution of (21) at $\nu=0$. Then

$$u(t)=u_0(t)+\Psi(u_0(t)),$$

for $u_0(t)\in E_0$ and $E_0=span\{q(\theta),q^*(\theta)\}$. Thus

$$u_0(t)=z(t)q(\theta)+\bar{z}(t)q^*(\theta)=z(t)q(\theta)+\bar{z}(t)\bar{q}(\theta),$$

and hence $u_t(\theta)$ on the center manifold can be written as

$$\begin{aligned}u_t(\theta)&=z(t)q(\theta)+\bar{z}(t)\bar{q}(\theta)+\Psi(z(t),\bar{z}(t))\\&=2Re\{z(t)q(\theta)\}+W(t,\theta),\end{aligned}\tag{30}$$

where

$$W(t,\theta)=u_t(\theta)-2Re\{z(t)q(\theta)\}.$$

It is clear that if $u_t(\theta)$ is real, then W is also real. Now, we set

$$W(t,\theta)=W(z,\bar{z},\theta)=W_{20}\frac{z^2}{2}+W_{11}z\bar{z}+W_{02}\frac{\bar{z}^2}{2}+W_{30}\frac{z^3}{6}+\cdots,\tag{31}$$

in which $z,\bar{z}$ are the coordinates of u_t with respect to q,q^*. The reduced equations up to order three for $(z,\bar{z})$ will be computed in the following. Equation (25) used with $\nu=\theta=0$ yields

$$\dot{u}(t)=A(0)u(t)+R(0)u(t),$$

which is, by (30), equivalent to

$$\dot{z}(t)q(0)+\dot{\bar{z}}(t)\bar{q}(0)+\dot{W}(t,0)=A(0)u(t)+R(0)u(t).\tag{32}$$

Since the inner product of the left-hand side of (32) with $q^*(0)$ is equal to $\dot{z}(t)$, then

$$\begin{aligned}
\dot{z}(t) &=< q^*(0), A(0)u(t) + R(0)u(t) > \\
&=< q^*(0), A(0)u(t) > + < q^*(0), R(0)u(t) > \\
&=< A^*(0)q^*(0), u(t) > + < q^*(0), R(0)u(t) > \\
&=< -i\xi^*\tau^* q^*(0), u(t) > + < q^*(0), R(0)u(t) > \\
&= i\xi^*\tau^* < q^*(0), u(t) > + < q^*(0), f(0, u(t)) > \\
&= i\xi^*\tau^* z(t) + \bar{q}^*(0) f(0, u(t)) \\
&= i\xi^*\tau^* z(t) + \bar{q}^*(0) f(0, z(t)q(0) + \bar{z}(t)\bar{q}(0) + W(t, 0)) \\
&= i\xi^*\tau^* z(t) + g(z(t), \bar{z}(t)),
\end{aligned}$$

where

$$g(z(t), \bar{z}(t)) = g_{20}\frac{z^2}{2} + g_{11}z\bar{z} + g_{02}\frac{\bar{z}^2}{2} + g_{21}\frac{z^2\bar{z}}{2} + \cdots . \tag{33}$$

To simplify the computations, we let

$$q_2(0) = \frac{-m - i\xi^*}{n} = M, \qquad q_1^*(0) = \frac{i\xi^* - a + ae^{i\xi^*\tau^*}}{n} = N.$$

Then (30) reads as

$$\begin{aligned}
u_t(\theta) &= z(t)q(\theta) + \bar{z}(t)\bar{q}(\theta) + W(t, \theta) \\
&= (1, M)^T e^{i\xi^*\tau^*\theta} z(t) + (1, \overline{M})^T e^{-i\xi^*\tau^*\theta} \bar{z}(t) \\
&\quad + W_{20}\frac{z^2}{2} + W_{11}z\bar{z} + W_{02}\frac{\bar{z}^2}{2} + \cdots .
\end{aligned} \tag{34}$$

It follows from (34) that

$$\begin{aligned}
u_1(t) &= z + \bar{z} + W^{(1)}(0), \\
u_2(t) &= Mz + \overline{M}\bar{z} + W^{(2)}(0), \\
u_1(t-1) &= ze^{-i\xi^*\tau^*} + \bar{z}e^{i\xi^*\tau^*} + W^{(1)}(-1), \\
u_2(t-1) &= Mze^{-i\xi^*\tau^*} + \overline{M}\bar{z}e^{i\xi^*\tau^*} + W^{(2)}(-1),
\end{aligned}$$

so that $W(t, \theta) = (W^{(1)}(\theta), W^{(2)}(\theta)) \in \mathbb{R}^2$. By replacing $\phi(\theta) = u_t(\theta)$ into (22) for $\nu = 0$ and the above relations, we have

$$
\begin{aligned}
&g(z,\bar{z})\\
&=\overline{q^*(0)}f(0,u_t)\\
&=\overline{q^*(0)}\tau^*\begin{pmatrix}-mu_1^2(t)-nu_1(t)u_2(t)\\ au_1(t-1)u_2(t-1)\end{pmatrix}\\
&=\overline{E}(\overline{N},1)\tau^*\begin{pmatrix}-m(z+\bar{z}+W^{(1)}(0))^2-n(z+\bar{z}+W^{(1)}(0))(Mz+\overline{M}\bar{z}+W^{(2)}(0))\\ a(ze^{-i\xi^*\tau^*}+\bar{z}e^{i\xi^*\tau^*}+W^{(1)}(-1))(Mze^{-i\xi^*\tau^*}+\overline{M}\bar{z}e^{i\xi^*\tau^*}+W^{(2)}(-1))\end{pmatrix}\\
&=-\overline{EN}\tau^*\{m(z+\bar{z}+W^{(1)}(0))^2+n(z+\bar{z}+W^{(1)}(0))(Mz+\overline{M}\bar{z}+W^{(2)}(0))\}\\
&\quad+\overline{E}\tau^*\{a(ze^{-i\xi^*\tau^*}+\bar{z}e^{i\xi^*\tau^*}+W^{(1)}(-1))(Mze^{-i\xi^*\tau^*}+\overline{M}\bar{z}e^{i\xi^*\tau^*}+W^{(2)}(-1))\}\\
&=-\overline{EN}\tau^*\Big\{2\,(m+nM)\,\frac{z^2}{2}+(2m+nRe\{M\})\,z\bar{z}+2\left(m+n\overline{M}\right)\frac{\bar{z}^2}{2}\\
&\quad+2\Big(2mW_{11}^{(1)}(0)+2mW_{20}^{(1)}(0)+nMW_{11}^{(1)}(0)+n\overline{M}W_{20}^{(1)}(0)+nW_{11}^{(2)}(0)\\
&\quad+nW_{20}^{(2)}(0)\Big)\frac{z^2\bar{z}}{2}\Big\}+\overline{E}\tau^*\Big\{2\left(aMe^{-2i\xi^*\tau^*}\right)\frac{z^2}{2}+(aRe\{M\})\,z\bar{z}+2\left(a\overline{M}e^{2i\xi^*\tau^*}\right)\frac{\bar{z}^2}{2}\\
&\quad+2\Big(ae^{-i\xi^*\tau^*}W_{11}^{(2)}(-1)+ae^{i\xi^*\tau^*}W_{20}^{(2)}(-1)+aMe^{-i\xi^*\tau^*}W_{11}^{(1)}(-1)\\
&\quad+a\overline{M}e^{i\xi^*\tau^*}W_{20}^{(1)}(-1)\Big)\frac{z^2\bar{z}}{2}\Big\}+\cdots.
\end{aligned}
\tag{35}
$$

We now replace (33) in the left-hand side of (35), and we get

$$
\begin{aligned}
g_{20}&=-2\overline{EN}\tau^*(m+nM)+2\overline{E}\tau^*(aMe^{-2i\xi^*\tau^*}),\\
g_{11}&=\overline{EN}\tau^*(2m+n\overline{M}+nM)+\overline{E}\tau^*(aM+a\overline{M}),\\
g_{02}&=-2\overline{EN}\tau^*(m+n\overline{M})+2\overline{E}\tau^*(a\overline{M}e^{2i\xi^*\tau^*}),\\
g_{21}&=-2\overline{EN}\tau^*\{(2m+nM)W_{11}^{(1)}(0)+(2m+n\overline{M})W_{20}^{(1)}(0)+nW_{11}^{(2)}(0)\\
&\quad+nW_{20}^{(2)}(0)\}+2\overline{E}\tau^*\{ae^{-i\xi^*\tau^*}W_{11}^{(2)}(-1)+ae^{i\xi^*\tau^*}W_{20}^{(2)}(-1)\\
&\quad+aMe^{-i\xi^*\tau^*}W_{11}^{(1)}(-1)+a\overline{M}e^{i\xi^*\tau^*}W_{20}^{(1)}(-1)\}.
\end{aligned}
\tag{36}
$$

As $W_{11}(\theta)$ and $W_{20}(\theta)$ are unknown, we should calculate them. To this end, we consider

$$\dot{W}=\dot{u}_t-q\dot{z}-\bar{q}\dot{\bar{z}}, \tag{37}$$

with

$$
\begin{aligned}
\dot{u}_t&=A(0)u_t+R(0)u_t=(A(0)+R(0))(qz+\bar{q}\bar{z}+W),\\
\dot{z}&=i\xi^*\tau^*z+\bar{q}^*(0)f_0,\\
\dot{\bar{z}}&=-i\xi^*\tau^*\bar{z}+q^*(0)\bar{f}_0.
\end{aligned}
$$

Then (37) becomes

$$
\begin{aligned}
\dot{W} =& A(0)(W + qz + \bar{q}\bar{z}) - (i\xi^*\tau^*z + \bar{q}^*(0)f_0)q - (-i\xi^*\tau^*\bar{z} + q^*(0)\bar{f}_0)\bar{q} \\
&+ R(0)u_t \\
=& A(0)W + A(0)qz + A(0)\bar{q}\bar{z} - (i\xi^*\tau^*z)q + (i\xi^*\tau^*\bar{z})\bar{q} \\
&- \bar{q}^*(0)f_0q - q^*(0)\bar{f}_0\bar{q} + R(0)u_t \\
=& A(0)W + (i\xi^*\tau^*q)z - (i\xi^*\tau^*\bar{q})\bar{z} - (i\xi^*\tau^*z)q + (i\xi^*\tau^*\bar{z})\bar{q} \\
&- \bar{q}^*(0)f_0q - q^*(0)\bar{f}_0\bar{q} + R(0)u_t \\
=& A(0)W - \bar{q}^*(0)f_0q + \overline{\bar{q}^*(0)f_0q} + R(0)u_t \\
=& A(0)W - 2Re\{\bar{q}^*(0)f_0q\} + R(0)u_t.
\end{aligned}
\tag{38}
$$

Next, from (24) and (38), we find that

$$
\begin{aligned}
\dot{W} &= \begin{cases} AW - 2Re\{\bar{q}^*(0)f_0q\} & \theta \in [-1, 0), \\ AW - 2Re\{\bar{q}^*(0)f_0q\} + f(0, u_t) & \theta = 0, \end{cases} \\
&= AW + H(z, \bar{z}, \theta),
\end{aligned}
\tag{39}
$$

where

$$
H(z, \bar{z}, \theta) = H_{20}(\theta)\frac{z^2}{2} + H_{11}(\theta)z\bar{z} + H_{02}(\theta)\frac{\bar{z}^2}{2} + \cdots . \tag{40}
$$

By replacing (31) and (40) into (39), we obtain that

$$
\begin{aligned}
&W_{20}(\theta)z\dot{z} + W_{11}(\theta)\dot{z}\bar{z} + W_{11}(\theta)z\dot{\bar{z}} + W_{02}(\theta)\bar{z}\dot{\bar{z}} + \cdots \\
&\quad = A(W_{20}(\theta)\frac{z^2}{2} + W_{11}(\theta)z\bar{z} + W_{02}(\theta)\frac{\bar{z}^2}{2}) + H_{20}(\theta)\frac{z^2}{2} \\
&\quad\quad + H_{11}(\theta)z\bar{z} + H_{02}(\theta)\frac{\bar{z}^2}{2} + \cdots .
\end{aligned}
$$

Also,

$$
\begin{aligned}
&W_{20}(\theta)z(i\xi^*\tau^*z + g(z, \bar{z})) + W_{11}(\theta)\bar{z}(i\xi^*\tau^*z + g(z, \bar{z})) \\
&\quad + W_{11}(\theta)z(-i\xi^*\tau^*\bar{z} + \overline{g(z, \bar{z})}) + W_{20}(\theta)\bar{z}(-i\xi^*\tau^*\bar{z} + \overline{g(z, \bar{z})}) \\
&\quad\quad = (AW_{20}(\theta) + H_{20}(\theta))\frac{z^2}{2} + (AW_{11}(\theta) + H_{11}(\theta))z\bar{z} \\
&\quad\quad\quad + (AW_{02}(\theta) + H_{02}(\theta))\frac{\bar{z}^2}{2} + \cdots .
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&(2i\xi^*\tau^*W_{20}(\theta))\frac{z^2}{2} + (-2i\xi^*\tau^*W_{20}(\theta))\frac{\bar{z}^2}{2} + \cdots \\
&= (AW_{20}(\theta) + H_{20}(\theta))\frac{z^2}{2} + (AW_{11}(\theta) + H_{11}(\theta))z\bar{z} + \cdots .
\end{aligned}
\tag{41}
$$

Comparing the similar terms in both sides of equality (41) leads to

$$AW_{20}(\theta) + H_{20}(\theta) = 2i\xi^*\tau^* W_{20}(\theta) \implies (A - 2i\xi^*\tau^*)W_{20}(\theta) = -H_{20}(\theta), \tag{42}$$

$$AW_{11}(\theta) + H_{11}(\theta) = 0 \qquad \implies \qquad AW_{11}(\theta) = -H_{11}(\theta). \tag{43}$$

From (39) for $\theta \in [-1, 0)$, we have that

$$\begin{aligned} H(z, \bar{z}, \theta) &= -2Re\{\bar{q}^*(0) f_0 q(\theta)\} \\ &= -\bar{q}^*(0) f_0 q(\theta) - \overline{\bar{q}^*(0) f_0 q(\theta)} \\ &= -g(z, \bar{z}) q(\theta) - \overline{g(z, \bar{z}) q(\theta)}. \end{aligned} \tag{44}$$

Then by comparing the coefficients of (33) and (40), we get

$$H_{20}(\theta) = -g_{20} q(\theta) - \bar{g}_{02} \bar{q}(\theta), \tag{45}$$

$$H_{11}(\theta) = -g_{11} q(\theta) - \bar{g}_{11} \bar{q}(\theta). \tag{46}$$

By substituting (45) into (42), we obtain

$$\begin{aligned} AW_{20}(\theta) &= 2i\xi^*\tau^* W_{20}(\theta) - H_{20}(\theta) \\ &= 2i\xi^*\tau^* W_{20}(\theta) + g_{20} q(\theta) + \bar{g}_{02} \bar{q}(\theta). \end{aligned} \tag{47}$$

By the definition of $A(\nu)$ for $\theta \in [-1, 0)$, we have $AW_{20}(\theta) = \dot{W}_{20}(\theta)$, and by replacing it into (47), we get

$$\begin{aligned} \dot{W}_{20}(\theta) &= 2i\xi^*\tau^* W_{20}(\theta) + g_{20} q(\theta) + \bar{g}_{02} \bar{q}(\theta) \\ &= 2i\xi^*\tau^* W_{20}(\theta) + g_{20} q(0) e^{i\xi^*\tau^*\theta} + \bar{g}_{02} \bar{q}(0) e^{-i\xi^*\tau^*\theta}, \end{aligned}$$

where $q(\theta) = q(0)e^{i\xi^*\tau^*\theta}$. Hence,

$$\begin{aligned} W_{20}(\theta) &= e^{2i\xi^*\tau^*\theta} \left\{ \int_0^\theta \left(g_{20} q(0) e^{i\xi^*\tau^*\theta} + \bar{g}_{02} \bar{q}(0) e^{-i\xi^*\tau^*\theta} \right) e^{-2i\xi^*\tau^*\theta} d\theta + C_1 \right\} \\ &= \frac{i g_{20}}{\xi^*\tau^*} q(0) e^{i\xi^*\tau^*\theta} + \frac{i\bar{g}_{02}}{3\xi^*\tau^*} \bar{q}(0) e^{-i\xi^*\tau^*\theta} + C_1 e^{2i\xi^*\tau^*\theta}. \end{aligned} \tag{48}$$

Similarly, it follows from (43), (46) and the definition of $A(\nu)$ that

$$\begin{aligned} &AW_{11}(\theta) = -H_{11}(\theta), \\ &\dot{W}_{11}(\theta) = g_{11} q(\theta) + \bar{g}_{11} \bar{q}(\theta) = g_{11} q(0) e^{i\xi^*\tau^*\theta} + \bar{g}_{11} \bar{q}(0) e^{-i\xi^*\tau^*\theta}. \end{aligned}$$

Then

$$
\begin{aligned}
W_{11}(\theta) &= \left\{\int_0^\theta \left(g_{11}q(0)e^{i\xi^*\tau^*\theta} + \bar{g}_{11}\bar{q}(0)e^{-i\xi^*\tau^*\theta}\right)d\theta + C_2\right\}\\
&= -\frac{ig_{11}}{\xi^*\tau^*}q(0)e^{i\xi^*\tau^*\theta} + \frac{i\bar{g}_{11}}{\xi^*\tau^*}\bar{q}(0)e^{-i\xi^*\tau^*\theta} + C_2. \qquad (49)
\end{aligned}
$$

We now need to find appropriate C_1 and C_2 for (48) and (49). From the definition of $A(\nu)$ for $\theta = 0$ and (42) we have that

$$
\int_{-1}^0 d\eta(\theta)W_{20}(\theta) = 2i\xi^*\tau^*W_{20}(0) - H_{20}(0), \qquad (50)
$$

where $\eta(0,\theta) = \eta(\theta)$. From the definition of $H(z,\bar{z},\theta)$ at (39) and (44) for $\theta = 0$, we have

$$
\begin{aligned}
H_{20}(0) &= -\bar{q}^*(0)f_0q(0) - q^*(0)\bar{f}_0\bar{q}(0) + f(0,u_t)\\
&= -g_{20}q(0) - \bar{g}_{02}\bar{q}(0) + \tau^*\begin{pmatrix} 2(m+nM) \\ -2aMe^{-2i\xi^*\tau^*}\end{pmatrix}. \qquad (51)
\end{aligned}
$$

Now, we substitute (48) and (51) into (50), and we get

$$
\begin{aligned}
&\tau^*\begin{pmatrix} -m & -n \\ 0 & -a\end{pmatrix}\left(\frac{ig_{20}}{\xi^*\tau^*}q(0) + \frac{i\bar{g}_{02}}{3\xi^*\tau^*}\bar{q}(0) + C_1\right)\\
&-\tau^*\begin{pmatrix} 0 & 0 \\ -a & -a\end{pmatrix}\left(\frac{ig_{20}}{\xi^*\tau^*}q(0)e^{-i\xi^*\tau^*} + \frac{i\bar{g}_{02}}{3\xi^*\tau^*}\bar{q}(0)e^{i\xi^*\tau^*} + C_1e^{-2i\xi^*\tau^*}\right)\\
&= 2i\xi^*\tau^*(\frac{ig_{20}}{\xi^*\tau^*}q(0) + \frac{i\bar{g}_{02}}{3\xi^*\tau^*}\bar{q}(0) + C_1) + g_{20}q(0) + \bar{g}_{02}\bar{q}(0)\\
&\quad + \tau^*\begin{pmatrix} 2(m+nM) \\ -2aMe^{-2i\xi^*\tau^*}\end{pmatrix}.
\end{aligned}
$$

It reduces to

$$
\begin{aligned}
&\tau^*\{\frac{ig_{20}}{\xi^*\tau^*}\begin{pmatrix} -m & -n \\ ae^{-i\xi^*\tau^*} & -a+ae^{-i\xi^*\tau^*}\end{pmatrix}q(0) + \frac{i\bar{g}_{02}}{3\xi^*\tau^*}\begin{pmatrix} -m & -n \\ ae^{i\xi^*\tau^*} & -a+ae^{i\xi^*\tau^*}\end{pmatrix}\bar{q}(0)\\
&+\begin{pmatrix} -m & -n \\ ae^{-2i\xi^*\tau^*} & -a+ae^{-2i\xi^*\tau^*}\end{pmatrix}C_1\}\\
&= -g_{20}q(0) - \bar{g}_{02}\bar{q}(0) + 2i\xi^*\tau^*C_1 + \tau^*\begin{pmatrix} 2(m+nM) \\ -2aMe^{-2i\xi^*\tau^*}\end{pmatrix}.
\end{aligned}
$$

Then we use the characteristic $(\lambda I - Q_1 - Q_2e^{-\lambda\tau})q(0) = 0$ and obtain

$$\tau^* \left\{ \frac{ig_{20}}{\xi^*\tau^*}(i\xi^* q(0)) + \frac{i\bar{g}_{02}}{3\xi^*\tau^*}(-i\xi^*\bar{q}(0)) + \begin{pmatrix} -m & -n \\ ae^{-2i\xi^*\tau^*} & -a + ae^{-2i\xi^*\tau^*} \end{pmatrix} C_1 \right\}$$
$$= -g_{20}q(0) - \bar{g}_{02}\bar{q}(0) + 2i\xi^*\tau^* C_1 + \tau^* \begin{pmatrix} 2(m+nM) \\ -2aMe^{-2i\xi^*\tau^*} \end{pmatrix}.$$

Therefore,

$$\begin{pmatrix} -m & -n \\ ae^{-2i\xi^*\tau^*} & -a + ae^{-2i\xi^*\tau^*} \end{pmatrix} C_1 = +2i\xi^* I C_1 + \begin{pmatrix} 2(m+nM) \\ -2aMe^{-2i\xi^*\tau^*} \end{pmatrix},$$

which is equivalent to

$$\begin{pmatrix} -m - 2i\xi^* & -n \\ ae^{-2i\xi^*\tau^*} & -a + ae^{-2i\xi^*\tau^*} - 2i\xi^* \end{pmatrix} \begin{pmatrix} C_1^{(1)} \\ C_1^{(2)} \end{pmatrix} = \begin{pmatrix} 2(m+nM) \\ -2aMe^{-2i\xi^*\tau^*} \end{pmatrix}.$$

It follows that

$$C_1^{(1)} = \frac{-2(ame^{-2i\xi^*\tau^*} - (a+2i\xi)(m+nM))}{ae^{-2i\xi^*\tau^*}(2i\xi + m - n) - 2i\xi(a + 2\xi + m) - am},$$
$$C_1^{(2)} = \frac{2a(m + nM - 2Mi\xi - mM)}{a(2i\xi + m - n) - (2i\xi(a + 2\xi + m) + am)e^{2i\xi^*\tau^*}}.$$

Similarly, from (43) and the definition of $A(\nu)$, we have

$$\int_{-1}^{0} d\eta(\theta) W_{11}(\theta) = -H_{11}(0), \tag{52}$$

and also from (39), we have

$$H_{11}(0) = -g_{11}q(0) - \bar{g}_{11}\bar{q}(0) + \tau^* \begin{pmatrix} 2m + nRe\{M\} \\ -aRe\{M\} \end{pmatrix}. \tag{53}$$

Now we substitute (49) and (53) into (52), and we get

$$\tau^* \begin{pmatrix} -m & -n \\ 0 & -a \end{pmatrix} (-\frac{ig_{11}}{\xi^*\tau^*}q(0) + \frac{i\bar{g}_{11}}{\xi^*\tau^*}\bar{q}(0) + C_2)$$
$$- \tau^* \begin{pmatrix} 0 & 0 \\ -a & -a \end{pmatrix} (-\frac{ig_{11}}{\xi^*\tau^*}q(0)e^{-i\xi^*\tau^*} + \frac{i\bar{g}_{11}}{\xi^*\tau^*}\bar{q}(0)e^{i\xi^*\tau^*} + C_2)$$
$$= g_{11}q(0) + \bar{g}_{11}\bar{q}(0) + \tau^* \begin{pmatrix} 2m + nRe\{M\} \\ -aRe\{M\} \end{pmatrix}.$$

Thus, we have

$$\begin{pmatrix} -m & -n \\ a & 0 \end{pmatrix} C_2 = \begin{pmatrix} 2m + nRe\{M\} \\ -aRe\{M\} \end{pmatrix},$$

which gives

$$C_2 = \begin{pmatrix} C_2^{(1)} \\ C_2^{(2)} \end{pmatrix} = \begin{pmatrix} -Re\{M\} \\ \frac{1}{n}((m-n)Re\{M\} - 2m) \end{pmatrix}.$$

By the above computations, we obtain coefficients of (36) and we can now determine the following quantities:

$$\begin{aligned} c_1(0) &= \frac{i}{2\xi^*\tau^*}\left(g_{11}g_{20} - 2|g_{11}|^2 - \frac{|g_{02}|^2}{3}\right) + \frac{g_{21}}{2}, \\ \mu_2 &= -\frac{Re(c_1(0))}{Re(\frac{d\lambda(\tau^*)}{d\tau})}, \\ \beta_2 &= 2Re(c_1(0)), \\ T_2 &= -\frac{Im(c_1(0) + \mu_2 Im(\frac{d\lambda(\tau^*)}{d\tau})}{\xi^*}. \end{aligned} \tag{54}$$

These values describe the bifurcating periodic solutions for the critical value $\tau = \tau^*$. The direction of the Hopf bifurcation is determined by μ_2, that is, when μ_2 is positive (negative), we have a supercritical (subcritical) Hopf bifurcation. For $\tau > \tau^*$ ($\tau < \tau^*$), there exist bifurcating periodic solutions, and β_2 determines the stability of the bifurcating periodic solutions: if $\beta_2 < 0$ ($\beta_2 > 0$), then bifurcating periodic solutions in the center manifold are stable (unstable). Period of the bifurcating periodic solutions is determined by T_2, the period increases (decrease) when $T_2 > 0$ ($T_2 < 0$). Thus (20) and (54) imply that the results of the direction the Hopf bifurcations satisfy.

5 Numerical Simulation

In this section, we study numerical results of system (4) at different values of τ. By choosing $r = 21,\ \omega = 21,\ \delta = 1,\ \beta = 3,\ a = 15$ in system (4), we have

$$\begin{aligned} \frac{dx}{dt} &= 21x(1 - \frac{x+y}{21}) - x - 3xy, \\ \frac{dy}{dt} &= 3x(t-\tau)y(t-\tau) - 15y. \end{aligned}$$

For $\tau = 0$, $E_2 = (5, 3.75)$ is an interior unique equilibrium point. These values show that (19) has a positive root $z = 129.6297769$, and then Theorem 3 shows that $\tau^* = 0.2648100601$. From the formulas in Section 4, we get

$$\begin{aligned}
M &= -0.3333333334 - 0.7241971227\,i,\\
N &= -0.1467688360 + 1.245730037\,i,\\
\overline{E} &= -0.3353048563 - 2.701455914\,i,\\
g_{20} &= 0.07001726116 + 0.0679380362\,i,\\
g_{11} &= 0.02285330684 - 0.1841225970\,i,\\
g_{02} &= 0.03920150800 - 0.9478826122\,i,\\
C_1^{(1)} &= 0.4358591088 + 0.5440217450\,i,\\
C_1^{(2)} &= -0.6386367604 - 0.1582198462\,i,\\
W_{20}^{(1)}(0) &= -0.2638607674 + 0.6954569963\,i,\\
W_{20}^{(2)}(0) &= -0.3302243769 - 0.5360810164\,i,\\
W_{20}^{(1)}(-1) &= 0.1400340404 - 0.2463831080\,i,\\
W_{20}^{(2)}(-1) &= 1.192302058 - 0.4408369488\,i,\\
C_2^{(1)} &= -0.3333333334,\\
C_2^{(2)} &= -0.4444444444,\\
W_{11}^{(1)}(0) &= -1.004521703 + 0.0\,i,\\
W_{11}^{(2)}(0) &= -0.1699352452 + 0.0\,i,\\
W_{11}^{(1)}(-1) &= -0.9494599988 + 0.0\,i,\\
W_{11}^{(2)}(-1) &= 0.0740685228 + 0.0\,i,\\
g_{21} &= -0.5523573046 + 1.0418494\,i.
\end{aligned}$$

By replacing the above values in (54), we obtain $c_1(0) = -0.2658449039 + 0.4681490320\,i$, which implies $\mu_2 = 49.95991907 > 0$, $\beta_2 = -0.5316898078 < 0$ and $T_2 = 0.04309591200 > 0$. The fact $T_2 > 0$ shows that the period of the bifurcating periodic solution increases, and finally, the periodic solution disappears. Now, by substituting the values of $r = 21,\ \omega = 21,\ \delta = 1,\ \beta = 3,\ a = 15$ and $\tau = 0.2648100601$ in system (5) and (6), we can simulate their dynamics by changing ε, numerically.

In Figure 1, we see that for $\tau = 0.05 < \tau^*$ and $\varepsilon = 100$, $E = (20, 0)$ in (6) and $E = (5, 3.75)$ in (4) and (5) are stable focus. Figure 2 shows that for $\tau = 0.2648100601$ and $\varepsilon = 0.001$, (5) and (6) in $E = (20, 0)$ have stable focus, but (4) has a periodic solution in $E = (5, 3.75)$. In Figure 3, with a slight increase in the amount of ε, no change in the dynamics of (5) and (6) is observed. In Figure 4, with increasing $\varepsilon = 10$, the stable focus of $E = (20, 0)$ in (6) remains unchanged. In contrast, the stable focus of (5) changes from $E = (20, 0)$ to $E = (12, 3.75)$, which is a sign of the effectiveness of viral therapy and reduction of tumor cells levels. In Figure 5, for $\varepsilon = 100$, around $E = (5, 3.75)$, a family of stable periodic solutions appears in (5) with period less than periodic solutions around $E = (5, 3.75)$ in (4), which indicates the coexistence between uninfected and infected cells, but $E = (20, 0)$ is a stable focus in (6). In Figures 6 and 7, we see that with increasing $\varepsilon = 1000$ and $\varepsilon = 1000000$, a family of stable periodic solutions with the same period will appear around $E = (5, 3.75)$, which shows at high values of ε, (4) and (5) have the same dynamic. While in (6), no change in equilibrium point and its dynamic is observed and indicate the ineffectiveness of viral therapy in reducing the number of infected tumor cells. In Figure 8, for $\varepsilon = 1000$ and $\tau = 1$, in (4) and (5), the periodic solution disappears. We have used Mathematica and Maple software for numerical simulating.

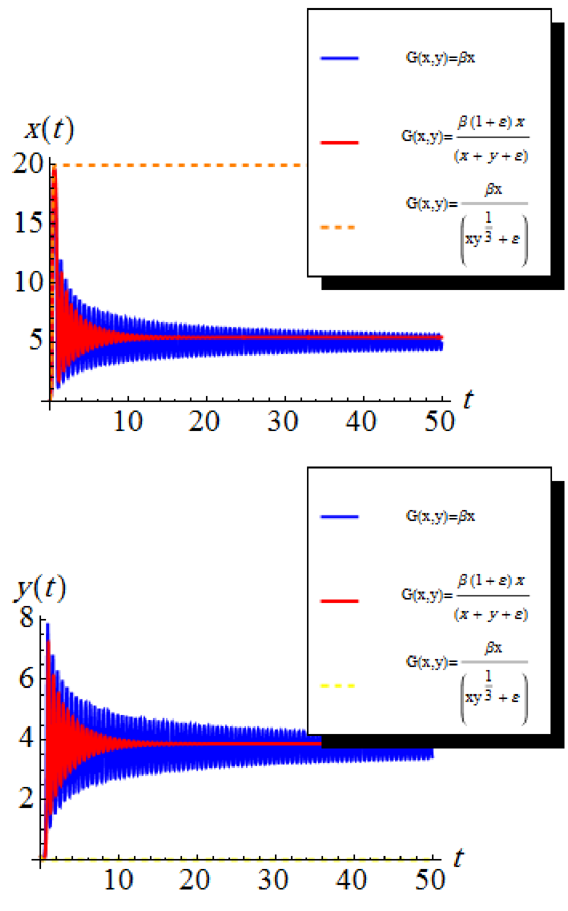


Figure 1: For $\varepsilon = 100$ and $\tau^* = 0.05$, $E = (20, 0)$ in (6), and $E = (5, 3.75)$ in (4) and (5) are stable focus.

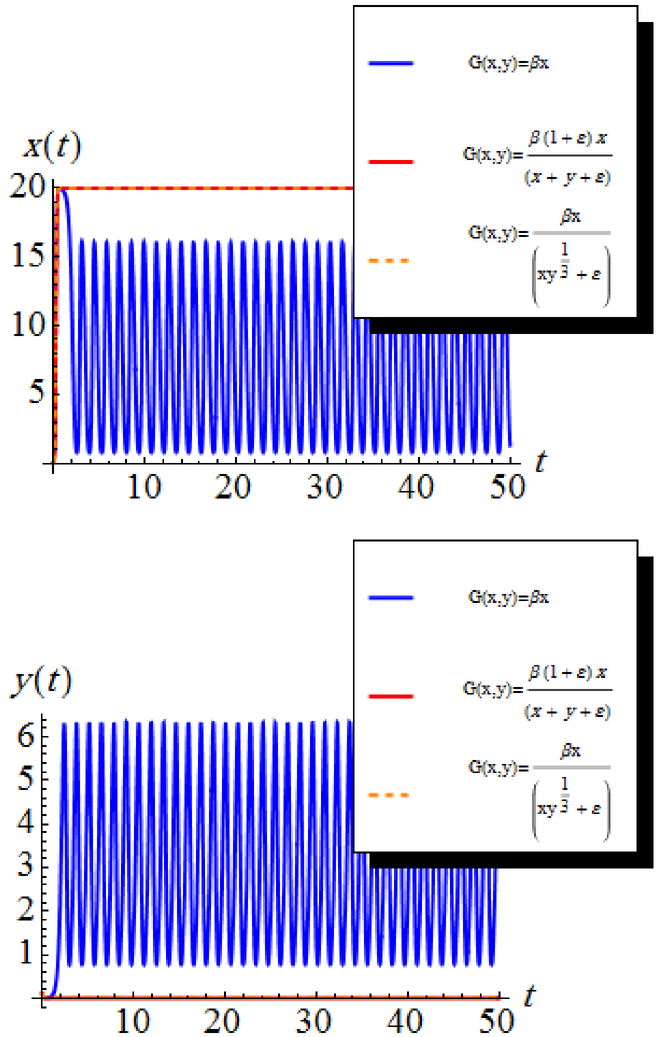
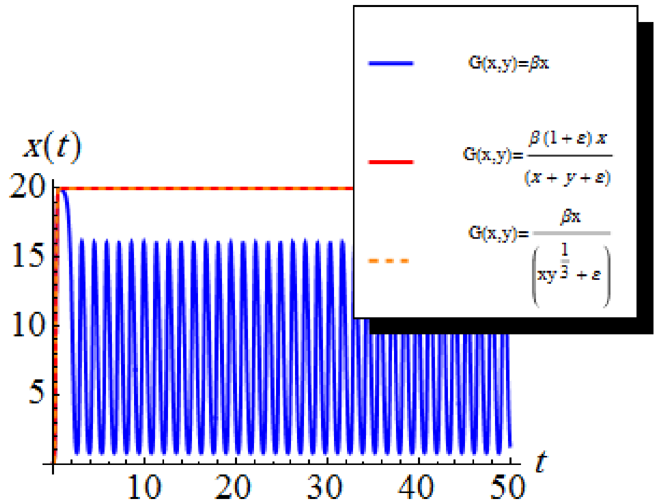


Figure 2: For $\varepsilon = 0.001$ and $\tau^* = 0.2648100601$ $E = (20, 0)$ in (5), and (6) is a stable focus and in (4) $E = (5, 3.75)$ enclosed by a periodic solution.



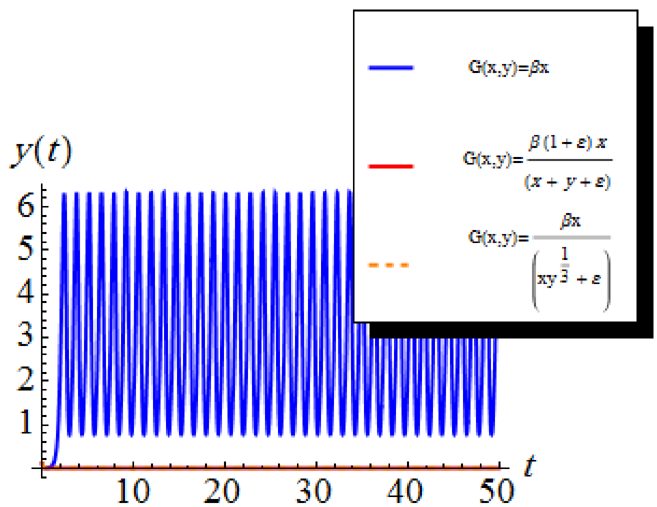


Figure 3: For $\varepsilon = 1$ and $\tau^* = 0.2648100601$ $E = (20, 0)$ in (5) and (6) is a stable focus and $E = (5, 3.75)$ in (4) enclosed by a periodic solution.

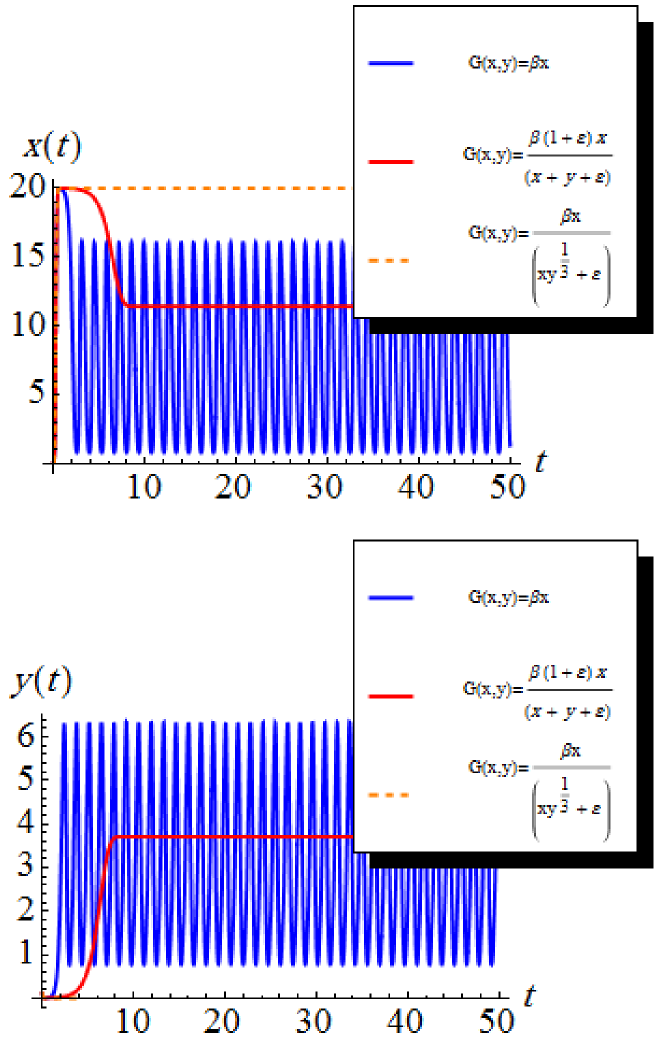


Figure 4: For $\varepsilon = 10$ and $\tau^* = 0.2648100601$ $E = (20, 0)$ in (6) is a stable focus, $E = (5, 3.75)$ in (4) enclosed by a periodic solution and $E = (12, 3.75)$ in (5) is a stable focus.

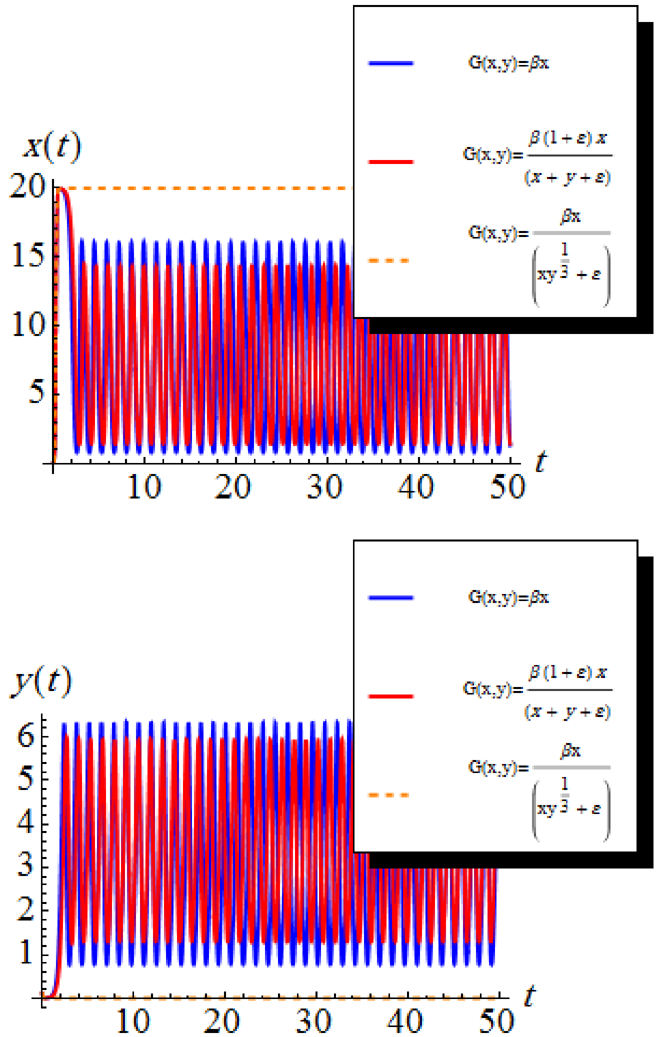
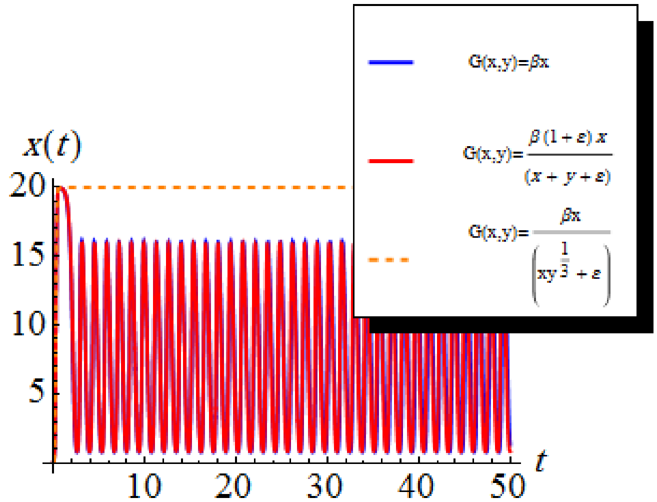


Figure 5: For $\varepsilon = 100$ and $\tau^* = 0.2648100601\, E = (20, 0)$ in (6) is a stable focus, $E = (5, 3.75)$ in (4) and in (5) enclosed by a periodic solution with different period of the bifurcating.



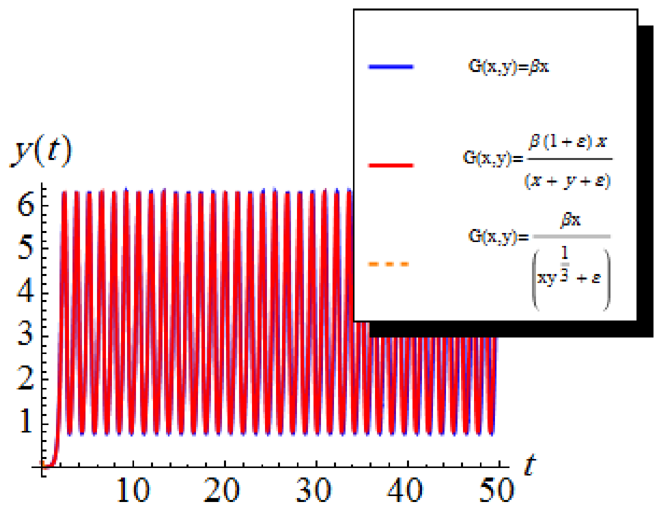


Figure 6: For $\varepsilon = 1000$ and $\tau^* = 0.2648100601\, E = (20, 0)$ in (6) is a stable focus, $E = (5, 3.75)$ in (4) and in (5) enclosed by a periodic solution with almost the same period.

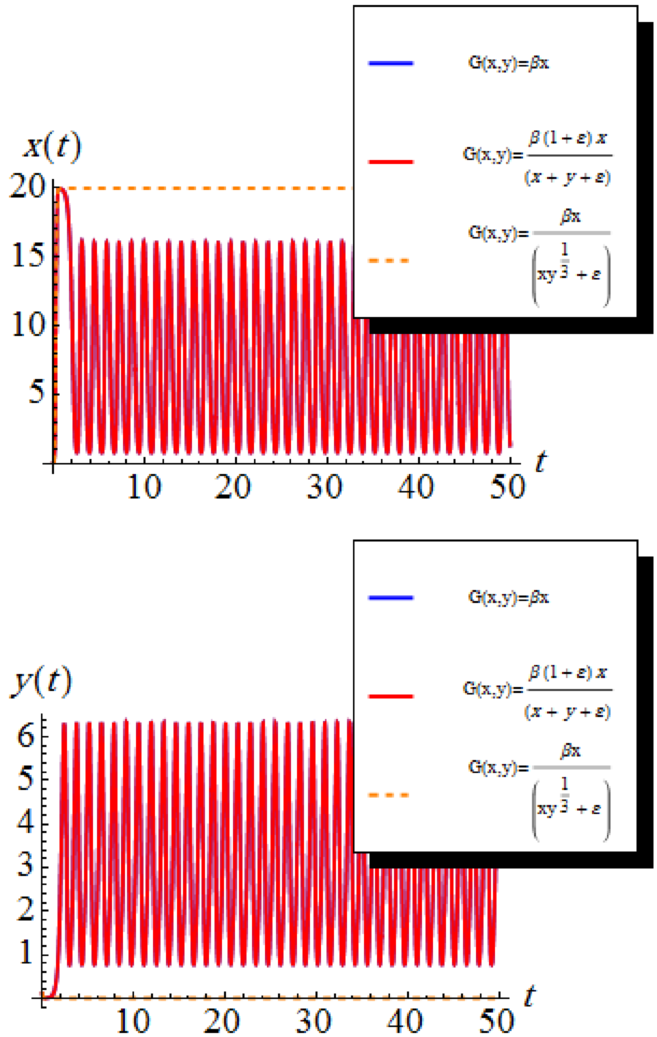


Figure 7: For $\varepsilon = 1000000$ and $\tau^* = 0.2648100601\, E = (20, 0)$ in (6) is a stable focus, $E = (5, 3.75)$ in (4) and (5) enclosed by the same periodic solution.

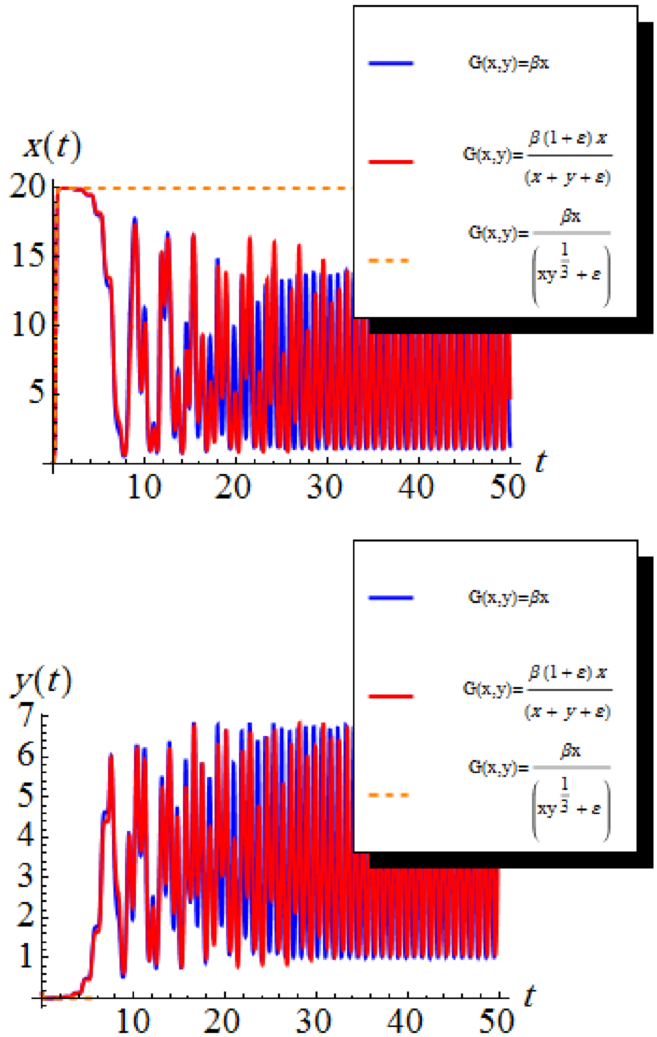


Figure 8: For $\varepsilon = 1000$ and $\tau = 1$ $E = (20, 0)$ in (6) is a stable focus, and in (4) and (5) the periodic solution disappears.

6 Discussion

Three nonlinear mathematical models were expressed with delayed differential equations. Two rates of infection, "rapid spread of the virus," and one rate of infection, "slow spread of the virus," are proposed. Due to the simplicity of calculating the mathematical model with a linear infection rate, the equilibrium points stability was studied without time delay. By introducing a time delay τ to the second equation in (4), we obtained a periodic solution and (4) undergoes a Hopf bifurcation. The existence of periodic solutions showed a coexistence between uninfected tumor cells and infected tumor cells, leading to a dynamical balance between in growth of uninfected cancer cells and infected ones. Now by placing the appropriate value τ obtained from (4) and different values of ε in (5) and (6), we simulated the existence of a periodic solution, numerically. In the simulation obtained from these three models, we found that in (4) and (5) with "rapid virus spread" rate, a periodic solution can be observed. From a biological point of view, it was stated that viral therapy in models with a "rapid virus spread" rate can create a coexistence interaction between uninfected and infected tumor cells. While simulation (6) showed that no limit cycles are observed for these values, viral treatment may not reduce tumor cells. By increasing the time delay, in (4) and (5), the period of the created periodic solutions increased until oscillations between two populations vanish. Hence, the growth of tumor cancer cells became stable.

References

1. Adam, J.A., and Bellomo, N., *A survey of models for tumor-immune system dynamics*, Birkhäuser Verlag AG, Basel-Berlin-Boston, 1996.

2. Ashyani, A., Mohammadinejad, H.M., and RabieiMotlagh, O., *Hopf bifurcation analysis in a delayed system for cancer virotherapy*, Indag. Math., 27(1) (2015), 318–339.

3. Chen, Y., and Su, Y.M., *An improved model of tumor therapy with the Oncolytic virus*, Journal of Henan University of Science and Technology, 37 (2016), 92–96.

4. Choudhury, B. S., and Nasipuri, B., *Efficient virotherapy of cancer in the presence of an immune response*, Int. J. Dyn. Control. 2 (2014), 314–325.

5. Hassard, B.D., Kazarinoff, N.D., and Wan, Y.H., *Theory and applications of Hopf bifurcation*, Cambridge University, Press, Cambridge, 1981.

6. Kim, P. H., Sohn, J. H., Choi, J. W., Jung, Y., Kim, S. W., Haam. S., and Yun, C. O., *Active targeting and safety profileof PEG-modified adenovirus conjugated with herceptin*, Biomaterials, 32(9) (2011), 2314–2326.

7. Mohammadian, H., and Asnafi, A., *Presentation of a model for virus therapy of cancer tumors using the modified prey-predator population dynamics*, Solids and Structures, 2 (2013), 17–22.

8. Ruan, S., *Absolute stability, conditional stability, and bifurcation in Kolmogrove-type predator-prey systems with discrete delays*, Quart. Appl. Math. 59 (2001), 159–173.

9. Si, W., and Zhang, W., *Control exponential growth of tumor cells with slow spread of the Oncolytic virus*, J. Theor. Biol. 367 (2015), 111–129.

10. Tian, J. P., Kuang, Y., and Yang, H., *Intracellular viral life-cycle induced rich dynamics in tumor virotherapy*, Canadian Journal of Applied Mathematics, (2016).

11. Wodarz, D., *Viruses as antitumor weapons: Defining conditions for tumor remission*, Cancer Res. 61 (2001), 3501–3507.

12. Wodarz, D., *Gene therapy for killing p53-Negative cancer cells: Use of replicating versus nonreplicating agents*, Human Gene Theory, 14 (2003), 153–159.

13. Wodarz, D., *Computational modeling approaches to the dynamics of Oncolytic viruses*, WIREs. Syst. Biol. Med. 8 (2016), 242–252.

14. Wodarz, D., and Komarova, N.,*Towards predictive computational models of Oncolytic virus therapy: basis for experimental validation and model selection,* PloS one, 4(1) (2009), 42–71.

15. Su, Y., Jia, C., and Chen, Y., *Optimal control model of tumor treatment with Oncolytic virus and MEK inhibitor*, Biomed Res Int. (2016), 1–8.

16. Zurakowski, R., and Wodarz, D., *Model-driven approaches for in vitro combination therapy using ONYX-015 replicating oncolytic adenovirus*, J. Theor. Biol. 245 (2007), 1–8.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 195–210
DOI:10.22067/ijnao.2021.11333.0

Research Article

ADI method of credit spread option pricing based on jump-diffusion model

R. Mohamadinejad*, A. Neisy and J. Biazar

Abstract

As the main contribution of this article, we establish an option on a credit spread under a stochastic interest rate. The intense volatilities in financial markets cause interest rates to change greatly; thus, we consider a jump term in addition to a diffusion term in our interest rate model. However, this decision leads us to a partial integral differential equation. Since the integral part might bring some difficulties, we put forward a fairly new numerical scheme based on the alternating direction implicit method. In the remainder of the article, we discuss consistency, stability, and convergence of the proposed approach. As the final step, with the help of the MATLAB program, we provide numerical results of implementing our method on the governing equation.

AMS subject classifications (2020): 45D05, 42C10, 65G99.

Keywords: Interest rate; Option pricing; Jump-diffusion models; Alternating direction implicit, Convergence.

1 Introduction

Spread options have great importance in financial markets. Modern and specialized investment in contemporary markets is based on this reality that we can represent strategies and models to invest in particular markets such

*Corresponding author
Received 21 October 2020; revised 20 January 2021; accepted 20 January 2021
Reyhane Mohamadinejad
Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran. e-mail: r.m.math81@gmail.com

Abdolsadeh Neisy
Department of Mathematics, Faculty of Mathematics Science and Computer, Allameh Tabataba'i University, Tehran, Iran; e-mail: a-neisy@atu.ac.ir

Jafar Biazar
Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran. e-mail: biazar@guilan.ac.ir

as the oil and exchange markets. These models are widely used in societies in which the inflation rate can change heavily during a short period, in short periods, and investors face less loss in these markets.

For example, we can consider an investment in the oil market. If the investor defines their call (buy) and put (sell) positions according to the oil assets and their derivatives, they do not face many volatility challenges. Because the asset price in the oil market changes over a period, the price of another asset will change equally in the same direction with a little increase or decrement. Another example of such markets, occurring in some countries, is intense volatilities of exchange. In this case, if investors investment to buy and sell currency (e.g., uses the difference between Dollar and Euro), they will not face intense price volatilities of this market from other markets.

The credit derivatives market has come under the spotlight recently. Credit derivatives pricing and valuation have received much attention over the last few years, and they are considered as one of the most significant new financial tools. The value of the portfolio with credit derivatives is extremely sensitive to changes in the spread between default-free bonds and defaultable bonds. Hence the investors can minimize and manage the exposure to credit risk. Credit default swaps and credit spread options are the most popular form of credit derivatives. Credit spread options payment depends on particular credit spread or credit-sensitive asset prices. The default or credit rating downgrade risks can be compensated by corporate bond credit spread; it is an additional remuneration paid to investors. Credit spread can protect the holders of corporate bonds from the loss of the rising yield spread. Deng, Johnson, and Sogomonian [1] studied the spread options in the energy market. In 2015, Zhou et al. [2] suggested a credit spread option pricing model that depends on the time, interest rate, and the logarithm of the credit spread, which was solved by GARCH and Longstaff–Schwartz methods. In 1995, Longstaff and Schwartz [3] studied that the credit spread logarithm follows the mean-reversion process and priced credit spread options on the assumption that the change in the distribution logarithm is normal. In 2012, Su and Wang [4] hypothesized that the interest rate would follow the Vasichek model and that the default intensity would follow the jump propagation trend. Tchuindjo [5], in 2011, proposed closed-form solutions for pricing credit risky discount bonds and their European call and put options in the intensity-based reduced-form framework, assuming the stochastic dynamics of both the risk-free interest rate and the credit spread are driven by two correlated Ho-Lee models.

These studies have not examined the abrupt changes in jumps and exchanges. In this study, a model investigates based on abrupt changes in the exchange. Also, the ADI numerical method with expected convergence is applied. The innovation of this article is in proving the sustainability of the method. Finally, implementing the model on the real data is one of the most significant tasks in this article. The ADI method is an appropriate method for solving large matrix equations and numerically solving parabolic and elliptic differential equations. This method is applied to prevent solving a large linear system. The advantages of this method are the optimal accuracy of stability, low computational volume, and low computer memory requirement.

This article is organized as follows: In section 2, the model, equation, and initial and boundary conditions will be introduced. In section 3, the ADI numerical method and its implementation on the model will be presented. Consistency, stability, and convergence of the method will be addressed in section 4. In Section 5, the proposed approach will be implemented on the real data. The last section provides some research titles for the followers.

2 Modeling of the credit spread option pricing

In this article, we consider the option pricing, according to the model presented by Longstaff and Schwartz [3]. That is if we take X_t as a credit spread option in a stock market, then its logarithm will satisfy the following stochastic differential equation [2]:

$$dX = (a - bX)dt + sdW_1,$$

also according to the Vasicek model, we have

$$dr = (\alpha - \beta r)dt + \sigma dW_2,$$

where the parameters a, b, s, α, β, and σ are constant numbers and W_1 and W_2 represent the Brownian motions and they are are independent variables with $corr(W_1, W_2) = \rho$. In mathematical modeling of financial topics, neglection of the jumps may lead to inaccurate results, by considering the economic and political problems, and also natural disasters such as earthquakes and storms as the main source of sudden movements, namely, jumps. So, in order to be able to present a pricing model for options, let us add the jump phrase to the Vasicek interest rate model. According to the information, we have

$$dX = (a - bX)dt + sdW_1,$$

$$dr = (\alpha - \beta r)dt + \sigma dW_2 + d(\sum_{i=1}^{M_t} Z_i),$$

where M_t is a Poisson process with the positive density rate λ that will be appeared soon, and the variables Z_i represent the jumps and are independent and identically distributed. The credit spread option price demonstrated by $P(X, r, t)$, equals to the solution of the following partial integral differential equation [6]:

$$P_t + (a - bX - qs)P_X + (\alpha - \beta r - q\sigma)P_r + \rho\sigma s P_{Xr} + \frac{1}{2}s^2 P_{XX} + \frac{1}{2}\sigma^2 P_{rr} - (r+\lambda)P + \int_0^{+\infty} P(X, r+y, t) f(y) dy = 0,$$

where $f(y)$ is the Poisson distribution probability density function and y is the jump value.
By applying the change of variable $t = T - \tau$, for $t \in [0, T]$, the last equation changes into the following form:

$$P_\tau = (a - bX - qs)P_X + (\alpha - \beta r - q\sigma + \frac{1}{\lambda})P_r + \rho\sigma s P_{Xr} + \frac{1}{2}s^2 P_{XX} + (\frac{1}{2}\sigma^2 + \frac{1}{\lambda^2})P_{rr} - (r + \lambda - 1)P. \tag{1}$$

The initial and boundary conditions for $0 \leq X \leq K, 0 \leq r \leq H$, and $0 \leq \tau \leq T$ are as follows:

$$\frac{\partial^2 P}{\partial X^2}(0, r, \tau) = 0, \frac{\partial^2 P}{\partial X^2}(K, r, \tau) = 0,$$

$$\frac{\partial^2 P}{\partial r^2}(X, 0, \tau) = 0, \frac{\partial^2 P}{\partial r^2}(X, H, \tau) = 0,$$

$$\frac{\partial P}{\partial X}(K, r, \tau) = 0, \frac{\partial P}{\partial r}(X, H, \tau) = 0,$$

$$P(0, r, \tau) = P(X, 0, \tau) = 1, P(X, r, 0) = G(x),$$

where $G(x) = \max\{X - \bar{K}, r - \bar{K}, 0\}$ is the payoff function of an European put option (in Figure 1, we display payoff function based on data in Table 1) and $\bar{K}$ is the strike price.

3 Numerical Solution

In order to solve (1), we apply the ADI method. To do this, we define the operator LP as follows:

$$LP = L^X P + L^r P + L^{Xr} P + L^{XX} P + L^{rr} P + \Phi.$$

We define the last-mentioned components corresponding to the ones in (1) (see [7, 8]),

$$L^X P = (a - bX - qs)P_X, \tag{2}$$

$$L^r P = (\alpha - \beta r - q\sigma)P_r, \tag{3}$$

$$L^{Xr} P = (\rho\sigma s)P_{Xr}, \tag{4}$$

$$L^{XX} P = \frac{1}{2}(s^2)P_{XX}, \tag{5}$$

$$L^{rr} P = (\frac{1}{2}(\sigma)^2 + \frac{1}{\lambda^2})P_{rr}, \tag{6}$$

$$\Phi = (r + \lambda - 1)P, \tag{7}$$

while q is the market price of the risk. Approximation of these derivatives by finite differences results in [9, 16] as follows:

$$\begin{aligned}(2):&\frac{P_{ij}^{n+\frac{1}{6}} - P_{ij}^{n}}{\Delta\tau} = (a - bx_i - qs)\frac{P_{i+1j}^{n+\frac{1}{6}} - P_{ij}^{n+\frac{1}{6}}}{h}\\ &\Rightarrow P_{ij}^{n+\frac{1}{6}} - P_{ij}^{n} = A(P_{i+1j}^{n+\frac{1}{6}} - P_{ij}^{n+\frac{1}{6}}),\end{aligned}$$

where $A = \dfrac{(a - bX - qs)(\Delta\tau)}{h}$. By rearranging the last equation, one has $(1 + A)P_{ij}^{n+\frac{1}{6}} - AP_{i+1j}^{n+\frac{1}{6}} = P_{ij}^{n}$, $i = 0, 1, 2, \ldots, n$, and the following system is obtained:

$$\begin{cases} i = 0: & (1 + A)P_{0j}^{n+\frac{1}{6}} - AP_{1j}^{n+\frac{1}{6}} = P_{0j}^{n},\\ i = 1: & (1 + A)P_{1j}^{n+\frac{1}{6}} - AP_{2j}^{n+\frac{1}{6}} = P_{1j}^{n},\\ i = 2: & (1 + A)P_{2j}^{n+\frac{1}{6}} - AP_{3j}^{n+\frac{1}{6}} = P_{2j}^{n},\\ i = n-1: & (1 + A)P_{n-1j}^{n+\frac{1}{6}} - AP_{nj}^{n+\frac{1}{6}} = P_{n-1j}^{n},\\ i = n: & (1 + A)P_{nj}^{n+\frac{1}{6}} - AP_{n+1j}^{n+\frac{1}{6}} = P_{nj}^{n}. \end{cases}$$

The second term of the left in the last equation should be determined by known values as follows:

$$\frac{P_{n+1j}^{n+\frac{1}{6}} - 2P_{nj}^{n+\frac{1}{6}} + P_{n-1j}^{n+\frac{1}{6}}}{h^2} = 0 \Rightarrow P_{n+1j}^{n+\frac{1}{6}} = 2P_{nj}^{n+\frac{1}{6}} - P_{n-1j}^{n+\frac{1}{6}}.$$

Substitution the recent formula in the last equation (for $i = n$) results

$$i = n: (1 - A)P_{nj}^{n+\frac{1}{6}} + P_{n-1j}^{n+\frac{1}{6}} = P_{nj}^{n+\frac{1}{6}}.$$

In the matrix representation, one gets

$$\begin{bmatrix} 1+A & -A & 0 & 0 & \cdots & 0 \\ 0 & 1+A & -A & 0 & \cdots & 0 \\ 0 & 0 & 1+A & -A & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1+A & -A \\ 0 & 0 & 0 & \cdots & A & 1-A \end{bmatrix} \begin{bmatrix} P_{0j}^{n+\frac{1}{6}} \\ P_{1j}^{n+\frac{1}{6}} \\ P_{2j}^{n+\frac{1}{6}} \\ \vdots \\ P_{n-1j}^{n+\frac{1}{6}} \\ P_{nj}^{n+\frac{1}{6}} \end{bmatrix} = \begin{bmatrix} P_{0j}^{n} \\ P_{1j}^{n} \\ P_{2j}^{n} \\ \vdots \\ P_{n-1j}^{n} \\ P_{nj}^{n} \end{bmatrix}$$

Now we apply the finite difference for (3) as follows:

$$\begin{aligned} \frac{P_{ij}^{n+\frac{1}{3}} - P_{ij}^{n+\frac{1}{6}}}{\Delta\tau} &= \frac{1}{2}(-^{r}P_{ij}^{n+\frac{1}{6}} + 3^{r}P_{ij}^{n+\frac{1}{3}}) \\ &= \frac{1}{2}(-B_j \frac{P_{ij+1}^{n+\frac{1}{6}} - P_{ij}^{n+\frac{1}{6}}}{k} + 3B_j \frac{P_{ij+1}^{n+\frac{1}{3}} - P_{ij}^{n+\frac{1}{3}}}{k}) \\ \Rightarrow (1+3B_j^{'})P_{ij}^{n+\frac{1}{3}} - 3B_j^{'}P_{ij+1}^{n+\frac{1}{3}} &= P_{ij}^{n+\frac{1}{6}} - B_j^{'}P_{ij+1}^{n+\frac{1}{6}} = P_{ij}^{n+\frac{1}{6}} - B_j^{'}P_{ij+1}^{n+\frac{1}{6}}, \end{aligned}$$

where $B_j^{'} = \frac{B_j\Delta\tau}{2k}$, $B_j = \alpha - \beta r_j - qs$, and $f_{ij} = P_{ij}^{n+\frac{1}{6}} - B_j^{'}P_{ij+1}^{n+\frac{1}{6}}$.

For $j = 0, 1, 2, \ldots, m$, we obtain the following system:

$$(1+3B_j^{'})P_{ij}^{n+\frac{1}{3}} - 3B_j^{'}P_{ij+1}^{n+\frac{1}{3}} = f_{ij}$$

$$\Rightarrow \begin{cases} j=0: & (1+3B_0^{'})P_{i0}^{n+\frac{1}{3}} - 3B_0^{'}P_{i1}^{n+\frac{1}{3}} = f_{i0}, \\ j=1: & (1+3B_1^{'})P_{i1}^{n+\frac{1}{3}} - 3B_1^{'}P_{i2}^{n+\frac{1}{3}} = f_{i1}, \\ j=2: & (1+3B_2^{'})P_{i2}^{n+\frac{1}{3}} - 3B_2^{'}P_{i3}^{n+\frac{1}{3}} = f_{i2}^{n}, \\ \vdots & \\ j=m-1: & (1+3B_{m-1}^{'})P_{im-1}^{n+\frac{1}{3}} - 3B_{m-1}^{'}P_{im}^{n+\frac{1}{3}} = f_{im-1}, \\ j=m: & (1+3B_m^{'})P_{im}^{n+\frac{1}{3}} - 3B_m^{'}P_{im+1}^{n+\frac{1}{3}} = f_{im}. \end{cases}$$

By applying the boundary conditions, the second term of the left in the last equation (for $j = m$) should be determined by known values:

$$P_{im+1}^{n+\frac{1}{3}} = 2P_{im}^{n+\frac{1}{3}} - P_{im-1}^{n+\frac{1}{3}}.$$

Substitution recent formula in the last equation (for $j = m$) results in:

$$j = m : (1+3B_m^{'})P_{im}^{n+\frac{1}{3}} - 3B_m^{'}(2P_{im}^{n+\frac{1}{3}} - P_{im-1}^{n+\frac{1}{3}}) = f_{im}$$

$$\Rightarrow (1-3B_m^{'})P_{im}^{n+\frac{1}{3}} + 3B_m^{'}P_{im-1}^{n+\frac{1}{3}} = f_{im}.$$

This system can be written as the following:

$$\begin{bmatrix} 1+3B_0^{'} & -3B_0^{'} & 0 & 0 & \cdots & 0 \\ 0 & 1+3B_1^{'} & -3B_1^{'} & 0 & \cdots & 0 \\ 0 & 0 & 1+3B_2^{'} & -3B_2^{'} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1+3B_{m-1}^{'} & -3B_{m-1}^{'} \\ 0 & 0 & 0 & \cdots & -3B_m^{'} & 1-3B_m^{'} \end{bmatrix} \begin{bmatrix} P_{i0}^{n+\frac{1}{3}} \\ P_{i1}^{n+\frac{1}{3}} \\ P_{i2}^{n+\frac{1}{3}} \\ \vdots \\ P_{n-1j}^{n+\frac{1}{3}} \\ P_{nj}^{n+\frac{1}{3}} \end{bmatrix} = \begin{bmatrix} f_{i0} \\ f_{i1} \\ f_{i2} \\ \vdots \\ f_{im-1} \\ f_{im} \end{bmatrix}$$

Using the finite difference method and the boundary conditions, (4) will transform into the following form:

$$\begin{aligned} &\frac{P_{ij}^{n+\frac{1}{2}} - P_{ij}^{n+\frac{1}{3}}}{\Delta\tau} = \frac{1}{12}(5^{Xr}P_{ij}^{n+\frac{1}{6}} - 16^{Xr}P_{ij}^{n+\frac{1}{3}} + 23^{Xr}P_{ij}^{n+\frac{1}{2}}) \\ &\Rightarrow (1-46C)P_{ij}^{n+\frac{1}{2}} - 23CP_{i+1j+1}^{n+\frac{1}{2}} + 23CP_{i+1j}^{n+\frac{1}{2}} + 23CP_{ij+1}^{n+\frac{1}{2}} + 23CP_{i-1j}^{n+\frac{1}{2}} \\ &\qquad + 23CP_{ij-1}^{n+\frac{1}{2}} - 23CP_{i-1j-1}^{n+\frac{1}{2}} = f_{ij}, \end{aligned}$$

in which f_{ij} are the sentences in $n+\frac{1}{3}$ and $n+\frac{1}{6}$ steps and $C = \frac{\rho\sigma s\Delta\tau}{12hk}$. Note that the finite difference in (4) follows the following formula:

$$P_{Xr} = \frac{P_{i+1j+1} - P_{i+1j}^{n+\frac{1}{2}} - P_{ij+1}^{n+\frac{1}{2}} + 2P_{ij}^{n+\frac{1}{2}} - P_{i-1j}^{n+\frac{1}{2}} - P_{ij-1}^{n+\frac{1}{2}} + P_{i-1j-1}^{n+\frac{1}{2}}}{hk}.$$

For $i = 0, 1, 2, \dots, n$, the following matrix equation is obtained:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots\cdots & \cdots & \cdots & 0 \\ 23C & 1-46C & 23C & 0 & \cdots\cdots & \cdots & \cdots & 0 \\ 0 & 23C & 1-46C & 23C & \cdots\cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots\ \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots\ 0 & 23C & 1-46C & 23C \\ 0 & 0 & 0 & 0 & \cdots\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{0j}^{n+\frac{1}{2}} \\ P_{1j}^{n+\frac{1}{2}} \\ P_{2j}^{n+\frac{1}{2}} \\ \vdots \\ P_{n-2j}^{n+\frac{1}{2}} \\ P_{n-1j}^{n+\frac{1}{2}} \\ P_{nj}^{n+\frac{1}{2}} \end{bmatrix}$$

$$+ \begin{bmatrix} -23C & 23C & \cdots & \cdots & \cdots & 0 \\ -23C & 23C & \cdots & \cdots & \cdots & 0 \\ 0 & -23C & 23C & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 23C & 0 \\ 0 & 0 & \cdots & \cdots & -23C & 23C \\ 0 & 0 & 0 & \cdots & -23C & 23C \end{bmatrix} \begin{bmatrix} P_{0j-1}^{n+\frac{1}{2}} \\ P_{1j-1}^{n+\frac{1}{2}} \\ \vdots \\ P_{n-2j-1}^{n+\frac{1}{2}} \\ P_{n-1j-1}^{n+\frac{1}{2}} \\ P_{nj-1}^{n+\frac{1}{2}} \end{bmatrix}$$

$$+\begin{bmatrix} 23C & -23C & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 23C & -23C & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 23C & -23C & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 23C & -23C & 0 \\ 0 & 0 & 0 & \cdots & 0 & 23C & -23C \\ 0 & 0 & 0 & \cdots & 0 & 23C & -23C \end{bmatrix} \begin{bmatrix} P_{0j+1}^{n+\frac{1}{2}} \\ P_{1j+1}^{n+\frac{1}{2}} \\ P_{2j+1}^{n+\frac{1}{2}} \\ \vdots \\ P_{n-2j+1}^{n+\frac{1}{2}} \\ P_{n-1j+1}^{n+\frac{1}{2}} \\ P_{nj+1}^{n+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} f_{0j} \\ f_{1j} \\ f_{2j} \\ \vdots \\ f_{n-2j} \\ f_{n-1j} \\ f_{nj} \end{bmatrix}$$

Applying the finite difference for (5), we get the following system of equations, $i = 0, 1, 2, \dots, n$:

$$\begin{aligned} &\frac{P_{ij}^{n+\frac{2}{3}} - P_{ij}^{n+\frac{1}{2}}}{\Delta\tau} \\ &= \frac{1}{24}(-9L^{XX}P_{ij}^{n+\frac{1}{6}} + 37L^{XX}P_{ij}^{n+\frac{1}{3}} - 59L^{XX}P_{ij}^{n+\frac{1}{2}} + 559L^{XX}P_{ij}^{n+\frac{2}{3}}) \\ &\Rightarrow \begin{cases} i = 0: & P_{0j}^{n+\frac{2}{3}} = f_{0j}, \\ j = 1: & (1+110D)P_{1j}^{n+\frac{2}{3}} - 55P_{0j}^{n+\frac{2}{3}} - 55P_{2j}^{n+\frac{2}{3}} = f_{1j}, \\ j = 2: & (1+110D)P_{2j}^{n+\frac{2}{3}} - 55P_{1j}^{n+\frac{2}{3}} - 55P_{3j}^{n+\frac{2}{3}} = f_{2j}, \\ \vdots & \\ j = n-2: & (1+110D)P_{n-2j}^{n+\frac{2}{3}} - 55P_{n-3j}^{n+\frac{2}{3}} - 55P_{n-1j}^{n+\frac{2}{3}} = f_{n-2j}, \\ j = n-1: & (1+110D)P_{n-1j}^{n+\frac{2}{3}} - 55P_{n-2j}^{n+\frac{2}{3}} - 55P_{nj}^{n+\frac{2}{3}} = f_{n-1j}, \\ j = n: & P_{nj}^{n+\frac{2}{3}} = f_{nj}, \end{cases} \end{aligned}$$

where f_{ij} are all the sentences in the steps $n+\frac{1}{2}$, $n+\frac{1}{3}$, $n+\frac{1}{6}$, and $D = \frac{s^2\Delta\tau}{48h^2}$.

The finite difference in (5) follows the following formula:

$$P_{XX} = \frac{P_{i+1j} - 2P_{ij} - P_{i-1j}}{h^2}.$$

Write the finite difference, regarding (6), and then, we obtain the following results for $j = 0, 1, 2, \dots, m$:

$$\begin{aligned} \frac{P_{ij}^{n+\frac{5}{6}} - P_{ij}^{n+\frac{2}{3}}}{\Delta\tau} =& \frac{1}{720}(251L^{rr}P_{ij}^{n+\frac{1}{6}} - 1274L^{rr}P_{ij}^{n+\frac{1}{3}} + 2616L^{rr}P_{ij}^{n+\frac{1}{2}} \\ &- 2774L^{rr}P_{ij}^{n+\frac{2}{3}} + 1901L^{rr}P_{ij}^{n+\frac{5}{6}}). \end{aligned}$$

The resulted matrix, by substitution $j = 0, 1, 2, \dots, m$ is as follows ($E = \frac{\Delta\tau}{720k^2}(\frac{1}{2}\sigma^2 + \frac{1}{\lambda^2})$):

$$
\begin{bmatrix}
1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\
-1901E & 1+3802E & -1901E & 0 & \cdots & \cdots & 0 \\
0 & -1901E & 1+3802E & -1901E & \cdots & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & -1901E & 1+3802E & -1901E & 0 \\
0 & 0 & \cdots & \cdots & -1901E & 1+3802E & -1901E \\
0 & 0 & \cdots & \cdots & 0 & 0 & 1
\end{bmatrix}
$$

$$
\times \begin{bmatrix} P_{i0}^{n+\frac{5}{6}} \\ P_{i1}^{n+\frac{5}{6}} \\ P_{i2}^{n+\frac{5}{6}} \\ \vdots \\ P_{im-2}^{n+\frac{5}{6}} \\ P_{im-1}^{n+\frac{5}{6}} \\ P_{im}^{n+\frac{5}{6}} \end{bmatrix} = \begin{bmatrix} f_{i0} \\ f_{i1} \\ f_{i2} \\ \vdots \\ f_{im-2} \\ f_{im-1} \\ f_{im} \end{bmatrix}
$$

The last step $(n+1)$ comes from (7) (see [6]) as follows:

$$
\frac{P_{ij}^{n+1} - P_{ij}^{n+\frac{5}{6}}}{\Delta\tau} = \varphi = -(r+\lambda-1)P_{ij}^{n}
$$
$$
\Rightarrow P_{ij}^{n+1} = -(r+\lambda-1)\Delta\tau P_{ij}^{n} + P_{ij}^{n+\frac{5}{6}}.
$$

4 Consistency, stability, and convergence

By proving the consistency and stability of the method, the convergence will be guaranteed via the Lax equivalence theorem [2].

4.1 Proof of consistency

Let Φ be the exact solution of (1), which depends on the independent variables r, X, and t (e.g., $L(\Phi) = 0$). Assume that the exact solution of the finite difference equation is ϕ (to write $F(\varphi) = 0$) and that v is an arbitrary continuous function of r, X, and t with a sufficient number of continuous derivatives such that $L(v)$ can be evaluated at $(i\eta, j\epsilon, kh)$. Then the error function at the point $v_{i,j,k} = v(i\eta, j\epsilon, hk)$ is defined as $\tau_{i,j,k}(v) = F(v_{i,j,k}) - L(v_{i,j,k})$. If $\tau_{i,j,k}(v) \to 0$ as η, ϵ , and h tend to zero, then the finite difference equation is consistent with the partial-differential equation as follows:

$$
\begin{aligned}
&\frac{\varphi_{i,j,k+1}-\varphi_{i,j,k}}{h}\\
&=(a-bX_i-qs)\frac{\varphi_{i+1,j,k}-\varphi_{i,j,k}}{\eta}+(\alpha-\beta r_j-q\sigma+\frac{1}{\lambda})\frac{\varphi_{i,j+1,k}-\varphi_{i,j,k}}{\epsilon}\\
&\quad+\rho\sigma s\frac{\varphi_{i+1,j+1,k}-\varphi_{i+1,j,k}-\varphi_{i,j+1,k}+2\varphi_{i,j,k}-\varphi_{i-1,j,k}-\varphi_{i,j-1,k}+\varphi_{i-1,j-1,k}}{\eta\epsilon}\\
&\quad+\frac{1}{2}s^2\frac{\varphi_{i+1,j,k}-2\varphi_{i,j,k}+\varphi_{i-1,j,k}}{\eta^2}\\
&\quad+(\frac{1}{2}\sigma^2+\frac{1}{\lambda^2})\frac{\varphi_{i,j+1,k}-2\varphi_{i,j,k}+\varphi_{i,j-1,k}}{\epsilon^2}-(r_j+\lambda-1)\varphi_{i,j,k}.
\end{aligned}
\tag{8}
$$

Let $X_i = X_0 + i\eta$, let $r_j = r_0 + j\epsilon$, let $A = a - bX_0 - qs$, $B = \alpha + \frac{1}{\lambda} - \beta r_0 - qs$, let $C = \rho\sigma s$, let $D = \frac{1}{2}s^2$, let $E = \frac{1}{2}\sigma^2 + \frac{1}{\lambda^2}$, and let $F = r_0 + \lambda - 1$. By considering the Taylor expansion and previous assumptions, (8) changes into the following equation:

$$
\begin{aligned}
\frac{h^2}{2}\frac{\partial^2\varphi}{\partial t^2}+\frac{h^2}{6}\frac{\partial^3\varphi}{\partial t^3}+\cdots =&(A-b\eta i)\frac{\eta}{2}\frac{\partial^2\varphi}{\partial X^2}+(B-\beta\epsilon j)\frac{\epsilon}{2}\frac{\partial^2\varphi}{\partial r^2}+\cdots\\
&+(A-b\eta i)\frac{\eta^2}{6}\frac{\partial^3\varphi}{\partial X^3}+(B-\beta\epsilon j)\frac{\epsilon^2}{3}\frac{\partial^3\varphi}{\partial r^3}+\cdots\ .
\end{aligned}
$$

By considering the smallest power of the above equation and assuming $\eta = \epsilon = \kappa h$ (where κ is a constant number), as h approaches zero, τ will approach to zero too and the consistency of this method is proved.

4.2 Proof of stability

Proving stability directly from the definition is quite difficult, in general. Instead, it is easier to use tools from the Fourier analysis to evaluate the stability of finite difference schemes. In this section, we analyze the stability of the ADI method. To study the stability, we use the Von Neumann approach [13, 14, 17]. The Von Neumann analysis is based on calculating the amplification factor of schemes, G, and deriving conditions under which $|G| \leq 1$. We have introduced the numerical amplification factor G as

$$
G = \frac{E_{k,l}^{n+1}}{E_{k,l}^{n}}. \tag{9}
$$

From (9), we may relate the error $E_{k,l}^{n}$ at the nth time step with the initial error $E_{k,l}^{0}$ [8] by

$$
E_{k,l}^{n} = G^n E_{k,l}^{0}, E_{k,l}^{0} = e^{ikx}e^{ily}.
$$

For finding G, a simple procedure is to replace $\varphi_{k,l}^{n}$ in the scheme by $G^n e^{ikx}e^{ily}$, for each k, l, and n,

$$\begin{aligned}\varphi_{k,l}^{n+1} &= G^{n+1} e^{ikx} e^{ily} = G\varphi_{k,l}^{n},\\ \varphi_{k+1,l}^{n} &= G^{n} e^{ik(x+\Delta x)} e^{ily} = G^{n} e^{ik\Delta x} e^{ikx} e^{ily} = \varphi_{k,l}^{n} e^{ik\Delta x},\\ \varphi_{k,l+1}^{n} &= G^{n} e^{ikx} e^{il(y+\Delta y)} = G^{n} e^{ikx} e^{ily} e^{il\Delta y} = \varphi_{k,l}^{n} e^{il\Delta y},\end{aligned} \tag{10}$$

where Δx and Δy are positive.
Without loss of generality of the problem, suppose that the coefficients of the relation (8) are assumed to be equivalent to constant A, where A is the largest value of the relation coefficients (8).
Substituting (10) in relation (8) leads to the following relation:

$$\begin{aligned}\frac{G-1}{\Delta t} =& A\Big(\frac{e^{ik\Delta x}-1}{\Delta x} + \frac{e^{il\Delta y}-1}{\Delta y} + \frac{e^{ik\Delta x}-2+e^{-ik\Delta x}}{(\Delta x)^2} + \frac{e^{il\Delta y}-2+e^{-il\Delta y}}{(\Delta y)^2}\\ &+ \frac{e^{i(k\Delta x+l\Delta y)} - e^{ik\Delta x} - e^{il\Delta y} + 2 - e^{-ik\Delta x} - e^{-il\Delta y} + e^{-i(k\Delta x+l\Delta y)}}{\Delta x\Delta y} + 1\Big)\end{aligned} \tag{11}$$

Suppose that $k\Delta x = l\Delta y = \theta$, that $\frac{\Delta t}{\Delta x} = \frac{\Delta t}{\Delta y} = h$, and that $\frac{\Delta t}{(\Delta x)^2} = \frac{\Delta t}{(\Delta y)^2} = \eta$. Then by simplifying relation (11), G is obtained as follows:

$$G = 1 + A(-2h + 2h\cos\theta - 2\eta(1 - cos2\theta) + \Delta t) + 2Ah\sin\theta i.$$

We calculate G^2 and establish $|G|^2 \leq 1$ to obtain stability conditions from it (Because in this article, Δx and Δy are considered values less than 1, we consider $h < \eta$ in the $|G|^2$ formula) as follows:

$$|G|^2 = (1 + A(-2\eta + 2\eta\cos\theta - 2\eta(1 - cos2\theta) + \Delta t))^2 + 4\eta^2 A^2 \sin^2\theta. \tag{12}$$

If the following conditions are met, relation (12) will be stable [15, 5]:

$$1 + A(-2\eta + 2\eta\cos\theta - 2\eta(1 - cos2\theta) + \Delta t) \leq 2A\eta\cos\theta,\ 4\eta^2 A^2 \leq 1. \tag{13}$$

The stability conditions of the problem, according to relations (13), are as follows:

$$\frac{\Delta t}{\Delta x^2} \leq \min\{\frac{1}{2|A|}, \frac{1}{A(2-\Delta x^2)}\},\ \frac{\Delta t}{\Delta y^2} \leq \min\{\frac{1}{2|A|}, \frac{1}{A(2-\Delta y^2)}\}.$$

Numerical results of this convergence and stability are given in the next section.

5 Numerical results

As an illustration, consider the credit spread and an interest rate to implement the ADI numerical method with $\frac{1}{6}$ step-size. Then, we obtain the changes in credit spread and an interest rate for these markets. Table 1 shows the parameters used in this study.

Table 1: Data

Parameters	definitions	values
K	$Max(X): maximum\ value\ of$ X	30
H	$Max(r): maximum\ value\ of$ r	30
q	$the\ market\ price\ of\ the\ risk$	1
σ	$volatility$	0.3
T	$Max(t):\ maximum\ value\ of$ t	0.5
a, b, and s	$constant$	$0.5, 0.5, and\ 0.25$
λ	$intensity\ rate\ of\ Poisson\ process$	0.1
ρ	$corr(W_1, W_2)$	0.5
r	$interest\ rate$	0.03

The payoff function can be illustrated in Figure 1

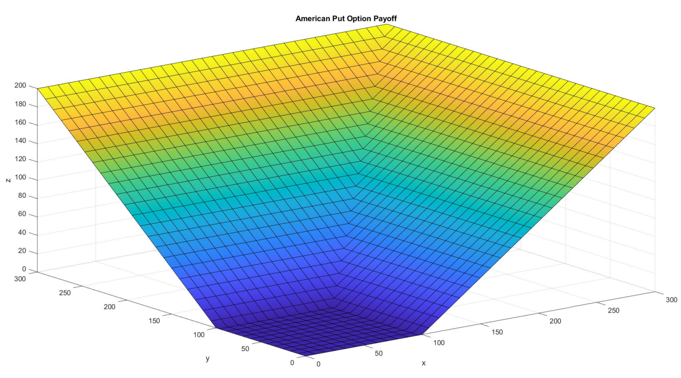


Figure 1: Payoff function.

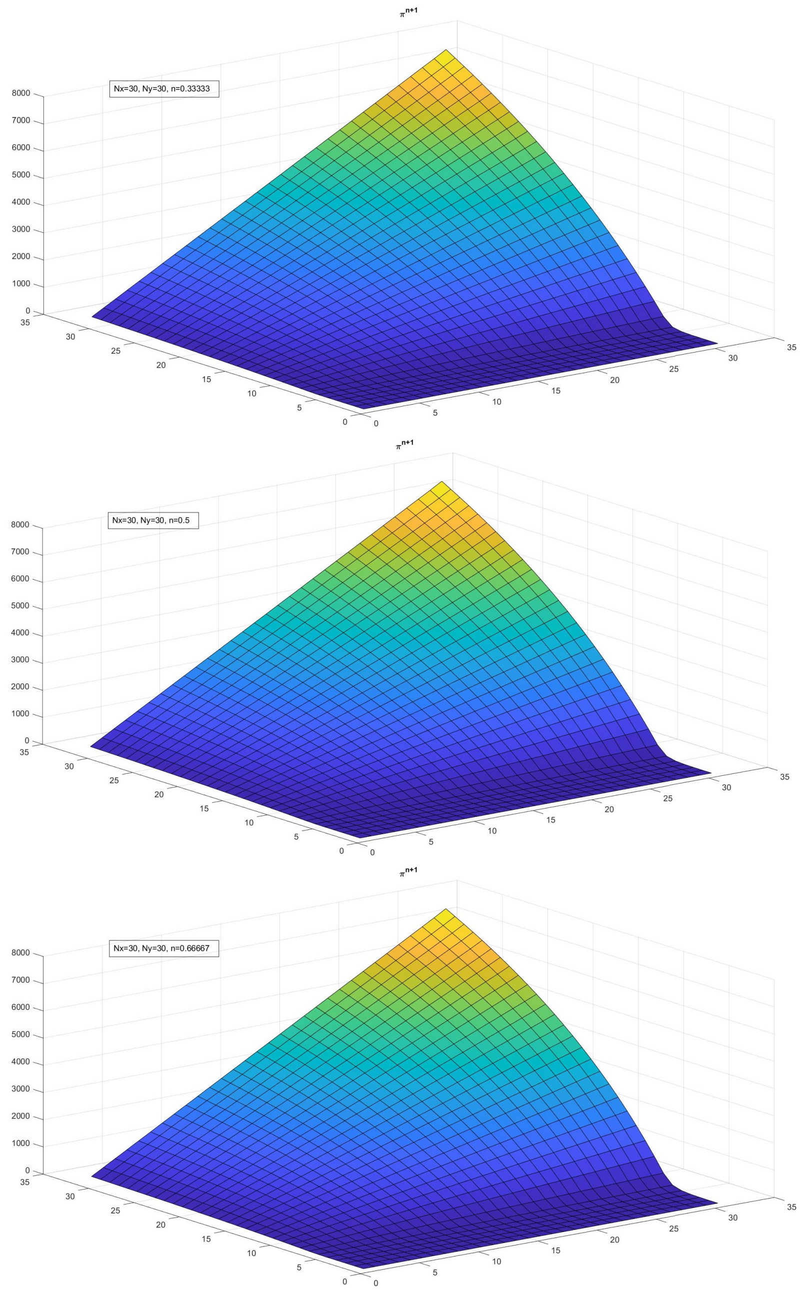


Figure 2: Numerical results using the ADI method with a step-size of $\frac{1}{6}$ for $n = 0.3333, 0.5,$ and 0.66667

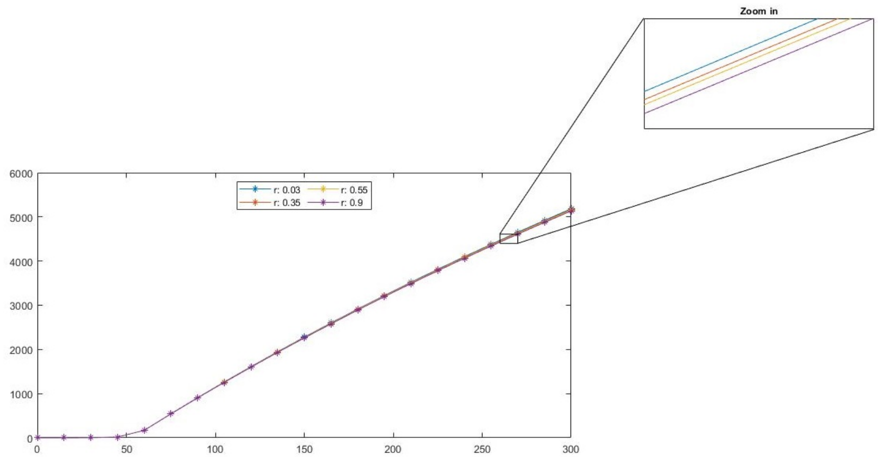


Figure 3: Comparison of option prices for different interest rates such as $r = 0.03, 0.35, 0.55,$ and 0.9

Eventually, regarding our model and the suggested numerical method, we can obtain the credit spread option price. As can be seen from the following figures, the price of the credit spread option varies for different amounts of the variable n (Figure 2 shows the credit spread option price functions at $t = 0.33333, 0.5, 0.66667$). Some option values for a different base on asset prices, for example, $70, 110, 170, 220,$ and 270, are listed in Table 2 for different interest rates, for instance, $r = 0.03, 0.35, 0.55,$ and 0.9. In addition, Figure 3 shows a comparison of option prices for the mentioned interest rates, which shows that interest rates and option prices are inversely related.

By selecting $\Delta x = \Delta y = 1$ and $\Delta t = \frac{1}{4}$, the error results of time steps were obtained $0.6515, 0.4798, 0.3173, 0.0213,$ and 0, respectively. As the time steps got smaller, these errors became smaller (by selecting $\Delta x = \Delta y = \frac{1}{2}$ and $\Delta t = \frac{1}{16}$, the error results of time steps were obtained $0.3173, 0.0030, 0.0009, 0.0006,$ and 0, respectively), but in both cases, because our selected condition has been extracted from the stability range, the result of the error confirms the correctness of this range. As you can see, the resulting errors decrease and tend to zero. So, this method is convergent and stable [9].

Table 2: Option values for the different underlying asset prices

P(x,r,τ)	x				
	70	110	170	220	270
r=0.03	925	1386.9	2079.8	2657.2	3234.7
r=0.35	922.2	1382.8	2073.7	2649.4	3225.3
r=0.55	920.6	1380.3	2069.9	2644.6	3219.3
r=0.9	917.6	1375.9	2063.2	2636.1	3209

6 Conclusions

The main goal of this article was to propose a pricing model, which is based on the credit spread and the interest rate, such that some jumps on the second are possible. Though the bringing jump in the modeling seems an interesting idea, it can bring some difficulties when it comes to implementing a numerical method. Therefore, to deal with this problem, as the second step, we postulate a fairly new numerical method call, ADI with step-size $\frac{1}{6}$ combined with Adams–Bashforth formula [9]. To continue this work, one good idea is to use other methods such as the machine learning, Chebyshev cardinal wavelets, and wavelets Galerkin method to obtain the results [4, 3]. Definitively, comparing the results of different methods could be highly instructive.

References

1. Deng, S-J.,Johnson, B. and Sogomonian, A. *Exotic electricity options and the valuation of electricity generation and transmission assets*, Decis. Support Syst. 30 (2001) 383–392.

2. Evans, G., Blackledge, J. and Yardley, P. *Numerical methods for partial differential equations*, Springer-Verlag London, Ltd., London, 2000.

3. Heydari, M.H., Hooshmandasl, M.R., Loghmani, Gh. Barid, and Cattani, C. *Wavelets Galerkin method for solving stochastic heat equation*, Int. J. Comput. Math. 93(9), (2015) 1579–1596.

4. Heydari, M.H., Mahmoudi, M.R., Shakiba, A. and Avazzadeh, Z. *Chebyshev cardinal wavelets and their application in solving nonlinear stochastic differential equations with fractional Brownian motion*, Commun. Nonlinear Sci. Numer. Simul. 64 (2018), 98–121.

5. in 't Hout, K.J. and Mishra, C. *Stability of ADI schemes for multidimensional diffusion equations with mixed derivative terms*, Appl. Numer. Math. 74 (2013), 83–94.

6. Jeong, D. and Kim, J. *A comparison study of ADI and operator splitting methods on option pricing models*, J. Comput. Appl. Math. 247 (2013), 162–171.

7. LeVeque, R.J. *Finite difference methods for ordinary and partial differential equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.

8. Li, L., Jiang, Z., Yin, Z. *Fourth-order compact finite difference method for solving two-dimensional convection-diffusion equation*, Adv. Difference Equ. 2018, Paper No. 234, 24 pp.

9. Liao, H-L. and Sun, Z-Z. *Maximum Norm Error Bounds of ADI and Compact ADI Methods for Solving Parabolic Equations*, Numer. Methods Partial Differential Equations 26 (2010), 37–60.

10. Longstaff, F. A., Schwartz, E.S. *Valuing credit derivatives*, J. Fixed Income, 6 (1995) 6–12.

11. Marchuk, G.I. *Splitting and alternating direction methods*, Handbook of numerical analysis, Vol. I, 197–462, Handb. Numer. Anal., I, North-Holland, Amsterdam, 1990.

12. Mohamadinejad, R., Biazar, J. and Neisy, A. *Spread Option Pricing Using Two Jump-Diffusion Interest Rates*, Politehn. Univ. Bucharest Sci. Bull. Ser. A Appl. Math. Phys. 82 (2020) 171–182.

13. O'Brien, G.G., Hyman, M.A. and Kaplan, S. *A study of the numerical solution of partial differential equations*, J. Math. Phys. 29, (1951) 223–251.

14. Østerby, O. *Numerical solution of parabolic equations*, Department of Computer Science, Aarhus University, 2015.

15. Østerby, O. *On the stability of ADI methods*, DAIMI Report Series, 42 (2017).

16. Safaei, M., Neisy, A. and Nematollahi, N. *New Splitting Scheme for Pricing American Options Under the Heston Model*, Comput. Econ. 52 (2018) 405–420.

17. Strikwerda, J. *Finite difference schemes and partial differential equations*, SIAM, Philadelphia, PA (2007).

18. Su, X. and Wang, W.*Pricing options with credit risk in reduced form model*, J. Korean Statist. Soc. 41 (2012) 437–444.

19. Tchuindjo, L. *Closed-form solutions for pricing credit-risky bonds and bond options*, Appl. Math. Comput. 217 (2011) 6133–6143.

20. Unger, A.J.A. *Pricing index-based catastrophe bonds: Part 1 Formulation and discretization issues using a numerical PDE approach*, Comput. and Geosci. 36 (2010) 150–160.

21. Zhou, R., Du, S., Yu, M. and Yang, F.*Pricing credit spread option with Longstaff-Schwartz and GARCH models in Chinese bond market*, J. Syst. Sci. Complex, 28 (2015) 1363–1373.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 211–219
DOI:10.22067/ijnao.2021.67048.0

Research Article

An adaptive descent extension of the Polak–Rebière–Polyak conjugate gradient method based on the concept of maximum magnification†

Z. Aminifard* and S. Babaie-Kafaki

Abstract

Recently, a one-parameter extension of the Polak–Rebière–Polyak method has been suggested, having acceptable theoretical features and promising numerical behavior. Here, based on an eigenvalue analysis on the method with the aim of avoiding a search direction in the direction of the maximum magnification by a symmetric version of the search direction matrix, an adaptive formula for computing parameter of the method is proposed. Under standard assumptions, the given formula ensures the sufficient descent property and guarantees the global convergence of the method. Numerical experiments are done on a collection of CUTEr test problems. They show practical effectiveness of the suggested formula for the parameter of the method.

AMS subject classifications (2020): 90C53, 65K05.

Keywords: Unconstrained optimization; Conjugate gradient method; Maximum magnification; Line search.

---

*Corresponding author
Received 19 November 2020; revised 31 December 2020; accepted 31 January 2021
Zohre Aminifard
Department of Mathematics, Semnan University, P.O. Box: 35195–363, Semnan, Iran.
E-mail: aminisor@semnan.ac.ir

Saman Babaie-Kafaki
Department of Mathematics, Semnan University, P.O. Box: 35195–363, Semnan, Iran.
E-mail: sbk@semnan.ac.ir

† This article was suggested by the Scientific Committee of the "13^{th} International Conference of Iranian Operations Research Society" for publication in IJNAO, which was accepted after independent review.

1 Introduction

Conjugate gradient (CG) methods can be regarded as the most popular optimization techniques due to their wide applications in the practical fields [1, 11, 12, 17]. CG algorithms are advantageous because of affordable memory storage, the simple structure of the iterative formula, promising computational performance, and acceptable convergence properties [5, 9, 13].

General form of an unconstrained optimization problem can be given by

$$\min_{x\in\mathbb{R}^n} f(x),$$

where f is a smooth real-valued nonlinear function with the gradient $g(x)$. Starting from some point $x_0 \in \mathbb{R}^n$, iterations of the CG algorithms are in the form of $x_{k+1} = x_k + s_k$ and $s_k = \alpha_k d_k$, for all $k \geq 0$, where $\alpha_k > 0$ is a step length often determined by some inexact line search techniques along the direction d_k calculated by

$$d_0 = -g_0,\ d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \geq 0, \tag{1}$$

in which $\beta_k \in \mathbb{R}$ is called the CG parameter and $g_k = g(x_k)$. Among the various classical CG techniques, the Polak–Rebière–Polyak (PRP) method with

$$\beta_k^{\mathrm{PRP}} = \frac{g_{k+1}^T y_k}{||g_k||^2},$$

in which $y_k = g_{k+1} - g_k$ and $||\cdot||$ denotes the ℓ_2 norm, is regarded as an efficiently popular classic method, mainly because of adaptive restarts when dealing with improper search directions [9].

Although being computationally advantageous, the PRP method fails to ensure the descent property [9]. So, significant attention have been paid to get descent modifications of the PRP method. For example, Zhang, Zhou, and Li [18] developed (ZZL) a three-term extension of the method by

$$d_0 = -g_0,\ d_{k+1}^{\mathrm{ZZL}} = -g_{k+1} + \beta_k^{\mathrm{PRP}} d_k - \frac{g_{k+1}^T d_k}{||g_k||^2} y_k,\ k \geq 0, \tag{2}$$

satisfying the sufficient descent condition, that is,

$$d_k^T g_k \leq -\tau ||g_k||^2, \quad k \geq 0, \tag{3}$$

where $\tau > 0$ is a constant. In another effort, Andrei [3] proposed a spectral PRP (SPRP) method with

$$d_0 = -g_0,\ d_{k+1}^{\mathrm{SPRP}} = -\frac{s_k^T y_k}{||g_k||^2} g_{k+1} + \beta_k^{\mathrm{PRP}} s_k - \frac{g_{k+1}^T s_k}{||g_k||^2} y_k,\ k \geq 0, \tag{4}$$

which in addition to (3), fulfills the effective Dai–Liao conjugacy condition [6]. Also, Babaie-Kafaki and Ghanbari [4] developed a class of one-parameter extension of β_k^{PRP} (EPRP) based on the Dai–Liao approach [6]; that is,

$$\beta_k^{\mathrm{EPRP}} = \beta_k^{\mathrm{PRP}} - t\frac{g_{k+1}^T d_k}{||g_k||^2}, \tag{5}$$

where t is a positive parameter. Then, to find an optimal choice for t, they noted that from (1) and (5) search directions of EPRP can be written as

$$d_{k+1} = -H_{k+1} g_{k+1},$$

where

$$H_{k+1} = I - \frac{d_k y_k^T}{||g_k||^2} + t\frac{d_k d_k^T}{||g_k||^2}.$$

Symmetrizing H_{k+1} by

$$P_{k+1} = \frac{H_{k+1} + H_{k+1}^T}{2} = I - \frac{1}{2}\frac{d_k y_k^T + y_k d_k^T}{||g_k||^2} + t\frac{d_k d_k^T}{||g_k||^2}, \tag{6}$$

in light of an eigenvalue analysis, the following family of two-parameter choices for t was suggested in [4]:

$$t_k^{p,q} = p\frac{||y_k||^2}{||g_k||^2} + q\left(\frac{1}{2}\frac{d_k^T y_k}{||d_k||\ ||g_k||} - \frac{||g_k||}{||d_k||}\right)^2, \tag{7}$$

with $p > \dfrac{1}{4}$ and $q \geq -1$, guaranteeing the descent condition.

Following such studies, here we deal with another choice for parameter of the EPRP method based on the concept of the maximum magnification by a matrix. Organization of our study is summarized as follows. In Section 2, after analyzing eigenvalues of P_{k+1}, we introduce our new formula for the parameter t of the EPRP method. Also, we conduct a brief global convergence analysis. In Section 3, we make some competitive computational experiments on a collection of CUTEr problems, using the Dolan–Moré performance profile. Finally, concluding remarks are given in Section 4.

2 An adaptive choice for parameter of the extended Polak–Ribièr–Polyak method

Here, firstly we conduct a concise eigenvalue analysis on P_{k+1} in order to explain our adaptive way of computing t in (5). Hereafter, we assume that $d_k^T y_k > 0$, as ensured by the strong Wolfe line search conditions [14], that is,

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k, \tag{8}$$
$$|\nabla f(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{9}$$

with $0 < \delta < \sigma < 1$. The following basic definition is the kernel of our analysis.

Definition 1. [15] For an arbitrary matrix $A \in \mathbb{R}^{n\times m}$, the scalar

$$maxmag(A) = \max_{x\neq 0} \frac{||Ax||}{||x||},$$

is called the maximum magnification by A. Hence, $maxmag(A) = ||A||$, and also, the vector $x \neq 0$ for which $||Ax|| = ||A||\ ||x||$, is in the direction of the maximum magnification by the matrix A.

Firstly, note that the matrix P_{k+1} given by (6) can be regarded as a symmetric approximation of the search direction matrix H_{k+1}. Based on the analysis of [4], eigenvalues of P_{k+1} are 1 with multiplicity $n-2$, and λ_k^+ and λ_k^- are given by

$$\begin{aligned}\lambda_k^{\pm} = 1 &+ \frac{1}{2||g_k||^2}\left(t||d_k||^2 - d_k^T y_k\right)\\ &\pm \frac{1}{2||g_k||^2}\sqrt{(t||d_k||^2 - d_k^T y_k)^2 + ||d_k||^2||y_k||^2 - (d_k^T y_k)^2}.\end{aligned}$$

It can be seen that with the choice (7), we have $\lambda_k^+ \geq 1 \geq \lambda_k^- > 0$, and consequently, $||P_{k+1}|| = \lambda_k^+$. Also, in light of similar analysis carried out in [2], the eigenvector of P_{k+1} corresponding to λ_k^+, here called v_1^k, can be written as $v_1^k = \gamma d_k + \vartheta y_k$ in which

$$\gamma = \frac{2(1-\lambda_k^+)||g_k||^2 - d_k^T y_k}{||d_k||^2}\vartheta.$$

So, v_1^k as a vector in the direction of the maximum magnification by P_{k+1} is specified.

As explained in [2], when the gradient is approximately parallel with the direction of the maximum magnification by H_{k+1}, then EPRP may face with some numerical errors and also, it may converge hardly. Based on this fact and since P_{k+1} is a symmetric approximation of H_{k+1}, it can be stated that if g_{k+1} is as far away as possible from the direction of the maximum magnification by P_{k+1}, then the mentioned possible errors may be diminished and the convergence may be improved. Hence, we obtain a formula for the EPRP parameter by making v_1^k to be orthogonal to g_{k+1} in the sense of solving the equation $g_{k+1}^T v_1^k = 0$; that is,

$$\bar{t}_k = \frac{||y_k||^2(g_{k+1}^T d_k)^2 - ||d_k||^2(g_{k+1}^T y_k)^2}{2(g_{k+1}^T d_k)\left((d_k^T y_k)(g_{k+1}^T d_k) - ||d_k||^2(g_{k+1}^T y_k)\right)}. \tag{10}$$

Now, for the sake of positiveness of the EPRP parameter and to achieve the sufficient descent property, we suggest the following modified version of (10):

$$t_k^* = \begin{cases} \max\left\{\bar{t}_k, \vartheta \dfrac{||y_k||^2}{||g_k||^2}\right\}, & \text{if denominator of } \bar{t}_k \text{ is nonzero,} \\ \\ \vartheta \dfrac{||y_k||^2}{||g_k||^2}, & \text{otherwise,} \end{cases} \tag{11}$$

with $\vartheta > \dfrac{1}{4}$.

Now, similar to the analysis conducted in the proof of [16, Theorem 3.2], the following convergence result can be established for the EPRP method. The proof is omitted to avoid repetition.

Theorem 1. Suppose that the level set $\Omega = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is bounded and in some neighborhood $\mathcal{N}$ of Ω, that the objective function f is smooth and also, that ∇f is Lipschitz continuous. For the EPRP method with the parameter (5), assume that t is equal to t_k^* defined by (11) and the line search fulfills the strong Wolfe conditions (8) and (9). If there exists a positive lower bound α^* for the step lengths α_k (for all $k \geq 0$), then $\lim\limits_{k\to\infty} ||g_k|| = 0$.

3 Computational experiments

In this section, we examine the numerical efficiency of the EPRP method in which t is computed by (11) with $\vartheta = 0.26$ and (7) with $(p, q) = (1, 0)$; here the corresponding methods are, respectively, called EPRP1 and EPRP2. The methods are compared by the two modified PRP methods of ZZL and SPRP, respectively, with the search directions (2) and (4) [3, 18]. We have implemented all the algorithms on a set of 45 test functions of the CUTEr library [8] with $n \geq 50$, as given in Table 1, in MATLAB software environment. Hardware and software detailed specifications have been clarified in [2], together with the strong Wolfe line search features and the stopping criteria. Detailed outputs have been provided in Table 1.

Efficiency of the algorithms was compared by applying the performance profile proposed by Dolan and Moré [7] on the norm of gradient, the CPU time (CPUT), and the total number of function and gradient evaluations (TNFGE), following the notation of [10]. Results are shown by Figures 1–3. As seen, EPRP is preferable to the other methods. Particularly, the results show that the choice (11) for the EPRP parameter is practically effective. Our experiments showed that averagely in 62.63% of the iterations of EPRP1, we had $t_k^* = \bar{t}_k$.

Table 1: Outputs

Function	n	EPRP1 TNFGE	EPRP1 CPUT	EPRP2 TNFGE	EPRP2 CPUT	SPRP TNFGE	SPRP CPUT	ZZL TNFGE	ZZL CPUT
ARGLINA	200	12	2.13E-01	12	9.40E-02	12	7.97E-02	12	8.93E-02
ARWHEAD	5000	103	2.17E-01	20095	4.25E+00	40053	8.25E+00	23730	4.89E+00
BDEXP	5000	12	9.73E-02	12	7.68E-02	12	6.97E-02	12	6.44E-02
BDQRTIC	5000	26021	1.62E+01	26069	1.34E+01	40023	1.13E+01	26073	6.95E+00
BQPGABIM	50	299	7.67E-02	238	4.76E-02	2131	1.02E-01	685	5.10E-02
BQPGASIM	50	299	3.74E-02	238	3.18E-02	2131	9.70E-02	685	5.05E-02
BROWNAL	200	4641	6.26E-01	38227	4.25E+00	40084	3.93E+00	40084	4.07E+00
BRYBND	5000	659	4.07E-01	1853	6.93E-01	12334	4.62E+00	493	2.71E-01
CHENHARK	5000	816	2.57E-01	633	1.79E-01	3196	5.23E-01	6832	1.46E+00
COSINE	10000	27	1.29E-01	37	1.15E-01	46	1.33E-01	38	1.18E-01
CRAGGLVY	5000	431	6.99E-01	447	5.75E-01	3204	2.90E+00	1127	1.38E+00
CURLY10	10000	457	3.66E-01	479	3.60E-01	2597	1.27E+00	1273	8.53E-01
DIXMAANA	3000	36	6.02E-02	31	4.79E-02	31	4.95E-02	31	4.92E-02
DIXMAANB	3000	32	6.26E-02	27	5.28E-02	32	5.96E-02	32	4.76E-02
DIXMAANC	3000	32	6.10E-02	32	4.90E-02	32	5.10E-02	32	5.40E-02
DIXMAAND	3000	37	5.74E-02	37	5.18E-02	37	5.10E-02	37	5.09E-02
DQDRTIC	5000	1408	5.81E-01	1652	5.66E-01	11183	2.71E+00	4661	1.62E+00
DQRTIC	5000	4	4.89E-02	4	3.88E-02	4	4.15E-02	4	4.44E-02
DRCAV1LQ	4489	4	9.58E-02	4	8.56E-02	4	8.18E-02	4	7.50E-02
DRCAV2LQ	4489	4	8.63E-02	4	1.03E-01	4	8.63E-02	4	1.03E-01
DRCAV3LQ	4489	4	8.66E-02	4	8.95E-02	4	9.29E-02	4	8.79E-02
EDENSCH	2000	76	8.14E-02	89	7.10E-02	94	6.73E-02	89	7.07E-02
EG2	1000	23	3.52E-02	23	4.09E-02	23	3.23E-02	23	2.21E-02
ENGVAL1	5000	58	1.14E-01	58	9.18E-02	58	9.11E-02	62	9.48E-02
FLETCBV2	5000	4	1.07E-01	4	8.07E-02	4	8.45E-02	4	7.25E-02
FLETCBV3	5000	164	1.89E-01	164	1.62E-01	160	1.88E-01	164	1.61E-01
FLETCHBV	5000	156	1.50E-01	156	1.44E-01	148	1.45E-01	156	1.48E-01
FREUROTH	5000	275	2.21E-01	167	1.73E-01	2109	9.82E-01	987	5.74E-01
GENHUMPS	5000	4	6.25E-02	4	6.28E-02	4	5.86E-02	4	6.33E-02
MANCINO	100	93	3.20E-01	93	2.49E-01	98	2.56E-01	98	2.55E-01
MOREBV	5000	95	9.49E-02	115	9.50E-02	971	2.78E-01	380	1.70E-01
NCB20	5010	372	6.34E-01	905	1.26E+00	1505	1.85E+00	545	7.49E-01
NONCVXU2	5000	4	5.49E-02	4	4.40E-02	4	4.72E-02	4	3.32E-02
PENALTY2	200	4	2.88E-02	4	1.95E-02	4	8.67E-03	4	9.30E-03
QUARTC	5000	4	5.72E-02	4	3.82E-02	4	4.24E-02	4	3.13E-02
SCHMVETT	5000	55	1.94E-01	54	1.63E-01	58	1.67E-01	54	1.54E-01
SENSORS	100	88	4.51E-01	87	3.83E-01	110	5.27E-01	98	4.10E-01
SINQUAD	5000	123	3.00E-01	106	2.88E-01	191	3.37E-01	117	2.82E-01
SPARSQUR	10000	274	4.10E-01	187	2.81E-01	226	3.12E-01	235	3.36E-01
TOINTGOR	50	655	7.83E-02	725	5.58E-02	4425	1.96E-01	1812	1.26E-01
TOINTGSS	5000	83	1.81E-01	64	1.07E-01	1072	1.02E+00	321	3.57E-01
TOINTQOR	50	123	2.57E-02	145	2.39E-02	166	2.21E-02	244	2.67E-02
VARDIM	200	8	4.23E-02	8	1.94E-02	8	3.49E-03	8	8.71E-03
VAREIGVL	50	113	3.45E-02	122	3.85E-02	132	3.05E-02	133	8.70E-03
WOODS	4000	164	1.22E-01	180	9.18E-02	746	1.84E-01	159	8.35E-02

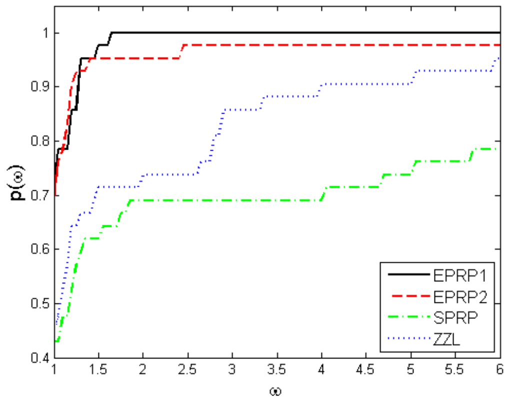


Figure 1: TNFGE performance profiles

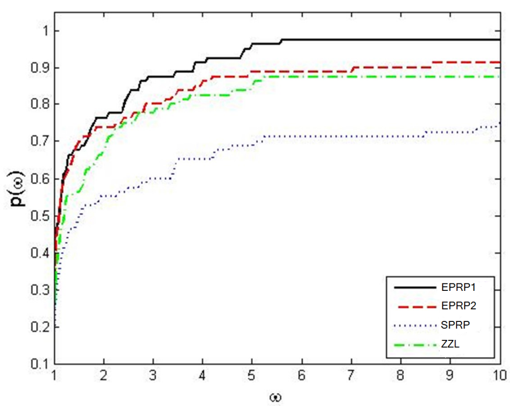


Figure 2: CPUT performance profiles

4 Conclusion

Based on the concept of maximum magnification, we have conducted an eigenvalue analysis to suggest an optimal choice for the parameter of the recently proposed EPRP method. The suggested formula guarantees the sufficient descent property as well as the global convergence of the method. Effect of the proposed formula has been numerically investigated in contrast to several other modifications made on the classical PRP method. Results showed the effectiveness of the suggested choice for the EPRP parameter.

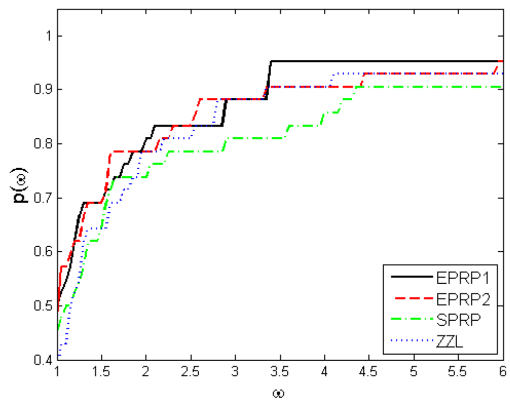


Figure 3: Norm of gradient performance profiles

References

1. Abubakar, A.B., Kumam, P. and Awwal, A.M. *Global convergence via descent modified three-term conjugate gradient projection algorithm with applications to signal recovery*, Result. Appl. Math. 4 (2019), 100069.

2. Aminifard, Z. and Babaie-Kafaki, S. *An optimal parameter choice for the Dai–Liao family of conjugate gradient methods by avoiding a direction of the maximum magnification by the search direction matrix*, 4OR, 17(3) (2019), 317–330.

3. Andrei, N. *A modified Polak–Ribière–Polyak conjugate gradient algorithm for unconstrained optimization*, Optimization, 60(12) (2011), 1457–1471.

4. Babaie-Kafaki, S. and Ghanbari, R. *A descent extension of the Polak–Ribière–Polyak conjugate gradient method*, Comput. Math. Appl. 68(12) (2014), 2005–2011.

5. Dai, Y.H., Han, J.Y., Liu, G.H., Sun, D.F., Yin, H.X. and Yuan, Y.X. *Convergence properties of nonlinear conjugate gradient methods*, SIAM J. Optim. 10(2) (1999), 348–358.

6. Dai, Y. H. and Liao, L. Z. *New conjugacy conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim. 43(1) (2001), 87–101.

7. Dolan, E.D. and Moré, J.J. *Benchmarking optimization software with performance profiles*, Math. Programming (Ser. A), 91(2) (2002), 201–213.

8. Gould, N.I.M., Orban, D. and Toint, Ph.L. *CUTEr: A constrained and unconstrained testing environment*, ACM Trans. Math. Software, 29(4) (2003), 373–394.

9. Hager, W.W. and Zhang, H. *A survey of nonlinear conjugate gradient methods*, Pac. J. Optim. 2(1) (2006), 35–58.

10. Hager, W. W. and Zhang, H. *Algorithm 851: CG-Descent, a conjugate gradient method with guaranteed descent*, ACM Trans. Math. Software, 32(1) (2006), 113–137.

11. Heravi, A.R. and Hodtani, G.A. *A new correntropy-based conjugate gradient backpropagation algorithm for improving training in neural networks*, IEEE Trans. Neural Netw. Learn. Syst. 29(12) (2018), 6252–6263.

12. Lin, J. and Jiang, C., *An improved conjugate gradient parametric detection based on space-time scan*, Signal Process. 169 (2020), 107412.

13. Nocedal, J. and Wright S.J. *Numerical optimization*, Springer, New York, 2006.

14. Sun, W. and Yuan, Y.X. *Optimization theory and methods: Nonlinear programming*, Springer, New York, 2006.

15. Watkins, D.S. *Fundamentals of matrix computations*, John Wiley and Sons, New York, 2002.

16. Yuan, G., *Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems*, Optim. Lett. 3(1) (2009), 11–21.

17. Yuan, G., Lu, J. and Wang, Z. *The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems*, Appl. Numer. Math. 152 (2020), 1–11.

18. Zhang, L., Zhou, W. and Li, D.H. *A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence*, IMA J. Numer. Anal. 26(4) (2006), 629–640.

Iranian Journal of Numerical Analysis and Optimization
Vol. 11, No. 1, (2021), pp 221–233
DOI:10.22067/ijnao.2021.68901.1015

Research Article

Some applications of Sigmoid functions

M. A. Jafari, and A. Aminataei*

Abstract

In numerical analysis, the process of fitting a function via given data is called interpolation. Interpolation has many applications in engineering and science. There are several formal kinds of interpolation, including linear interpolation, polynomial interpolation, piecewise constant interpolation, trigonometric interpolation, and so on. In this article, by using Sigmoid functions, a new type of interpolation formula is presented. To illustrate the efficiency of the proposed new interpolation formulas, some applications in quadrature formulas (in both open and closed types), numerical integration for double integral, and numerical solution of an ordinary differential equation are included. The advantage of this new approach is shown in the numerical applications section.

AMS subject classifications (2020): 65D05, 65D30, 65D32.

Keywords: Sigmoid function; Interpolation; Numerical integration; Quadrature formula; Numerical solution of ordinary differential equation.

1 Introduction

Interpolation is the process of finding a function via given data. Interpolation has many applications in engineering and science. There are several formal kinds of interpolation, including linear interpolation, polynomial interpolation, piecewise constant interpolation, trigonometric interpolation, and so on. For example, consider the polynomial interpolation. Let $f_0, f_1, \ldots, f_n$ be known values for an arbitrary function $f : \mathbb{R} \to \mathbb{R}$ at points $x = x_i$, $i = 0, 1, \ldots, n$. Then the Lagrange basis functions are defined as

*Corresponding author
Received 13 Febuary 2021; revised 26 Febuary 2021; accepted 26 Febuary 2021
Mohammad Ali Jafari
Department of Mathematical Finance, Faculty of Financial Sciences, Kharazmi University, Tehran, Iran. e-mail: m.a.jafari.math@gmail.com

Azim Aminataei
Faculty of Mathematics, K. N. Toosi University of Technology, Tehran, Iran. e-mail: ataei@kntu.ac.ir

$$L_{n,i}(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{(x - x_k)}{(x_i - x_k)}, \quad i = 0, 1, \ldots, n. \tag{1}$$

Then the interpolating polynomial of degree n is defined as

$$p(x) = \sum_{i=0}^{n} L_{n,i}(x) f_i. \tag{2}$$

Clearly, $p(x)$ satisfies $p(x_i) = f_i$ for $i = 0, 1, \ldots, n$. Moreover, for any $f \in C^{n+1}[a, b]$, we have

$$f(x) = \sum_{i=0}^{n} L_{n,i}(x) f_i + (x - x_0)(x - x_1) \cdots (x - x_n) \frac{f^{(n+1)}(\eta_x)}{(n+1)!}, \quad \eta_x \in [a, b]. \tag{3}$$

Using relation (2), the classical Newton–Cotes quadrature formulas are obtained [2, 3, 6, 7, 8, 9]. In this article, by using Sigmoid functions [1, 4, 5], new types of interpolation formulas are presented. These new types of interpolation result in new integration formulas and new formulas for solving ordinary differential equations.

Sigmoid functions are mathematical functions with S-shaped curves or sigmoid curves. These functions are bounded and monotone and have the first derivative that is bell shapes. Additionally, these functions are constrained by a pair of horizontal asymptotes as $x \longrightarrow \pm\infty$. Some of them are listed below.

a. **Logistic function** is defined by the formula

$$\phi(x) = \frac{1}{1 + e^{-x}}. \tag{4}$$

b. **Hyperbolic tangent function** is defined by the formula

$$\phi(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{5}$$

c. **Arctangent function** is defined by

$$\phi(x) = \arctan(x). \tag{6}$$

d. **Gudermannian function** is defined by

$$\phi(x) = gd(x) = \int_0^x \frac{1}{\cosh x} dx = 2 \arctan(\tanh(\frac{x}{2})). \tag{7}$$

e. **Error function** is defined as

$$\phi(x) = erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \tag{8}$$

f. **Generalised logistic function** is defined as

$$\phi(x) = (1 + e^{-x})^{-\alpha}, \quad \alpha > 0. \tag{9}$$

g. Some algebraic functions, for example,

$$\phi(x) = \frac{x}{\sqrt{1 + x^2}}. \tag{10}$$

h. **Smoothstep function** is defined for $N \geq 1$ as

$$\phi(x) = \begin{cases} \left(\int_0^1 (1 - u^2)^N du \right)^{-1} \int_0^x (1 - u^2)^N du, & |x| \leq 1, \\ sgn(x), & |x| \geq 1. \end{cases} \tag{11}$$

I. The integral of any continuous, nonnegative, and "bump-shaped" function will be sigmoid function; therefore the cumulative distribution functions for many common probability distributions are sigmoid functions.

This article is organized as follows. In section 2, new methods of interpolation are introduced. In section 3, some applications of the proposed new interpolation formulas are provided. Finally, in section 4, conclusions are presented.

2 Main results

In this section, the new types of interpolation formulas for one- and two-dimensional cases by using Sigmoid functions are obtained.

2.1 One-dimensional case

Theorem 1. Let ϕ be an arbitrary Sigmoid function and let

$$\begin{cases} \lim\limits_{x \to -\infty} \phi(x) = a, \\ \lim\limits_{x \to +\infty} \phi(x) = b. \end{cases} \tag{12}$$

Let $f_0, f_1, \ldots, f_n$ be known values for an arbitrary function at points $x_0 < x_1 < \cdots < x_n$, and let $x_{i+1} - x_i = h_i$ for $i = 0, 1, 2, \ldots, n$. Then

$$S(x) = \sum_{i=0}^{n} \left(\frac{1}{b - a} \right) f_i \left[\phi(a_i x - b_i) - \phi(a_{i+1} x - b_{i+1}) \right] \tag{13}$$

is the interpolation formula for the above data points. Moreover, a_i for $i = 0, 1, 2, \ldots, n+1$ are positive and big enough numbers, $\dfrac{b_i}{a_i} = x_i - \dfrac{h_i}{2}$ for $i = 0, 1, 2, \ldots, n$, and $\dfrac{b_{n+1}}{a_{n+1}} = x_n + \dfrac{h_i}{2}$.

Proof. According to the properties of the Sigmoid functions, $0 \leqslant \phi(a_i x - b_i) - \phi(a_{i+1}x - b_{i+1}) \leqslant b - a$. Therefore, for positive and big enough a_i, in a neighborhood of $x = \dfrac{b_i}{a_i}$, the function $[\phi(a_i x - b_i) - \phi(a_{i+1}x - b_{i+1})]$ changes from zero to $b - a$ (for $i = 0, 1, \ldots, n$). Similarly, at $x = \dfrac{a_{i+1}}{b_{i+1}}$, the function $[\phi(a_i x - b_i) - \phi(a_{i+1}x - b_{i+1})]$ changes from $b - a$ to zero (for $i = 0, 1, \ldots, n$). Therefore, for the interior points in intervals $\left[\dfrac{b_i}{a_i}, \dfrac{b_{i+1}}{a_{i+1}}\right]$, for $i = 0, 1, \ldots, n$, $S(x)$ is equal to f_i. In a neighborhood of $x = \dfrac{b_i}{a_i}$ (for $i = 1, 2, \ldots, n$), $S(x)$ is changed from f_{i-1} to f_i. In the same way, in a neighborhood of $x = \dfrac{b_{i+1}}{a_{i+1}}$ (for $i = 0, 1, \ldots, n-1$), $S(x)$ is transformed from f_i to f_{i+1}. In addition, in a neighborhood of $x = \dfrac{b_0}{a_0}$, $S(x)$ is changed from zero to f_0, and in a neighborhood of $x = \dfrac{b_{n+1}}{a_{n+1}}$, $S(x)$ is changed from f_n to zero. Speed of changing can be increased by increasing a_i (for $i = 0, 1, \ldots, n+1$). In other words, a_i and $\dfrac{b_i}{a_i}$, $i = 0, 1, \ldots, n+1$, define the speed and locations of the changes. Therefore, $S(x)$ is the interpolation formula for aforesaid points. $\square$

2.2 Two-dimensional case

By using the above idea for two-dimensional problems, a new type of two-dimensional interpolation formulas can be obtained.

Theorem 2. Let $F(x_i, y_j) = z_{i,j}$ for $i = 0, 1, 2, \ldots, n$ and $j = 0, 1, 2, \ldots, m$. Let ϕ and ψ be two arbitrary Sigmoid functions and let

$$\begin{cases} \lim\limits_{x\to-\infty} \phi(x) = a, \\ \lim\limits_{x\to+\infty} \phi(x) = b, \end{cases} \qquad \begin{cases} \lim\limits_{x\to-\infty} \psi(x) = c, \\ \lim\limits_{x\to+\infty} \psi(x) = d. \end{cases} \tag{14}$$

Let $x_0 < x_1 < \cdots < x_n$ and $y_0 < y_1 < \cdots < y_m$ (i.e., $x_{i+1} - x_i = h_i$, for $i = 0, 1, \ldots, n-1$, and $y_{j+1} - y_j = k_j$ for $j = 0, 1, \ldots, m-1$). Then

$$S(x,y)=\sum_{i=0}^{n}\sum_{j=0}^{m}\left(\frac{1}{b-a}\right)\left(\frac{1}{d-c}\right)z_{i,j} \quad (15)$$
$$[\phi(a_i x-b_i)-\phi(a_{i+1}x-b_{i+1})]\left[\psi(a_j^{'}y-b_j^{'})-\psi(a_{j+1}^{'}y-b_{j+1}^{'})\right]$$

is the interpolation formula for the above data points. Moreover, a_i for $i=0,1,2,\ldots,n+1$ and $a_j^{'}$ for $j=0,1,2,\ldots,m+1$ are positive and big enough numbers. In addition

$$\begin{cases} \dfrac{b_i}{a_i}=x_i-\dfrac{h_i}{2}, \quad i=0,1,2,\ldots,n \quad \textit{and} \quad \dfrac{b_{n+1}}{a_{n+1}}=x_n+\dfrac{h_n}{2}, \\ \dfrac{b_j^{'}}{a_j^{'}}=y_j-\dfrac{k_j}{2}, \quad j=0,1,2,\ldots,m \quad \textit{and} \quad \dfrac{b_{m+1}^{'}}{a_{m+1}^{'}}=y_m+\dfrac{k_m}{2}. \end{cases} \quad (16)$$

Proof. By using relation (13) two times, relation (15) is obtained. □

In the next section, some numerical applications in interpolation, numerical integration, and numerical solution of ordinary differential equations are included.

3 Numerical applications

In this section, some applications of formula (13) in one-dimensional interpolation, numerical integration, and numerical solving of ordinary differential equations by using the Hyperbolic tangent function are given. Other formulas (i.e., (4)–(11)) can easily be applied. Also, in Example 6, formula (15) by using the Hyperbolic tangent function is applied.

Example 1 (Interpolation problem)**.** Let $f(0)=1$, $f(1)=2$, $f(2)=4$, $f(3)=7$, $f(4)=5$, and $f(5)=6$. Then using formula (13), the interpolation function is obtained as

$$\begin{aligned} S(x) &= \left(\frac{1}{2}\right)\sum_{i=0}^{5} f(x_i)\left[\tanh(a_i x-b_i)-\tanh(a_{i+1}x-b_{i+1})\right] \\ &= \left(\frac{1}{2}\right)(1\times[\tanh(a_0x-b_0)-\tanh(a_1x-b_1)] \\ &\quad +2\times[\tanh(a_1x-b_1)-\tanh(a_2x-b_2)] \\ &\quad +4\times[\tanh(a_2x-b_2)-\tanh(a_3x-b_3)] \\ &\quad +7\times[\tanh(a_3x-b_3)-\tanh(a_4x-b_4)] \\ &\quad +5\times[\tanh(a_4x-b_4)-\tanh(a_5x-b_5)] \\ &\quad +6\times[\tanh(a_5x-b_5)-\tanh(a_6x-b_6)]). \end{aligned} \quad (17)$$

This interpolation function in two cases (i.e., $a_0 = a_1 = \cdots = a_6 = 5$ and $a_0 = a_1 = \cdots = a_6 = 10$) is plotted in Figures 1 and 2.

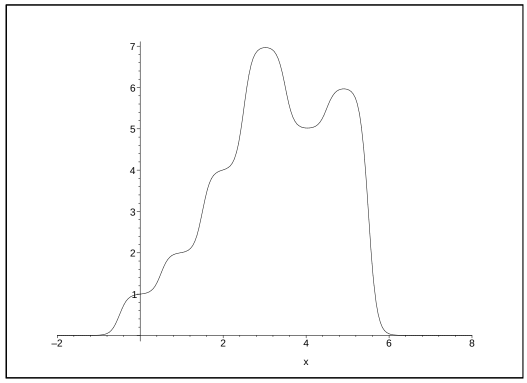


Figure 1: Interpolation function with $a_i = 5$ for $i = 0, 1, \ldots, 6$.

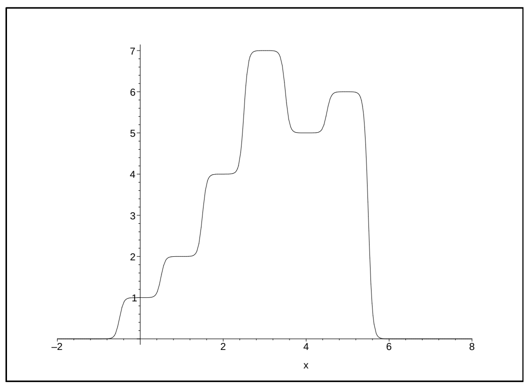


Figure 2: Interpolation function with $a_i = 10$ for $i = 0, 1, \ldots, 6$.

Example 2 (Interpolation problem (Runge's function))**.** Consider the polynomial interpolation of Runge's function defined as (see [9])

$$g(x) = \frac{1}{1 + 25x^2}, \quad x \in [-1, 1]. \tag{18}$$

The polynomial interpolating with $N = 15$ equidistant nodes is plotted in Figure 3. Also, using formula (13), the new interpolation is obtained. This interpolation function is plotted in Figure 4. The error function of polynomial interpolation is plotted in Figure 5. As you can see, the error at nonnodal points does not decrease with the increase of nodes; see [3, 9]. Also, the error function of the new interpolation is plotted in Figure 6.

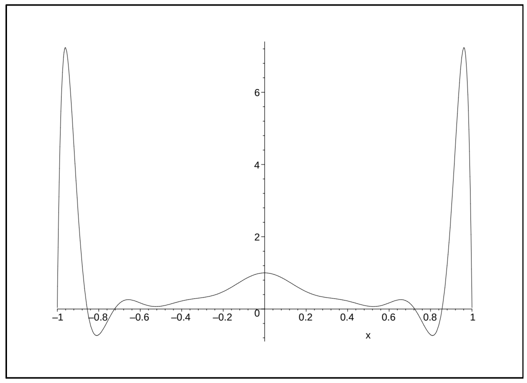


Figure 3: Polynomial interpolation function of Runge's function with $N = 15$ equidistant nodes in $[-1, 1]$.

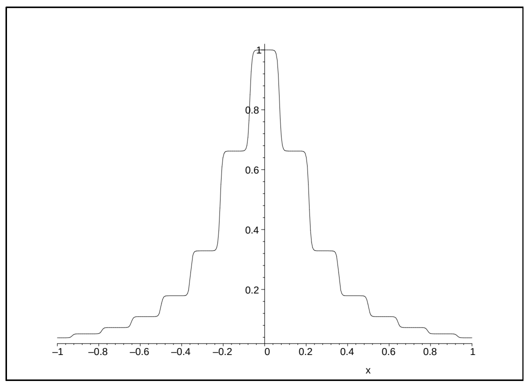


Figure 4: The new interpolation function of Runge's function with $N = 15$ equidistant nodes in $[-1, 1]$ and $a[i] = 100$ for $i = 0, 1, 2, \dots, 15$.

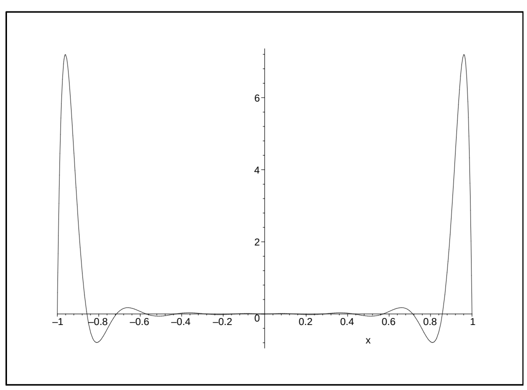


Figure 5: Error function of the polynomial interpolation function for Runge's function with $N = 15$ equidistant nodes in $[-1, 1]$.

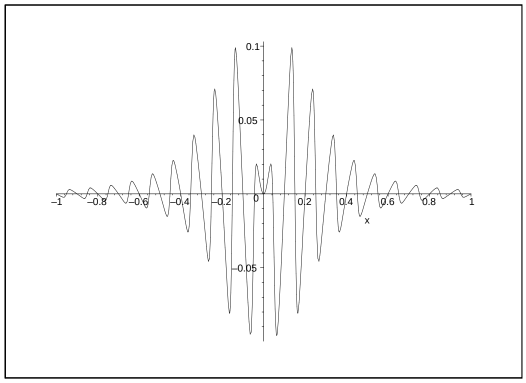


Figure 6: Error function of the new interpolation formula for Runge's function with $N = 15$ equidistant nodes in $[-1, 1]$ and $a[i] = 100$ for $i = 0, 1, 2, \dots, 15$.

Example 3 (Numerical integration (closed type))**.** Quadrature formula of Newton–Cotes on the finite interval $[a, b]$ is defined as (see [7, 8])

$$\int_a^b f(x)dx = \sum_{i=0}^{n} w_i f(x_i) + R_n(f), \tag{19}$$

where x_i are equidistantly distributed with the step size $\dfrac{b-a}{n}$ and $x_i = a+ih$, $i = 0, 1, \dots, n$. In addition, $R_n(f) = 0$ whenever $f \in \mathcal{P}_{n-1}$, where $\mathcal{P}_{n-1}$ is the space of all algebraic polynomials of degree at most $n-1$. Now, using relation (13) for $\phi(x) = \tanh(x)$ results in

$$\begin{aligned}\int_a^b f(x)dx &\approx \int_a^b S(x)dx \\ &= \int_a^b \sum_{i=0}^{n} \left(\frac{1}{2}\right) f_i \left[\tanh(a_i x - b_i) - \tanh(a_{i+1}x - b_{i+1})\right] dx \end{aligned} \tag{20}$$

Therefore, relation (20) is simplified as

$$\begin{aligned}\int_a^b f(x)dx \approx \sum_{i=0}^{n} \Bigg[&\left(\frac{1}{2a_i}\right) [\ln(\cosh(a_i b - b_i)) - \ln(\cosh(a_i a - b_i)] \\ &- \left(\frac{1}{2a_{i+1}}\right) [\ln(\cosh(a_{i+1} b - b_{i+1})) - \ln(\cosh(a_{i+1} a - b_{i+1})] \Bigg] f_i. \end{aligned} \tag{21}$$

As it is mentioned above, explicit forms of quadrature formulas based on this new interpolation formula can be directly obtained. As an example, consider the following integral:

Table 1: Absolute error of quadrature formula (21) for $n = 5(5)30$ and $f(x) = \exp(x^2)$.

n	Absolute error
5	1.8(-2)
10	4.5(-3)
15	2.0(-3)
20	1.0(-3)
25	4.0(-4)
30	1.2(-4)

$$\int_0^1 \exp(x^2)dx \approx 1.46265174590718. \tag{22}$$

The absolute error of formula (21) to approximate the above integral, for $a_i = 100, i = 0, 1, \ldots, n$, is presented in Table 1.

Example 4 (Numerical integration (open type))**.** In relation (19), if $w_0 = w_n = 0$, then the open type of Newton–Cotes formula is obtained. In this case, $R_n(f) = 0$ whenever $f \in \mathcal{P}_{n-3}$. The new quadrature formula in this case is as follows:

$$\begin{aligned}\int_a^b f(x)dx &\approx \int_a^b S(x)dx \qquad (23)\\ &= \int_a^b \sum_{i=1}^{n-1} \left(\frac{1}{2}\right) f_i \left[\tanh(a_i x - b_i) - \tanh(a_{i+1}x - b_{i+1})\right] dx.\end{aligned}$$

Therefore, relation (23) is simplified as

$$\begin{aligned}\int_a^b f(x)dx \approx \sum_{i=1}^{n-1} \Bigg[&\left(\frac{1}{2a_i}\right) [\ln(\cosh(a_i b - b_i)) - \ln(\cosh(a_i a - b_i)] \\ &- \left(\frac{1}{2a_{i+1}}\right) [\ln(\cosh(a_{i+1}b - b_{i+1})) - \ln(\cosh(a_{i+1}a - b_{i+1})] \Bigg] f_i.\end{aligned}$$

As an example, consider the following integral:

$$\int_0^1 x^{100} \sin\left(\frac{1}{x}\right) dx \approx 0.008384044187. \tag{24}$$

The absolute error of formula (24) to approximate the above integral, for $a_i = 100, i = 0, 1, \ldots, n$, is presented in Table 2.

Example 5 (Numerical methods for solving ordinary differential equation)**.** Consider the following ODE:

Table 2: Absolute error of quadrature formula (24) for $n = 5(5)30$ and $f(x) = x^{100}\sin(\frac{1}{x})$.

n	Absolute error
5	8.38(-3)
10	8.38(-3)
15	8.32(-3)
20	8.12(-3)
25	7.79(-3)
30	7.39(-3)

$$\begin{aligned} &\frac{dy}{dx} = x + y, \\ &y(0) = 0, \end{aligned} \tag{25}$$

with exact solution $y(x) = \exp(x) - x - 1$. Integrating both sides of relation (25) on interval $[0, x]$ and using initial condition result in

$$y(x) = y(0) + \int_0^x (s + y)ds = \frac{1}{2}x^2 + \int_0^x yds. \tag{26}$$

Now, using the new interpolation formula for y results in

$$\begin{aligned} &\sum_{i=0}^{n} \left(\frac{1}{2}\right) y(x_i) \left[\tanh(a_i x - b_i) - \tanh(a_{i+1}x - b_{i+1})\right] \\ &= \frac{1}{2}x^2 + \int_0^x \left[\sum_{i=0}^{n} \left(\frac{1}{2}\right) y(x_i) \left[\tanh(a_i s - b_i) - \tanh(a_{i+1}s - b_{i+1})\right]\right] ds. \end{aligned} \tag{27}$$

Let $Q_i(x) = \tanh(a_i x - b_i) - \tanh(a_{i+1}x - b_{i+1})$ for $i = 0, 1, 2, \ldots, n$. Then relation (27) is simplified as

$$\begin{aligned} \sum_{i=0}^{n} \left(\frac{1}{2}\right) y(x_i)Q_i(x) &= \frac{1}{2}x^2 + \int_0^x \left[\sum_{i=0}^{n} \left(\frac{1}{2}\right) y(x_i)Q_i(s)\right] ds \\ &= \frac{1}{2}x^2 + \left(\frac{1}{2}\right) \sum_{i=0}^{n} y(x_i) \int_0^x Q_i(s)ds. \end{aligned} \tag{28}$$

Finally, relation (28) is simplified as

$$\sum_{i=0}^{n} y(x_i) \left(Q_i(x) - \int_0^x Q_i(s)ds\right) = x^2. \tag{29}$$

By collocating (29) at x_i, $i = 1, 2, \ldots, n$, the unknown values of $y(x_i)$ are obtained. For example, the error function on $[0, 1]$ for $n = 10$, and $x_i = \dfrac{i}{10}$, $i = 1, 2, \ldots, 10$, is plotted in Figure 7. Also, the error function on $[0, 1]$ for $n = 20$ and $x_i = \dfrac{i}{20}$, $i = 1, 2, \ldots, 20$, is plotted in Figure 8.

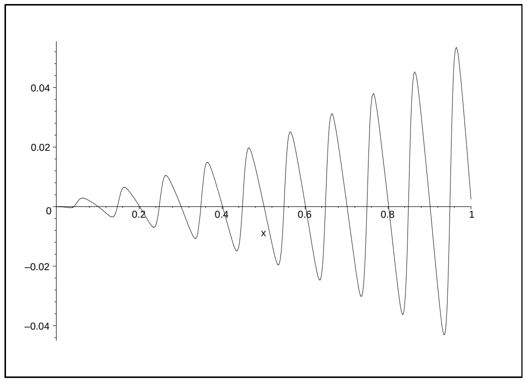


Figure 7: The error function of Example 5, with $n = 10$ in $[0, 1]$ and $a[i] = 100$ for $i = 0, 1, 2, \ldots, 10$.

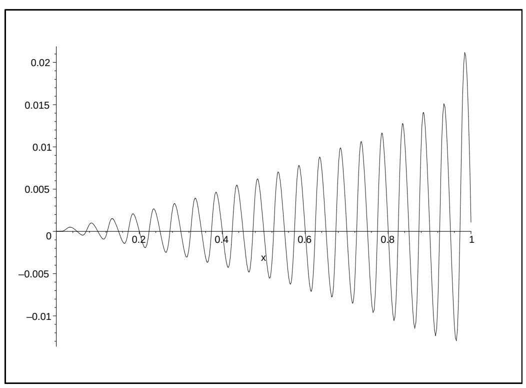


Figure 8: The error function of Example 5, with $n = 20$ in $[0, 1]$ and $a[i] = 100$ for $i = 0, 1, 2, \ldots, 20$.

Example 6 (Numerical integration for double integral)**.** Now using relation (15) results in

Table 3: Absolute error of quadrature formula (30) for $n = 5(5)30$ and $F(x,y) = \exp(x^2+y^2)$.

n	Absolute error
5	5.30(-2)
10	1.32(-2)
15	5.83(-3)
20	2.97(-3)
25	1.19(-3)
30	3.46(-4)

$$\int_a^b \int_c^d F(x,y)dydx \approx \int_a^b \int_c^d S(x,y)dydx \tag{30}$$

$$= \int_a^b \int_c^d \sum_{i=0}^{n} \sum_{i=0}^{m} \left(\frac{1}{2}\right)^2 z_{i,j} \times [\tanh(a_i x - b_i) - \tanh(a_{i+1}x - b_{i+1})] \times \left[\tanh(a_i^{'} y - b_i^{'}) - \tanh(a_{i+1}^{'} y - b_{i+1}^{'})\right] dxdy. \tag{31}$$

As an example, consider the following integral:

$$\int_0^1 \int_0^1 \exp(x^2+y^2)dydx \approx 2.139350130. \tag{32}$$

The absolute error of formula (30) to approximate the above integral, for $a_i = 100, i = 0, 1, \ldots, n$, is presented in Table 3.

4 Conclusions

In this article, by using Sigmoid functions, the new types of interpolation formulas were presented. To show the efficiency of the proposed new interpolation formulas, some applications in quadrature formulas (in both open and closed types), numerical integration for double integral, and numerical solution of ordinary differential equation were included. Numerical results were obtained by using the Hyperbolic tangent function, which can be easily changed by other Sigmoid functions.

Acknowledgements

Authors are grateful to two anonymous referees and editor-in-chief for their constructive comments.

References

1. Costarelli, D. and Kyurkchiev, N. *A Note on the smooth approximation to* $|x(1-x)\cdots(n-1-x)|$ *using Gaussian error function*, Commun. Appl. Anal. 25(1) (2021), 1–10.

2. Eslahchi, M.R., Masjed-Jamei, M. and Babolian, E. *On numerical improvement of Gauss-Lobatto quadrature rules*, Appl. Math. Comput. 164 (2005), 707–717.

3. Gautschi, W. *Numerical analysis*, Birkhauser, Boston, 1997.

4. Kyurkchiev, N., Iliev A. and Rahnev, A. *A look at the new logistic models with "polynomial variable transfer"*, LAP LAMBERT Academic Publishing, 2020.

5. Kyurkchiev, N. and Markov, S. *Sigmoid functions: Some approximation and modelling aspects*, LAP LAMBERT Academic Publishing, 2015.

6. Masjed-Jamei, M. and Dragomir, S.S. *A new generalization of the Ostrowski inequality and applications*, Filomat, 25 (2011), 115–123.

7. Masjed-Jamei, M., Jafari, M.A. and Srivastava, H.M. *Some applications of the Stirling numbers of the first and second kind*, J. Appl. Math. Comput. 47 (2015), 153–174.

8. Masjed-Jamei, M., Milovanovic, G.V. and Jafari, M.A. *Closed expressions for coefficients in weighted Newton-Cotes quadratures*, Filomat 27(4) (2013), 649–658.

9. Stoer, J. and Bulirsch, R. *Introduction to numerical analysis*, Translated from the German by R. Bartels, W. Gautschi and C. Witzgall. Second edition. Texts in Applied Mathematics, 12. Springer-Verlag, New York, 1993.

Aims and scope

Iranian Journal of Numerical Analysis and Optimization (IJNAO) is published twice a year by the Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad. Papers dealing with different aspects of numerical analysis and optimization, theories and their applications in engineering and industry are considered for publication.

Journal Policy

After receiving an article, the editorial committee will assign referees. Refereeing process can be followed via the web site of the Journal.
The manuscripts are accepted for review with the understanding that the work has not been published and also it is not under consideration for publication by any other journal. All submissions should be accompanied by a written declaration signed by the author(s) that the paper has not been published before and has not been submitted for consideration elsewhere.

Instruction for Authors

The Journal publishes all papers in the fields of numerical analysis and optimization. Articles must be written in English.
All submitted papers will be refereed and the authors may be asked to revise their manuscripts according to the referee's reports. The Editorial Board of the Journal keeps the right to accept or reject the papers for publication.
The papers with more than one authors, should determine the corresponding author. The e-mail address of the corresponding author must appear at the end of the manuscript or as a footnote of the first page.
It is strongly recommended to set up the manuscript by Latex or Tex, using the template provided in the web site of the Journal. Manuscripts should be typed double-spaced with wide margins to provide enough room for editorial remarks.
References should be arranged in alphabetical order by the surname of the first author as examples below:
[1] Stoer, J. and Bulirsch, R. *Introduction to Numerical Analysis*, Springer-verlag, New York, 2002.
[2] Brunner, H. *A survey of recent advances in the numerical treatment of Volterra integral and integro-differential equations*, J. Comput. Appl. Math. 8 (1982), 213-229.

Submission of Manuscripts

Authors may submit their manuscripts by the online submission via the website of the Journal at:

https://ijnao.um.ac.ir/about/submissions

Copyright Agreement

Upon the acceptance of an article by the Journal, the corresponding author will be asked to sign a ”Copyright Transfer Agreement” (see the web site) and send it to the Journal address. This will permit the publisher to publish and distribute the work.

Reviewers Acknowledgment

Peer review is the system used to assess the validity, quality and the originality of articles for publication in scholarly journals. We would like to sincerely thank the following reviewers who have taken part in the peer-review process for Iranian Journal of Numerical Analysis and Optimization during the last year.

Abbasbandy, Saeid
Al-mdallal, Qasem
Aminikhah, Hossein
Amintoosi, Mahmood
Area, Iván
Darehmiraki, Majid
Dolati, Ardeshir
Ebrahimnejad, Ali
Effati, Sohrab
Erfanian, Majid
Farhadinia, Bahram
Gachpazan, Mortaza
Gao, Wenwu
Ghanbari, Reza
Ghomanjani, Fatemeh
Ghorbani-moghadam, Khatere
Hashemi Borzabadi, Akbar
Hashemi Mehne, Sayed Hamed
Hadizadeh Yazdi, Mahmoud
Irandoust Pakchin, Safar
Karasözen, B.
Keyanpour, Mohammad
Li, Xiaolin
Lohmander, Peter
Mădălin-gabriel, CatanĂ
Marzban, Hamid Reza
Mesgarani, Hamid
Mirzaee, Farshid
Mirzaei, Davoud
Mokhtari, Reza
Namjoo, Mehran
Nazemi, Alireza
Pakdaman, Morteza
Pishbin, Saeed
Rezaee Pajand, Mohammad
Salahi, Maziar
Shirazian, Mohammad
Shirdel, Gholam Hassan
Shivanian, Elyas
Shokri, Ali
Soheili, Ali R.
Tohidi, Emran
Zaferanieh, Mehdi
Zamani, Moslem
Zeidabadi, Hamed

Iranian Journal of Numerical Analysis and Optimization

Former : MRJMS

CONTENTS

Vol. 11, No.1, pp 1-233, 2021

The rest of contents is on back of the page

web site : http://ijnao.um.ac.ir

Email : ijnao@um.ac.ir

ISSN-Print : 2423-6977

ISSN-Online : 2423-6969