Ferdowsi University of Mashhad

Iranian Journal of Numerical Analysis and Optimization

Volume 12, Number 3

Special Issue 2022

Serial Number: 23

Ferdowsi University of Mashhad, Iran

Call for Papers

Special Issue of
Iranian Journal of Numerical Analysis and Optimization

On Occasions of 75th Birthday of

Submit your Paper at
https://ijnao.um.ac.ir

Volume 12 , Number 2
Summer 2022
Serial Number: 22

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 12, Number 3 (Special Issue), 2022

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
- Zentralblatt
- ISC
- SID
- Civilica
- Magiran
- DOAJ
- OAJI
- AcademicKeys
- COPE
- Mendeley
- Academia.edu
- Linkedin

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

Iranian Journal of Numerical Analysis and Optimization

Volume 12, Number 3 (Special Issue), 2022

Ferdowsi University of Mashhad - Iran

Iranian Journal of Numerical Analysis and Optimization

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

* Full Professor

** Associate Professor

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.

The Iranian Journal of Numerical Analysis and Optimization is proud to publish a special issue on the occasion of their 75th birthday for two well-known colleagues of numerical linear algebra and optimal control in Iran. Professor Ali Vahidian Kamyad (born March 11, 1948, Mashhad, Iran) and Professor Faezeh Toutounian (born November 16, 1947, Mashhad, Iran) have been members of the editorial board since the beginning of this journal and played a very important role in improving the quality of IJNAO.

Professor Toutounian received her B. Sc. Mathematics from the Ferdowsi University of Mashhad (Iran) in 1970. She studied mathematics and statistic at the Pierre Marie Curie (Paris VI) University, France, and received her Master of Science degree in 1971 and her Ph.D. in 1975 under the direction of Professor Jean-Paul Benzecri.

Professor Vahidian received the B.Sc. degree from Ferdowsi University of Mashhad, Mashhad, Iran, in 1970, the M.Sc. degree from the Tehran Institute of Mathematics, Tehran, Iran, in 1973, and the Ph.D. degree from Leeds University, Leeds, U.K., in 1988, under supervision of J. E. Rubio. Since 1972, he has been at Ferdowsi University of Mashhad, where he is currently a Full Professor in the faculty of Mathematical sciences. His research interests include optimal control of distributed parameter systems and applications of fuzzy theory.

After approving the publication of a special issue and announcing the call for papers, all articles have been peer-refereed according to the scientific standards of the journal, and the accepted articles have been published under this issue. In the end, I wish good health, success and happiness to Professor Toutounian and Professor Vahidian.

Ali R. Soheili

Editor-in-Chief

Short Biography: Professor Faezeh Toutounian

Faezeh Toutounian was born on November 15, 1947 in Mashhad, Iran. She received her B. Sc. Mathematics from the Ferdowsi University of Mashhad (Iran) in 1970. She studied mathematics and statistic at the Pierre Marie Curie (Paris VI) University, France, and received her Master of Science degree in 1971 and her Ph.D. in 1975 under the direction of Professor Jean-Paul Benzecri.

After finishing her studies at the Pierre Marie Curie (Paris VI) University, Faezeh joined the Department of Mathematics at Ferdowsi university of Mashhad, Iran in 1976 and became a full professor there in 1998. When she arrived at Ferdowsi university, the Department of Mathematics was small and in early stage of development. Due to his personality, research, and teaching, Toutounian became one of the dominant faculty members, and his role in the development of the department to its present state has been substantial.

After joining the Ferdowsi university, Faezeh visited France two times, each time for an academic year (1985-1986, 1995-1996).

Professor Toutounian is an excellent and devoted teacher. Over the years, she has given a variety of courses in linear programming, numerical analysis, numerical linear algebra, iterative methods for linear systems. Undoubtedly, her courses stimulated some students to choose numerical analysis or numerical linear algebra as their future main research area. She has translated 8 English books into Persian language for her courses. F. Toutounian mentored 22 Ph.D. students many of whom have had distinguished careers of their own. She has published 75 articles and given more than 20 invited seminars and conference talks. Faezeh was Head of the Department of Mathematics from 1998–2000. In 2015 was awarded the price of Numerical Linear Algebra of professor Rajabalipoor (One of the prizes of Iranian Society of Mathematics). She is an Editorial board of Iranian Journal of Numerical Analysis and Optimization and Journal of Mathematical Modeling from 2010– now and a reviewer for Mathematical Reviews since 2019. Toutounian served on a number of committees of the Ferdowsi university of Mashhad and was a member of the scientific advisory committee for some 20 international conferences.

Reaching the compulsory retirement age, Professor Toutounian retired from the Ferdowsi University in October 2013. Nevertheless, she continues to be partially active in teaching and vigorously engaged in research.

Short Biography: Professor Ali Vahidian Kamyad

Ali Vahidian Kamyad was born in 1948, in Mashhad, and passed elementary school education at Dyanat School. He finished his high school education at Ebne Yamin high school in Mashhad. In 1967, he began undergraduate studies in mathematics at Ferdowsi University of Mashhad and got B.Sc. degree in 1971. Ali began his postgraduate studies at the Tehran Institute of mathematics under of management of Prof. Mosaheb, and in 1973, got MPhild degree in mathematics. Professor Fatemi, the head of the mathematics department of Ferdowsi University, invited him to be a lecturer to teach undergraduate mathematics courses at the Ferdowsi University of Mashhad. In 1985, he got a scholarship to continue his postgraduate studies at Leeds University in England on control theory and optimization under the supervision of Dr. Rubio. After 28 months from the start of Mr. Vahidian's doctoral studies at the University of Leeds, the results of his research were accepted by his supervisor and the University of Leeds. Although the minimum period of studying for a doctorate in England is three years, but Dr. Ali Vahidian Kamyad was able to defend his thesis and obtain a Ph.D. degree after 28 months.When Ali got his Ph.D. degree, Dr. Rubio suggested that he continue his research work with Prof. Prichard as a lecturer at Warwick University. But, Dr. Vahidian Preferred to return to Iran and continue his academic job at the Ferdowsi University of Mashhad. In 1988, Dr. Vahidian started his academic job as an assistant professor in applied mathematics. In 1993, Dr. Vahidian continued teaching and research work as associate professor, and in 1997 he continued his research works and teaching as professor and supervising many Master's and Ph.D. students at Ferdowsi University in applied mathematics and control engineering as a joint academic member in the department of power engineering of Ferdowsi University. Professor Vahidian published many papers on control theory, optimization, industrial mathematics, and medicine mathematics and also completed much industrial research in various factories in Iran, especially in Mashhad, and also supervised many MSc and Ph.D. students in applied mathematics and control engineering and economy and management and agricultural studies. Now many MSc and Ph.D. students under his supervision are academic members in the many universities in Iran and abroad.

Contents

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 497–512
DOI:10.22067/ijnao.2022.73876.1079
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Estimation of the regression function by Legendre wavelets

M. Hamzehnejad* , M.M. Hosseini and A. Salemi

Abstract

We estimate a function f with N independent observations by using Legendre wavelets operational matrices. The function f is approximated with the solution of a special minimization problem. We introduce an explicit expression for the penalty term by Legendre wavelets operational matrices. Also, we obtain a new upper bound on the approximation error of a differentiable function f using the partial sums of the Legendre wavelets. The validity and ability of these operational matrices are shown by several examples of real-world problems with some constraints. An accurate approximation of the regression function is obtained by the Legendre wavelets estimator. Furthermore, the proposed estimation is compared with a nonparametric regression algorithm and the capability of this estimation is illustrated.

AMS subject classifications (2020): 65T60; 41A30; 65D10; 62G08.

Keywords: Legendre wavelet; Operational matrix; Wavelet approximation; Regression function; Error analysis.

---

* Corresponding author

Received 29 November 2021; revised 5 March 2022; accepted 9 March 2022

Mehdi Hamzehnejad
Department of Mathematic, Graduate University of Advanced Technology, Kerman, Iran. e-mail: mhdhamzehnejad@gmail.com

Mohammad Mehdi Hosseini
Department of Applied Mathematics and Mahani Mathematical Research Center, Shahid Bahonar University of Kerman, Kerman, Iran. e-mail: Hossem25@gmail.com

Abbas Salemi
Department of Applied Mathematics and Mahani Mathematical Research Center, Shahid Bahonar University of Kerman, Kerman, Iran. e-mail: Salemi@uk.ac.ir

1 Introduction

Let $f : [a, b] \longrightarrow \mathbb{R}$ with independent observations $\{(x_i, y_i), i = 1, \ldots, N\}$. Consider the following nonparametric regression model to provide an estimate for f:

$$y_i = f(x_i) + \epsilon_i, \tag{1}$$

where $x_i \in [a, b]$ and ϵ_i have Gaussian noise. It is well known that the following optimization problem approximate the regression function f [7, 13]:

$$\min_{f \in \mathbf{S}} \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i))^2 + \frac{\lambda}{N} \int_a^b (f^{(r)}(x))^2 dx, \tag{2}$$

where $\mathbf{S}$ denotes the set of functions f satisfying the constraints and the constant λ is called smoothing parameter. The first term measures closeness to the data, while the second term penalizes curvature in the function. This optimization problem appears in many branches of applied mathematics including economics, stochastic processes, statistics, machine learning, and control theory, and several studies have been conducted to determine the function f [7, 9, 18, 5, 13].

Using linear combinations of basis functions, such as orthogonal polynomials, wavelets, and splines is a popular approach to estimating the function f [7, 18, 5, 17, 11, 3, 6, 16]. This kind of method can be expressed as a matrix equation that contains a penalty term. Although it is not possible to get a clear and accurate answer to this problem, it is necessary to use approximate methods to solve it. Calculating the penalty term is an important issue for the authors. Wand and Ormerod [18] obtained an exact explicit expression for each entry of the penalty matrix by solving numerical integrals.

It is well known that a single method cannot work for all functions without any restrictions. Some of these restrictions include monotonicity, convexity, unimodality, or combinations of several types of constraints. For example, Mammen et al. [8] considered the regression function under the monotonicity constraint and Meyer [9] considered the regression function under constraints of convexity and monotone. Also in [1, 12], the authors considered the regression function under combinations of several types of restrictions.

In this paper, by using properties of the Legendre wavelets, we provide an exact explicit expression for the penalty term only by matrix multiplications, which reduce the complexity of the problem. Also, an accurate approximation of differentiable functions is obtained by Legendre wavelets. For this purpose, we provide an upper bound for the first term of (2). Moreover, by using the examples that have been mentioned in [9, 1, 4], we show that the Legendre wavelets are a good candidate for the estimation of regression functions under various constraints.

The rest of this paper is organized as follows. In Section 2, we state some definitions and properties of the Legendre wavelets. Furthermore, we recall

the operational matrix of derivatives, and by using this operational matrix, we provide an exact explicit expression for the penalty matrix. In Section 2, a new upper bound on the approximation error of the partial sums of the Legendre wavelets is presented. In Section 3, the performance of the proposed estimation is compared with a nonparametric regression method, by numerical examples.

2 Legendre polynomials and wavelets

In this section, we study Legendre polynomials and wavelets by presenting some necessary definitions and theorems. The well-known Legendre polynomials are defined on the interval $[-1,1]$ and can be determined by the following recurrence formulas [15].

$$(m+1)L_{m+1}(x) = (2m+1)xL_m(x) - mL_{m-1}(x), \quad m = 1,2,3,\ldots,$$

where $L_0(t) = 1$ and $L_1(x) = x$. The following relation is hold for Legendre polynomials [15, eq. 3.176a]

$$L_m(x) = \frac{1}{2m+1}\left(L'_{m+1}(x) - L'_{m-1}(x)\right). \tag{3}$$

Moreover, we have the following uniform bound for Legendre polynomials [15]

$$|L_m(x)| \leq 1, \quad x \in [-1,1],\ m \geq 0. \tag{4}$$

Legendre wavelets are defined on the interval $[0,1]$ as follows:

$$\psi_{n,m}(t) = \begin{cases} \sqrt{(m+\frac{1}{2})}2^{\frac{k+1}{2}} L_m(2^{k+1}t - (2n+1)), & \frac{n}{2^k} \leq t < \frac{n+1}{2^k}, \\ 0, & \text{otherwise}, \end{cases}$$

where $k \in \mathbb{N}$, $m = 0,1,\ldots,M-1$, and $n = 0,1,\ldots,2^k-1$. The Legendre wavelets are an orthonormal basis for $L^2[0,1]$ and the following orthogonality holds:

$$\int_0^1 \psi_{m,n}(t)\psi_{r,s}(t)dt = \delta_{mr}\delta_{ns}.$$

Let $f(t) \in L^2[0,1]$. Then

$$f(t) \simeq \sum_{n=0}^{2^k-1}\sum_{m=0}^{M-1} c_{n,m}\psi_{n,m}(t) = C^T\Psi(t),$$

where $c_{n,m} = \int_0^1 f(t)\psi_{n,m}(t)dt$. The vectors C and $\Psi(t)$ are $2^kM \times 1$ vectors given by

$$C = \left[c_{0,0}, \ldots, c_{0,M-1}, c_{1,0}, \ldots, c_{1,M-1}, \ldots, c_{2^k-1,0}, \ldots, c_{2^k-1,M-1}\right]^T,$$

$$\Psi(t) = [\psi_{0,0}(t), \ldots, \psi_{0,M-1}(t), \psi_{1,0}(t), \ldots, \psi_{1,M-1}(t), \ldots, \psi_{2^k-1,0}(t), \ldots, \psi_{2^k-1,M-1}(t)]^T.$$

The Legendre wavelets approximation finds a shape constrained f to the minimization problem (2). In the minimization problem (2), we set

$$f(t) \simeq \sum_{n=0}^{2^k-1} \sum_{m=0}^{M-1} c_{n,m}\psi_{n,m}(t).$$

For simplicity, we can set$\psi_{((i-1)\times M)+j+1}(t) := \psi_{i,j}(t)$ and $c_{((i-1)\times M)+j+1} := c_{ij}$ for $i = 1, \ldots, 2^k$ and $j = 0, \ldots, M-1$. Hence the following vectors are obtained:

$$\Psi(t) = [\psi_1(t), \ldots, \psi_{2^k M}(t)]^T, \quad C = [c_1, c_2, \ldots, c_{2^k M}]^T. \tag{5}$$

Therefore, we have

$$f(t) = \sum_{j=1}^{2^k M} c_j \psi_j(t),$$

where $\psi_j(t)$ are the Legendre wavelets. Therefore the objective function to minimize (2) is the following penalized least square:

$$\min_{c_j} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{2^k M} c_j \psi_j(x_i) \right)^2 + \frac{\lambda}{N} \int_0^1 \left(\sum_{j=1}^{2^k M} c_j \psi_j^{(r)}(t) \right)^2 dt,$$

where

$$\int_0^1 \left(\sum_{j=1}^{2^k M} c_j \psi_j^{(r)}(t) \right)^2 dt = \sum_{i=1}^{2^k M} \sum_{j=1}^{2^k M} c_i c_j \int_0^1 \psi_i^{(r)}(t)\psi_j^{(r)}(t)dt.$$

Suppose that V is a matrix by elements of the form $V_{ij} := \frac{1}{N} \sum_{l=1}^{N} \psi_i(x_l)\psi_j(x_l)$, that P is a matrix by elements $P_{ij} = \int_0^1 \psi_i^{(r)}(t)\psi_j^{(r)}(t)dt$, and that the elements of vector b are defined by $b_i = \frac{1}{N} \sum_{l=1}^{N} \psi_i(x_l)y_l,\ i, j = 1, \ldots, 2^{k-1}M$, so the minimization problem (2) has the following quadratic form of minimization [5]:

$$\min_{C\in\mathbb{R}^{2^k M}} \frac{1}{2}C^T VC - bC + \lambda(\frac{1}{2}C^T PC). \tag{6}$$

By taking the derivative of (6) in terms of C and put it equal zero, we obtain the following equation:

$$(V + \lambda P)C = b. \tag{7}$$

Now focus on the second term, to determine an appropriate operator matrix to solve the problem (2). An important issue is to calculate the elements of the matrix P. We use Legendre wavelets operational matrix of derivative, to get the new structure of the matrix P. The following theorems determine the Legendre wavelet operational matrices of derivatives, which are used to solve differential equations.

Theorem 1. [10, Theorem 1] Let $\Psi(t)$ be the Legendre wavelets vector as in (5). Then the derivative of the vector $\Psi(t)$ can be expressed by

$$\frac{d\Psi(t)}{dt} = D\Psi(t),$$

where D is the $2^k M$ operational matrix

$$D = \begin{bmatrix} F & 0 & \cdots & 0 \\ 0 & F & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F \end{bmatrix},$$

where F is an $M \times M$ matrix such that (r, s)th entry of F is defined as follows:

$$F_{r,s} = \begin{cases} 2^{k+1}\sqrt{(2r-1)(2s-1)}, & \begin{cases} r = 2, \ldots, M, \\ s = 1, \ldots, r-1, \end{cases} \quad (r+s) \text{ odd}, \\ 0, & \text{otherwise}. \end{cases}$$

Theorem 2. [10, Theorem 2] By using Theorem 1, the operational matrix for nth derivative can be derived as

$$\frac{d^n\Psi(t)}{dt^n} = D^n\Psi(t),$$

where D^n is the nth power of the matrix D.

Therefore, using these operational matrices, the elements of the matrix P in (7) are introduced in the next theorem.

Theorem 3. Let $\Psi(t)$ be the Legendre wavelets vector defined in (5). Assume that r is a nonnegative integer and that the elements of the matrix $P = [P_{ij}]$ are $P_{ij} = \int_0^1 \psi_i^{(r)}(t)\psi_j^{(r)}(t)dt$. Then P_{ij} has the following exact explicit expression

$$P_{ij} = (D_i^r)(D_j^r)^T, \quad i,j = 1,\ldots,2^k M, \tag{8}$$

where D_i^r is the ith row of the operational matrix D^r as in Theorem 2.

Proof. By using Theorem 2, the elements of the matrix P are as follows:

$$P_{ij} = \int_0^1 \psi_i^{(r)}(t)\psi_j^{(r)}(t)dt = \int_0^1 (D_i^r\Psi(t))(D_j^r\Psi(t))dt, \quad i,j = 1,\ldots,2^k M. \tag{9}$$

Let $D_i^r\Psi(t) = \sum_{s=1}^{2^k M} d_{is}^{(r)}\psi_s(t)$. Then

$$\begin{aligned} P_{ij} &= \int_0^1 \left(d_{i1}^{(r)}\psi_1(t) + \cdots + d_{i2^kM}^{(r)}\psi_{2^kM}(t)\right)\left(d_{j1}^{(r)}\psi_1(t) + \cdots + d_{j2^kM}^{(r)}\psi_{2^kM}(t)\right)dt \\ &= \int_0^1 \sum_{s=1}^{2^kM}\sum_{l=1}^{2^kM} d_{is}^{(r)}d_{jl}^{(r)}\psi_s(t)\psi_l(t)dt = \sum_{s=1}^{2^kM}\sum_{l=1}^{2^kM} d_{is}^{(r)}d_{jl}^{(r)}\int_0^1 \psi_s(t)\psi_l(t)dt. \end{aligned}$$

According to the property of orthogonality, we have

$$\int_0^1 \psi_s(t)\psi_l(t)dt = \delta_{sl}. \tag{10}$$

By using (10), $P_{ij} = \sum_{s=1}^{2^kM} d_{is}^{(r)}d_{js}^{(r)} = (D_i^r)(D_j^r)^T$. □

Therefore, we can calculate the elements of the matrix P only by a matrix multiplication. By solving system (7), the appropriate weight coefficients are obtained to approximate the function f.

3 Error analysis

In this section, we present an error estimate of the partial sums of Legendre wavelets to the regression function f. For this purpose, we benefit from the well-known mean-square error (MSE). By using the MSE [16], we measure the performance of the estimator $\hat{f}$ as follows:

$$MSE(\hat{f},f) = \frac{1}{N}\sum_{i=1}^{N} E\left[\hat{f}(x_i) - f(x_i)\right]^2.$$

The Legendre wavelets estimator $\hat{f}$ can be written as

$$\hat{f} = (\hat{f}(x_1),\ldots,\hat{f}(x_N)) = \left(\sum_{n=0}^{2^k-1}\sum_{m=0}^{M-1} c_{n,m}\psi_{n,m}(x_1),\ldots,\sum_{n=0}^{2^k-1}\sum_{m=0}^{M-1} c_{n,m}\psi_{n,m}(x_N)\right).$$

We present a new approximation error of the function f, using the partial sums of Legendre wavelets. We know that

$$\begin{aligned} f(t) &= \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} c_{n,m}\psi_{n,m}(t) \\ &= \sum_{n=0}^{2^k-1}\sum_{m=0}^{M-1} c_{n,m}\psi_{n,m}(t) + \sum_{n=0}^{2^k-1}\sum_{m=M}^{\infty} c_{n,m}\psi_{n,m}(t) + \sum_{n=2^k}^{\infty}\sum_{m=0}^{\infty} c_{n,m}\psi_{n,m}(t). \end{aligned} \tag{11}$$

The last part in (11), $\sum_{n=2^k}^{\infty}\sum_{m=0}^{\infty} c_{n,m}\psi_{n,m}(t) = 0$, because the Legendre wavelets $\psi_{n,m}(t)$ are zero outside of the interval $[0,1]$. Then

$$\begin{aligned} \left\| f(t) - \sum_{n=0}^{2^k-1}\sum_{m=0}^{M-1} c_{n,m}\psi_{n,m}(t) \right\|^2 &= \left\| \sum_{n=0}^{2^k-1}\sum_{m=M}^{\infty} c_{n,m}\psi_{n,m}(t) \right\|^2 \\ &\leq \sum_{n=0}^{2^k-1}\sum_{m=M}^{\infty} |c_{n,m}|^2 \left\|\psi_{n,m}(t)\right\|^2. \end{aligned}$$

We know that $\left\|\psi_{n,m}(t)\right\|^2 = 1$. Therefore

$$\left\| f(t) - \hat{f}(t) \right\|^2 = \left\| f(t) - \sum_{n=0}^{2^k-1}\sum_{m=0}^{M-1} c_{n,m}\psi_{n,m}(t) \right\|^2 \leq \sum_{n=0}^{2^k-1}\sum_{m=M}^{\infty} |c_{n,m}|^2. \tag{12}$$

Hence, the approximation error of the truncated series of Legendre wavelets depends on the Legendre wavelets coefficients $c_{n,m}$. Now, we obtain an upper bound for Legendre wavelets coefficients.

Theorem 4. Suppose that $k \in \mathbb{N}$ and that $f, f', \ldots, f^{(r)}$ are absolutely continuous on $[0,1]$. Suppose that $V = \max\left\{V_n,\ n = 0, \ldots, 2^k - 1\right\}$, where

$$V_n = \int_{\frac{n}{2^k}}^{\frac{n+1}{2^k}} \left| f^{(r+1)}(t) \right| dt, \qquad n = 0, 1, \ldots, 2^k - 1.$$

Then for $m \geq r+1$,

$$|c_{n,m}| \leq \begin{cases} \frac{V}{2^{rk}(2m-2r+3)\cdots(2m-1)(2m+3)\cdots(2m+2r-1)\sqrt{2^k(2m-2r+1)}}, & r \text{ odd}, \\ \frac{V}{2^{rk}(2m-2r+3)\cdots(2m+1)(2m+5)\cdots(2m+2r-1)\sqrt{2^k(2m-2r+1)}}, & r \text{ even}. \end{cases} \tag{13}$$

Proof. For each $0 \leq i \leq r$, define the following sequence

$$c_{n,m}^{(i)} = \int_{\frac{n}{2^k}}^{\frac{n+1}{2^k}} f^{(i)}(t)\psi_{n,m}(t)dt$$
$$= \sqrt{(m+\frac{1}{2})}2^{\frac{k+1}{2}} \int_{\frac{n}{2^k}}^{\frac{n+1}{2^k}} f^{(i)}(t)L_m(2^{k+1}t-(2n+1))dt, \qquad (14)$$

where $c_{n,m}^{(0)} = c_{n,m}$. Let $x = 2^{k+1}t - (2n+1)$. Then

$$c_{n,m}^{(r+1)} = \sqrt{(m+\frac{1}{2})}2^{\frac{k+1}{2}} \int_{-1}^{1} f^{(r+1)}\left(\frac{x+2n+1}{2^{k+1}}\right) L_m(x)\frac{dx}{2^{k+1}}$$
$$= \frac{\sqrt{(m+\frac{1}{2})}}{2^{\frac{k+1}{2}}} \int_{-1}^{1} f^{(r+1)}\left(\frac{x+2n+1}{2^{k+1}}\right) L_m(x)dx. \qquad (15)$$

By using the equation (3), we have

$$c_{n,m}^{(r)} = \frac{\sqrt{(m+\frac{1}{2})}}{2^{\frac{k+1}{2}}(2m+1)} \int_{-1}^{1} f^{(r)}\left(\frac{x+2n+1}{2^{k+1}}\right)\left(L'_{m+1}(x) - L'_{m-1}(x)\right)dx. \qquad (16)$$

Using integration by parts, we have

$$c_{n,m}^{(r)} = \frac{\sqrt{(m+\frac{1}{2})}}{2^{\frac{k+1}{2}}(2m+1)}\left[f^{(r)}\left(\frac{x+2n+1}{2^{k+1}}\right)\left(L_{m+1}(x) - L_{m-1}(x)\right)\right]_{-1}^{1}$$
$$+ \frac{\sqrt{(m+\frac{1}{2})}}{2^{\frac{k+1}{2}}2^{k+1}(2m+1)} \int_{-1}^{1} f^{(r+1)}\left(\frac{x+2n+1}{2^{k+1}}\right)\left(L_{m+1}(x) - L_{m-1}(x)\right)dx. \qquad (17)$$

Using the properties $L_m(1) = 1^m$ and $L_m(-1) = (-1)^m$ for $m \geq 0$, easy computations shows that the first term of (17) vanishes. Thus we have

$$c_{n,m}^{(r)} = \frac{\sqrt{(m+\frac{1}{2})}}{2^{\frac{k+1}{2}}2^{k+1}(2m+1)} \int_{-1}^{1} f^{(r+1)}\left(\frac{x+2n+1}{2^{k+1}}\right)\left(L_{m+1}(x) - L_{m-1}(x)\right)dx. \qquad (18)$$

From (14) and (18), we obtain the following relation between the coefficients

$$c_{n,m}^{(r)} = \frac{1}{2^{k+1}(2m+1)}\left(c_{n,m-1}^{(r+1)} - c_{n,m+1}^{(r+1)}\right). \qquad (19)$$

Now, we obtain an upper bound for $c_{n,m}^{(r+1)}$. We can see that

$$c_{n,m}^{(r+1)} = \frac{\sqrt{(m+\frac{1}{2})}}{2^{\frac{k+1}{2}}} \int_{-1}^{1} f^{(r+1)} \left(\frac{x+2n+1}{2^{k+1}} \right) L_m(x)dx$$
$$= \sqrt{(m+\frac{1}{2})2^{k+1}} \int_{\frac{n}{2^k}}^{\frac{n+1}{2^k}} f^{(r+1)}(t) L_m(2^{k+1}t-(2n+1))dt.$$

From (9) and by easy computation, we obtain

$$\left|c_{n,m}^{(r+1)}\right| = \sqrt{(m+\frac{1}{2})2^{k+1}} \int_{\frac{n}{2^k}}^{\frac{n+1}{2^k}} \left|f^{(r+1)}(t)\right| \left|L_m(2^{k+1}t-(2n+1))\right| dt$$
$$\leq \sqrt{(2m+1)2^k} \int_{\frac{n}{2^k}}^{\frac{n+1}{2^k}} \left|f^{(r+1)}(t)\right| dt \leq V\sqrt{2^k(2m+1)}. \tag{20}$$

Applying (20) in (19), we have

$$\left|c_{n,m}^{(r)}\right| \leq \frac{1}{2^{k+1}(2m+1)} \left(\left|c_{n,m-1}^{(r+1)}\right| + \left|c_{n,m+1}^{(r+1)}\right| \right)$$
$$\leq \frac{V\sqrt{2^k(2m-1)} + V\sqrt{2^k(2m+3)}}{2^{k+1}(2m+1)}. \tag{21}$$

Since

$$\sqrt{2m-1} + \sqrt{2m+3} \leq 2\sqrt{2m+1},$$

(21) becomes to

$$\left|c_{n,m}^{(r)}\right| \leq \frac{2V\sqrt{2^k(2m+1)}}{2^{(k+1)}(2m+1)} = \frac{V}{\sqrt{2^k(2m+1)}}. \tag{22}$$

Also, by using (22) in (19), we obtain the following upper bound for $c_{n,m}^{(r-1)}$:

$$\left|c_{n,m}^{(r-1)}\right| \leq \frac{1}{2^{k+1}(2m+1)} \left(\left|c_{n,m-1}^{(r)}\right| + \left|c_{n,m+1}^{(r)}\right| \right)$$
$$\leq \frac{1}{2^{k+1}(2m+1)} \left(\frac{V}{\sqrt{2^k(2m-1)}} + \frac{V}{\sqrt{2^k(2m+3)}} \right)$$
$$= \frac{V}{2^{k+1}(2m+1)\sqrt{2^k}} \left(\frac{\sqrt{(2m+3)} + \sqrt{(2m-1)}}{\sqrt{(2m-1)(2m+3)}} \right)$$
$$\leq \frac{2V\sqrt{(2m+3)}}{2^{k+1}(2m+1)\sqrt{2^k(2m-1)(2m+3)}}$$
$$= \frac{V}{2^k(2m+1)\sqrt{2^k(2m-1)}}.$$

If we continue the above process, then by easy computation for an integer $s \geq 2$, we obtain the following upper bound for $c_{n,m}^{(r-s-1)}$:

$$|c_{n,m}^{(r-s)}| \leq \begin{cases} \frac{V}{2^{(s-1)k}(2m-2s+5)\cdots(2m-1)(2m+3)\cdots(2m+2s-3)\sqrt{2^k(2m-2s+3)}}, & s \text{ odd}, \\ \frac{V}{2^{(s-1)k}(2m-2s+5)\cdots(2m+1)(2m+5)\cdots(2m+2s-3)\sqrt{2^k(2m-2s+3)}}, & s \text{ even}. \end{cases}$$

Then (13) holds when $s+1=r$. □

Now, we are ready to provide an approximation error of the partial sums of Legendre wavelets. We show that if the regression function f is smooth, then the partial sums of Legendre wavelets converge to it rapidly.

Theorem 5. Suppose that $k \in \mathbb{N}$ and that $f, f', \ldots, f^{(r)}$ are absolutely continuous on $[0,1]$. Moreover, suppose that $E_{k,M}(f(t)) = \left\| f(t) - \hat{f}(t) \right\|$. Then for $M \geq r+1$ and $r \geq 1$,

$$E_{k,M}(f(t)) \leq \begin{cases} \frac{V}{r2^{(r-1)k}(2M-2r+1)\cdots(2M-1)(2M+3)\cdots(2M+2r-7)\sqrt{2^k(2M-2r+1)}}, & r \text{ odd}, \\ \frac{V}{r2^{(r-1)k}(2M-2r+1)\cdots(2M+1)(2M+5)\cdots(2M+2r-7)\sqrt{2^k(2M-2r+1)}}, & r \text{ even}. \end{cases}$$

Proof. Let r be an odd integer. Applying (13) in (12), we obtain

$$\begin{aligned} & E_{k,M}(f(t)) \\ & \leq \sum_{n=0}^{2^k-1} \sum_{m=M}^{\infty} \frac{V}{2^{rk}(2m-2r+3)\cdots(2m-1)(2m+3)\cdots(2m+2r-1)\sqrt{2^k(2m-2r+1)}} \\ & \leq \frac{V}{2^{rk}\sqrt{2^k(2M-2r+1)}} \sum_{n=0}^{2^k-1} \sum_{m=M}^{\infty} \frac{1}{(2m-2r+3)\cdots(2m-1)(2m+3)\cdots(2m+2r-1)} \\ & = \frac{V}{2^{rk}\sqrt{2^k(2M-2r+1)}} \sum_{n=0}^{2^k-1} \sum_{m=M}^{\infty} \frac{1}{(2m+2r-1)^{r-1}(1-\frac{4r-4}{(2m+2r-1)})\cdots(1-\frac{4}{(2m+2r-1)})} \\ & \leq \frac{V}{2^{rk}\sqrt{2^k(2M-2r+1)}(1-\frac{4r-4}{(2M+2r-3)})\cdots(1-\frac{4}{(2M+2r-3)})} \\ & \quad \sum_{n=0}^{2^k-1} \int_{M-1}^{\infty} \frac{1}{(2x+2r-1)^{r-1}} dx \\ & = \frac{2^k V}{r2^{rk}(2M-2r+1)\cdots(2M-1)(2M+3)\cdots(2M+2r-7)\sqrt{2^k(2M-2r+1)}} \\ & = \frac{V}{r2^{(r-1)k}(2M-2r+1)\cdots(2M-1)(2M+3)\cdots(2M+2r-7)\sqrt{2^k(2M-2r+1)}} \end{aligned} \tag{23}$$

By a similar approach, the results hold for an even integer r and complete the proof. □

Remark 1. The aim of this remark is to draw an approximation error for a function $f(x)$, using the partial sums of the Legendre wavelets. Consider two functions $f(x) = 1 + x - 0.45\exp[-5(x-0.5)^2]$ and $f(x) = \frac{1}{6}x^2|x|$. The

function $f(x) = 1 + x - 0.45\exp[-5(x-0.5)^2]$ is infinitely differentiable. In Table 1, numerical results are shown for this function for some values of M, k, and r. The numerical results obtained from this table indicate that by increasing M, k, and r, the partial sums of Legendre wavelets converge to the function $f(x)$ rapidly. Also, consider the function $f(x) = \frac{1}{6}x^2|x|$

Table 1: Approximation errors of the function $f(x) = 1 + x - 0.45\exp[-5(x-0.5)^2]$ evaluated by Theorem 5.

M	k	r	$E_{k,M}(f(x))$	M	k	r	$E_{k,M}(f(x))$
10	1	3	1.920×10^{-3}	10	1	5	6.977×10^{-5}
15	2	3	5.669×10^{-5}	15	2	5	1.686×10^{-7}
20	3	3	3.373×10^{-6}	20	3	5	8.670×10^{-10}

[19]. This function and its derivatives are absolutely continuous on $[0, 1]$ and $f^{(2)}(x) = |x|$. Also, $f^{(3)}(x) = 2H(x) - 1$, where $H(x)$ is the Heaviside step function, which is of bounded variation and $f^{(4)}(x) = 2\delta(x)$, where $\delta(x)$ is the Dirac delta function. In Table 55, the numerical results are listed for some values of M, k, and r. Moreover, the logarithm of absolute errors is displayed

Table 2: Approximation errors of the function $f(x) = \frac{1}{6}x^2|x|$ evaluated by Theorem 5.

M	k	r	$E_{k,M}(f(x))$	M	k	r	$E_{k,M}(f(x))$
10	1	3	1.067×10^{-4}	10	2	3	1.887×10^{-5}
15	1	3	3.251×10^{-5}	15	2	3	5.747×10^{-6}
20	1	3	1.459×10^{-5}	20	2	3	2.579×10^{-6}

in Figure 1.

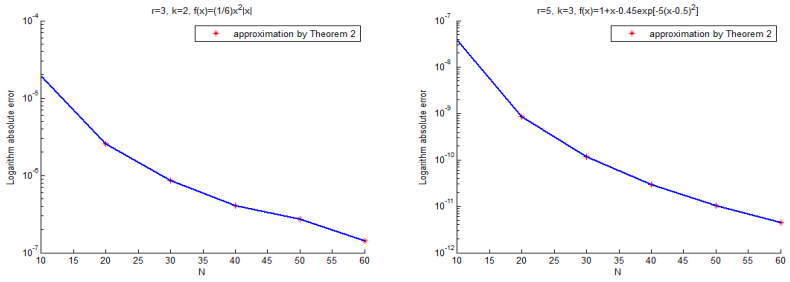


Figure 1: Approximation error of the functions $f(x) = \frac{1}{6}x^2|x|$ and $f(x) = 1 + x - 0.45\exp[-5(x-0.5)^2]$.

4 Numerical results

In this section, we present some examples to illustrate the validity and ability of the Legendre wavelets. For this purpose, we use some real-world test functions. Suppose that $(x_i, y_i), i = 1, \ldots, N$ are N independent data with the same distribution such that $X_i, i = 1, \ldots, N$ have normal distribution, that is, $x_i \sim N(\mu, \sigma)$. Let $y_i = f(x_i) + \epsilon_i$ and let x_i, ϵ_i, f be independent with penalization order $r = 2$. We consider different kinds of regression functions, which have different constraints on interval $[0, 1]$.

Remark 2. Choosing the suitable smoothing parameter λ is also an important issue in solving the minimization problem (2). Corlay [5] showed that $\lambda = \sigma_{x_i}^{2r-1}$ is a suitable smoothing parameter, where the quantity σ_{x_i} is the standard deviation, which scales proportionally with x_i. Hence, in all examples, the coefficient of the penalty term $\frac{\lambda}{N} = \frac{\sigma_{x_i}^{2r-1}}{N}$ is used.

Example 1. Consider two real regression functions $f_1(x) = 15(x - 0.25)^2$ [9] and $f_2(x) = 1 + x - 0.45\exp[-5(x - 0.5)^2]$ [4]. Then $f_1(x)$ is convex over $[0, 1]$ and $f_2(x)$ is strictly monotone over $[0, 1]$. Penalized Legendre wavelets regression of samples are plotted in Figure 2.

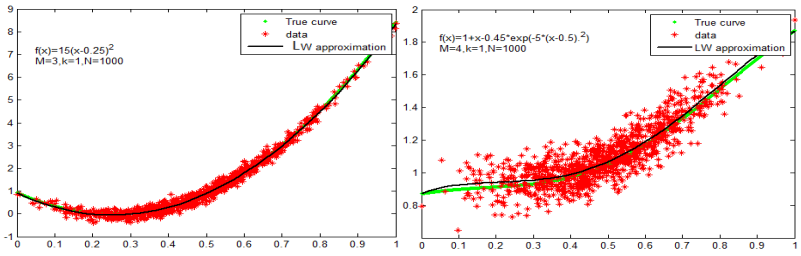


Figure 2: Approximate solution for the regression functions $f_1(x)$ and $f_2(x)$ in Example 1

Example 2. Consider the real regression function $f_3(x) = 15x^2 \sin(3.7x) + \frac{2}{\sigma\sqrt{2\pi}} \exp[(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2)]$ [7, 1], where $\sigma = 0.1$ and $\mu = 0.3$. This function is unimodal (first increasing and then decreasing), concave on $[0.55, 1]$, and twice differentiable. We approximate the minimization problem (2) for $N = 1000$ samples of (x_i, y_i). In Figure 3, the numerical results are shown.

Example 3. Consider the real regression function $f_4(x) = 10(x - 0.5)^3 - \exp[-100(x - 0.25)^2]$[4]. In Figure 4, the numerical results are shown.

In the following example, we compare our method by a nonparametric Regression (NR) method. NR methods are very sensitive to parameters such as

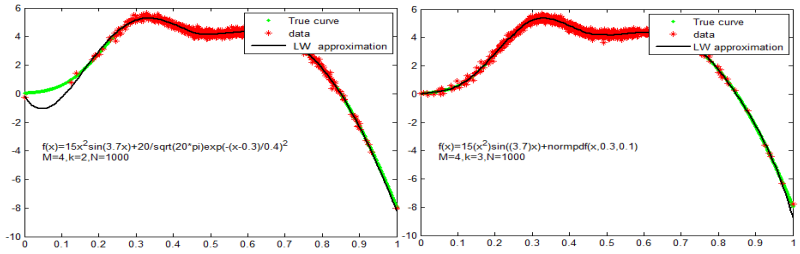


Figure 3: Approximate solution for the regression function $f_3(x)$.

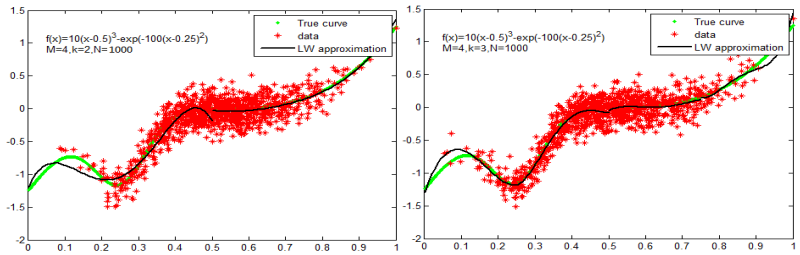


Figure 4: Approximate solution for the regression function $f_4(x)$.

the bandwidth selection, the regression order, and the shape of the smoothing kernel. In these methods, the choice of order and especially the bandwidth parameter can be a hassle [14]. In the previous example, we observed that the Legendre wavelets regression (LWR) method provides a good estimate for N samples (x_i, y_i), which does not depend on any parameter except the choice of k and M, where k specifies the level of resolution, 2^k sub-intervals on $[0, 1]$, and M determines the degree of wavelets. Note that the selection of these two parameters is easy.

Example 4. Consider the functions $f_1(x) = 1 + x - 0.45\exp[-5(x - 0.5)^2]$, $f_2(x) = 15x^2\sin(3.7x) + \frac{2}{\sigma\sqrt{2\pi}}\exp(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2)$ and $f_3(x) = -x^3 - x^2$. In Figure 5, we approximate the minimization problem (2) for $N = 250$ samples of (x_i, y_i) and compare this method by a nonparametric regression method, which are shown in Figure 5.

5 Conclusion

In this paper, Legendre wavelets were used to approximate the regression function. A new operational matrix was introduced to simplify the minimization problem in (2), which is useful for new research in financial mathematics and numerical analysis. Moreover, a new approximation error of

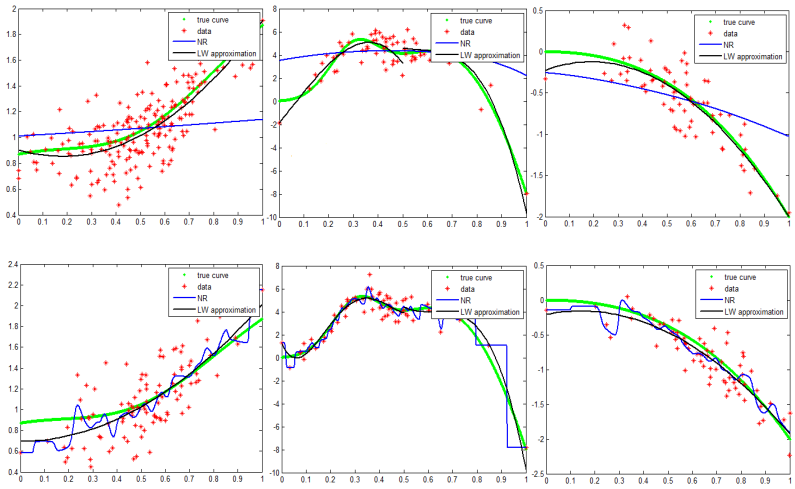


Figure 5: Comparing the Legendre wavelets estimation (black curve) with the nonparametric regression (blue curve). Due to nonoptimal choices of h, under-fitting occurred in the first row and over-fitting occurred in the second row for nonparametric regression for the functions mentioned in Example 3.

a differentiable function f using the partial sums of the Legendre wavelets was provided. Numerical experiments were performed for a variety of real regression functions (see [9, 1, 4]). The proposed method was executed on some popular functions, and the numerical results were compared with the nonparametric regression method. Finally, the capability of the proposed method was illustrated.

References

1. Abraham, C. *Bayesian regression under combinations of constraints,* J. Statist. Plann. Inference, 142 (2012) 2672–2687.

2. Abraham, C. and Khadraoui, K. *Bayesian regression with B-splines under combinations of shape constraints and smoothness properties*,Stat. Neerl. 69 (2015), 150–170.

3. Angelini, C., Canditiis, D.D. and Leblanc, F. *Wavelet regression estimation in nonparametric mixed effect models*, J. Multivariate Anal. 85 (2003) 267–291.

4. Bowman, W., Jones, M.C. and Gijbels, I. *Testing monotonicity of regression*, J. Comput. Graph Stat. 7 (1998) 489–500.

5. Corlay, S. *B-spline techniques for volatility modeling,* J. Comput. Finance, 19 (2016) 97–135.

6. Hamzehnejad, M., Hosseini, M.M. and Salemi, A. *An improved upper bound for ultraspherical coefficients*, Journal of Mathematical Modeling, 10 (2022), 1–11.

7. Khadraoui, K. *A smoothing stochastic simulated annealing method for localized shapes approximation*, JJ. Math. Anal. Appl. 446 (2017), 1018–1029.

8. Mammen, E., Marron, J., Turlach, B. and Wand, M. *A general projection framework for constrained smoothing*, Stat. Sci., 16 (2001) 232–248.

9. Meyer, M.C. *Inference using shape-restricted regression splines,* Ann. Appl. Stat. 2 (2008), 1013–1033.

10. Mohammadi, F. and Hosseini, M.M. *A new Legendre wavelet operational matrix of derivative and its applications in solving the singular ordinary differential equations,* J. Franklin Inst. 348 (2011) 1787–1796.

11. Mohammadi, M. and Bahrkazemi, M. *Bases for polynomial-based spaces*,J. Math. Model. 7 (2019) 21–34.

12. Polpo, A., Louzada, F.,Rifo, L.L.R., Stern, J.M. and Lauretto, M. *Interdisciplinary Bayesian Statistics*, Proceedings of the 12th Brazilian Meeting on Bayesian Statistics (EBEB 2014) held in Atibaia, March 10–14, 2014. Springer Proceedings in Mathematics & Statistics, 118. Springer, Cham, 2015.

13. Rasmussen, C.E. and Williams, C.K.I. *Gaussian processes for machine learning,* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006.

14. Raykar, C. and Duraiswami, R. *Fast optimal bandwidth selection for kernel density estimation,* Proceedings of the Sixth SIAM International Conference on Data Mining, 524–528, SIAM, Philadelphia, PA, 2006.

15. Shen, J., Tang, T. and Wang, L.L. *Spectral methods: Algorithms, analysis and applications*, Vol. 41. Springer Science & Business Media, 2011.

16. Vidakovic, B. *Statistical modeling by wavelets*, Wiley Series in Probability and Statistics: Applied Probability and Statistics. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1999.

17. Wahba, G. *Spline models for observational data,* CBMS-NSF Regional Conference Series in Applied Mathematics, 59. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

18. Wand, M. and Ormerod, J. *On semiparametric regression with O'Sullivan penalized splines,* Aust. N. Z. J. Stat., 50 (2008) 179–198.

19. Wang, H. *A new and sharper bound for Legendre expansion of differentiable functions,* Appl. Math. Lett. 85 (2018) 95–102.

How to cite this article
M. Hamzehnejad, M.M. Hosseini and A. Salemi Estimation of the regression function by Legendre wavelets. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 497-512. doi: 10.22067/ijnao.2022.73876.1079.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 513–532
DOI:10.22067/ijnao.2022.70466.1035
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Using shifted Legendre orthonormal polynomials for solving fractional optimal control problems

R. Naseri* , A. Heydari and A.S. Bagherzadeh

Abstract

Shifted Legendre orthonormal polynomials (SLOPs) are used to approximate the numerical solutions of fractional optimal control problems. To do so, first, the operational matrix of the Caputo fractional derivative, the SLOPs, and Lagrange multipliers are used to convert such problems into algebraic equations. Also, the method is proposed for solving multidimensional problems, and its convergence is proved. This method is tested on some nonlinear examples. The results indicate that the technique can efficiently solve multidimensional problems.

AMS subject classifications (2020): 49M25; 49J30; 34A08.

Keywords: Shifted Legendre orthonormal polynomials (SLOPs); Fractional optimal control problem (FOCP); Caputo fractional derivative

1 Introduction

For the first time, fractional calculus was introduced in the 17th century. Liouville, Grünwald, Letnikov, Riemann, and Caputo substantially contributed

* Corresponding author

Received 18 May 2021; revised 5 March 2022; accepted 9 March 2022

Roghayeh Naseri
Department of Mathematics, Payame Noor University (PNU), P.O.Box 19395-4697, Tehran, Iran. e-mail: phd.naseri.r@pnum.ac.ir

Aghileh Heydari
Department of Mathematics, Payame Noor University (PNU), P.O.Box 19395-4697, Tehran, Iran. e-mail: a_heidari@pnu.ac.ir

Amir Saboor Bagherzadeh
Department of Applied Mathematics, Faculty of Mathematical Science, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: amirsb@um.ac.ir

to the development of its theoretical foundations [6]. They worked on mass and heat transfer problems using the terms semi-derivative and semi-integral. The first book on fractional calculus was written by Oldham and Spanier [27]. Further details on fractional calculus and some of its applications can be found in [11, 12, 21, 22].

In recent years, the applications of fractional calculus in engineering and sciences, including mathematics, fluid dynamics, and physics, have attracted considerable attentions. Fractional calculus is used to extend the usual notions of derivative and integral to ones with real orders and is based on the concepts of fractional derivative in the sense of Caputo and fractional integral in the sense of Riemann–Liouville [22, 27].

When we use a term involving fractional-order derivative(s) in differential equations of optimal control problems, we obtain *fractional optimal control problems* (FOCPs). Many scientific studies confirm the applications of FOCPs in mathematics, mechanics, medicine, and engineering [13, 23]. For example, such problems have been used to obtain numerical solutions of the fractional models of some diseases, such as the fractional-order tumor-immune model, HIV epidemic, and the glucose-insulin system [2, 15, 24].

Orthonormal polynomials have been applied in various linear and nonlinear problems, because they can be used to convert these problems into easy-to-solve algebraic equations. They have many useful properties that facilitate the solution of mathematical problems and provide a way for solving, expanding, and interpreting solutions in some types of differential equations [1, 5, 10, 12].

In this article, we use the SLOPs as the basis functions of the method proposed to solve fractional differential equations. The common approach adopted in the past studies was to solve the one-dimensional problem. Moreover, most of the studies like [5, 4, 10], just obtained the error bound of the operational matrix in fractional derivatives. Hence, none of them proved the convergence of the method under consideration.

Therefore, we aim to develop the method for multidimensional problems in this paper. Moreover, we prove the convergence of the method. The outputs reveal that the method is efficient for multidimensional problems.

We organized the paper as follows. In Section 2, we present the important properties of shifted Legendre polynomials, some preliminary definitions from fractional calculus, and the operational matrix of fractional derivatives. In Section 3, we explain the method and the necessary conditions for the FOCPs. Section 4 discusses the convergence of the proposed technique. In Section 5, we compare our results with those of the previous researches for nonlinear and multidimensional examples. Finally, in Section 6, we present the conclusion.

2 Shifted Legendre orthonormal polynomials

Definition 1. [5] For a function $\xi(t)$, the *Riemann–Liouville fractional integral* of order $\alpha \geq 0$ is defined by

$$I^{\alpha}\xi(t) = \begin{cases} \frac{1}{\Gamma(\alpha)} \int_0^t (t-z)^{\alpha-1}\xi(z)\,dz, & \alpha > 0, \quad t > 0, \\ \xi(t), & \alpha = 0, \end{cases} \tag{1}$$

where

$$\Gamma(\alpha) = \int_0^{\infty} z^{\alpha-1}e^{-z}dz,$$

denotes the gamma function.

Definition 2. [5] For a function $\xi(t)$, the *Caputo fractional derivative* of order α is defined by

$$D^{\alpha}\xi(t) = \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-z)^{n-\alpha-1} \frac{d^n}{dz^n}\,\xi(z)\,dz, \qquad n-1 < \alpha \leqslant n, \quad t > 0, \tag{2}$$

where n is an integer.

Some properties of these operators can be written as

$$D^{\alpha}\,c = 0, \qquad c \text{ is a constant}, \tag{3}$$

$$I^{\alpha}\left(D^{\alpha}\,\xi(t)\right) = \xi(t) - \sum_{k=0}^{n-1} \xi^{(k)}(0)\,\frac{t^k}{k!}, \tag{4}$$

$$D^{\alpha}\,t^{\delta} = \frac{\Gamma(\delta+1)}{\Gamma(\delta+1-\alpha)}\,t^{\delta-\alpha}, \tag{5}$$

and

$$D^{\alpha}\left(\beta\,\xi(t) + \gamma\,\tau(t)\right) = \beta\,D^{\alpha}\,\xi(t) + \gamma\,D^{\alpha}\,\tau(t), \tag{6}$$

where δ, β, and γ are scalar coefficients.

Definition 3. [3] The *Legendre polynomial* of degree i, $p_i(z)$, is defined on the interval $[-1, 1]$ by the recurrence relation

$$p_{i+1}(z) = \frac{2i+1}{i+1}\,z\,p_i(z) - \frac{i}{i+1}\,p_{i-1}(z), \qquad i \geqslant 1, \tag{7}$$

where

$$p_0(z) = 1, \qquad p_1(z) = z. \tag{8}$$

We obtain the *shifted Legendre polynomials* $p_i^*(t)$ on $[0,1]$ if we use the change of variable $z = 2t - 1$:

$$p_{i+1}^*(t) = \frac{2i+1}{i+1}\,(2t-1)\,p_i^*(t) - \frac{i}{i+1}\,p_{i-1}^*(t), \qquad i \geqslant 1, \tag{9}$$

$$p_0^*(t) = 1, \qquad p_1^*(t) = 2t - 1. \tag{10}$$

These polynomials are orthogonal, in the sense that

$$\langle p_j^*(t), p_i^*(t)\rangle = \int_0^1 p_j^*(t)\,p_i^*(t)\,dt = \begin{cases} \frac{1}{2i+1}, & j = i, \\ 0, & j \neq i. \end{cases} \tag{11}$$

As shown in [3], if we introduce the SLOPs $\widehat{p}_i(t) \equiv \sqrt{2i+1}\,p_i^*(t)$, then

$$\int_0^1 \widehat{p}_i(t)\,\widehat{p}_j(t)dt = \begin{cases} 1, & j = i, \\ 0, & j \neq i, \end{cases} \tag{12}$$

and

$$\widehat{p}_i(t) = \sqrt{2i+1}\,\sum_{k=0}^{i} (-1)^{i+k}\,\frac{(i+k)!}{(i-k)!\,(k!)^2}\,t^k. \tag{13}$$

Assume that ζ is any element of $L^2[0,1]$ and

$$\rho_M = \text{span}\{\widehat{p}_0(t), \widehat{p}_1(t), \ldots, \widehat{p}_M(t)\}. \tag{14}$$

Now, for any $h \in \rho_M$, we can write $h \simeq \sum_{i=0}^{M} d_i\,\widehat{p}_i(t)$, where the coefficients d_i are determined as follows:

$$d_i = \int_0^1 h(t)\,\widehat{p}_i(t)\,dt, \quad i = 0, 1, \ldots, M. \tag{15}$$

We call $\zeta_\rho \in \rho_M$ *the best approximation* of ζ out of ρ_M whenever

$$\text{for all } h \in \rho_M : \ \|\zeta - \zeta_\rho\|_2 \leqslant \|\zeta - h\|_2. \tag{16}$$

Since $\zeta_\rho \in \rho_M$, there exist coefficients $c_i, i = 0, 1, \ldots, M$, such that

$$\zeta_\rho(t) \simeq \sum_{i=0}^{M} c_i\,\widehat{p}_i(t). \tag{17}$$

So, the matrix form of $\zeta_\rho(t)$ is

$$\zeta_\rho(t) \simeq F^T \triangle_M(t), \tag{18}$$

where

$$F = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_M \end{pmatrix}, \qquad \triangle_M(t) = \begin{pmatrix} \widehat{p}_0(t) \\ \widehat{p}_1(t) \\ \vdots \\ \widehat{p}_M(t) \end{pmatrix}. \tag{19}$$

Theorem 1. For the SLOPs vector $\triangle_M(t)$, the fractional derivative of order α, in the sense of Caputo, is defined as follows:

$$D^\alpha \triangle_M(t) = D_{(\alpha)} \triangle_M(t). \tag{20}$$

Herein, $D_{(\alpha)}$ denotes the $(M+1)\times(M+1)$ operational matrix of the fractional derivative, given by

$$D_{(\alpha)} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ W_\alpha(n,0) & W_\alpha(n,1) & W_\alpha(n,2) & \cdots & W_\alpha(n,M) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_\alpha(M,0) & W_\alpha(M,1) & W_\alpha(M,2) & \cdots & W_\alpha(M,M) \end{bmatrix},$$

where

$$W_\alpha(k,j) \tag{21}$$
$$= \sqrt{(2j+1)\,(2k+1)} \sum_{i=n}^{k} \sum_{l=0}^{j} \frac{(-1)^{k+j+i+l}\,(k+i)!\,(l+j)!}{(k-i)!\,i!\,\Gamma(i-\alpha+1)\,(j-l)!\,(l!)^2\,(i+l-\alpha+1)},$$

and rows 0 to n-1 are zero.

Proof. See [3]. □

3 The numerical method

To solve the following problem, we use the operational matrix of fractional derivatives, the SLOPs and Lagrange multipliers.

$$\min J = \int_{t_0}^{t_1} f\big(t, x(t), u(t)\big)\, dt, \tag{22}$$

$$D^\alpha x(t) = \phi\big(t, x(t), u(t)\big), \qquad n-1 < \alpha \leqslant n,\ t \in [t_0, t_1], \tag{23}$$

$$D^{(k)}\, x(t_0) = x_k, \qquad k = 0, 1, \ldots, n-1. \tag{24}$$

Here, $\phi\big(t, x(t), u(t)\big) = g\big(t, x(t)\big) + b(t)\, u(t)$, and S is the feasible solution set. Also, $u(t)$ and $x(t)$ denote the control and state variables, respectively, $u(t)$ is continuous, $x(t)$ is continuously differentiable, $g\big(t, x(t)\big)$, $f\big(t, x(t), u(t)\big)$, and $b(t)$ are smooth functions, $b(t)$ is invertible, $f\big(t, x(t), u(t)\big)$ and $\phi\big(t, x(t), u(t)\big)$ are convex functions, S is a convex set, and $f\big(t, x(t), u(t)\big)$ is integrable on $I = [t_0, t_1]$. Moreover, $f\big(t, x(t), u(t)\big)$ and $g\big(t, x(t)\big)$ satisfy the Lipschitz property. In fact,

$$\|f\big(t, x_1(t), u_1(t)\big) - f\big(t, x_2(t), u_2(t)\big)\| \leq L(\|x_1(t) - x_2(t)\| + \|u_1(t) - u_2(t)\|), \tag{25}$$

and

$$\|g\big(t, x_1(t)\big) - g\big(t, x_2(t)\big))\| \leq K(\|x_1(t) - x_2(t)\|), \tag{26}$$

where L and K are positive constants. Approximate $x(t)$ by the SLOPs $\widehat{p}_i(t)$ as

$$\overline{x}_M(t) = C^T \triangle_M(t), \tag{27}$$

where C^T is an unknown scalar coefficient vector given by

$$C^T = \begin{pmatrix} c_0 & c_1 & \cdots & c_M \end{pmatrix}. \tag{28}$$

We defined $\widehat{p}_i(t)$ and $\triangle_M(t)$ in (10) and (19), respectively. By (27), we can rewrite the dynamic constraint (23) as

$$C^T\, D_{(\alpha)}\, \triangle_M(t) = g\big(t, C^T\, \triangle_M(t)\big) + b(t)\, u(t). \tag{29}$$

So, we obtain

$$u(t) = \frac{1}{b(t)} \big(C^T\, D_{(\alpha)}\, \triangle_M(t) - g\big(t, C^T\, \triangle_M(t)\big)\big). \tag{30}$$

Then, we can rewrite the initial conditions (24) in the form

$$C^T\, D_{(k)}\, \triangle_M(t_0) - x_k = 0, \qquad k = 0, 1, \ldots, n-1. \tag{31}$$

Due to (27), (30) and (31), the performance index J can be approximated by

$$J_M\,[C^T] = \int_{t_0}^{t_1} \widehat{f}\big(t, \overline{x}_M(t), D^{\alpha}\overline{x}_M(t)\big)dt + \sum_{k=0}^{n-1} \big(C^T\, D_{(k)}\, \triangle_M(t_0) - x_k\big)\lambda_k, \tag{32}$$

where

$$\widehat{f}\big(t,\overline{x}_M(t),D^{\alpha}\overline{x}_M(t)\big)=f\big(t,C^T\triangle_M(t),\frac{1}{b(t)}\left(C^T D_{(\alpha)}\triangle_M(t)-g(t,C^T\triangle_M(t))\right)\big), \quad (33)$$

and λ_k denotes the Lagrange multiplier, which should be determined [11].

The necessary conditions for the optimality of (22) are subject to the dynamic constraints (23) and (24) in the form

$$\frac{\partial J_M}{\partial c_i}=0,\ i=0,1,\ldots,M, \qquad \frac{\partial J_M}{\partial \lambda_k}=0, \quad k=0,1,\ldots,n-1. \quad (34)$$

We can use any standard iterative method to solve the aforementioned system for c_i, $i=0,1,\ldots,M$, and λ_k, $k=0,1,\ldots,n-1$. As a result, we obtain $x(t)$ and $u(t)$ as given in (27) and (30), respectively [3].

4 Convergence analysis

The use of SLOPs operates as a proof of convergence in three steps. In the first step, we show that the usage is indeed justifiable. In the second step, we show that the functional derivative of a shifted Legendre polynomial is a proper approximation for the same derivative. In the third step, we indicate the difference between the target function for any optimized solution and the value of the target function of the shifted Legendre approximation, tends to zero as the number of the shifted Legendre orthonormal basis increases. We complete these steps by the hypotheses, Lemmas 1 and 2. To find an upper bound for the operational matrix errors in fractional derivatives and to prove the convergence, we use the following theorems.

Theorem 2. Let $\mathcal{H}$ be a Hilbert space, and let Y be a finite-dimensional subspace of $\mathcal{H}$. Also, assume that $\{y_1,y_2,\ldots,y_M\}$ is any basis for Y. Given any x in $\mathcal{H}$, let y_0 denotes the unique best approximation of x out of Y. Then,

$$\|x-y_0\|_2^2=\frac{G(x,y_1,y_2,\ldots,y_M)}{G(y_1,y_2,\ldots,y_M)}, \quad (35)$$

where

$$G(x,y_1,y_2,\cdots,y_M)=\begin{vmatrix}\langle x,x\rangle & \langle x,y_1\rangle & \cdots & \langle x,y_M\rangle\\ \langle y_1,x\rangle & \langle y_1,y_1\rangle & \cdots & \langle y_1,y_M\rangle\\ \vdots & \vdots & \vdots & \vdots\\ \langle y_M,x\rangle & \langle y_M,y_1\rangle & \cdots & \langle y_M,y_M\rangle\end{vmatrix}, \quad (36)$$

and

$$G(y_1, y_2, \cdots, y_M) = \begin{vmatrix} \langle y_1, y_1 \rangle & \cdots & \langle y_1, y_M \rangle \\ \vdots & \vdots & \vdots \\ \langle y_M, y_1 \rangle & \cdots & \langle y_M, y_M \rangle \end{vmatrix}. \tag{37}$$

Proof. See [5]. □

We show that the upper bound of operational matrix errors in fractional derivatives $D^{(\alpha)}$ can be obtained as

$$\varepsilon_D^{\alpha} := D^{(\alpha)} \triangle_M(t) - \widehat{D}^{\alpha} \triangle_M(t), \tag{38}$$

where $\widehat{D}^{\alpha}$ is an approximation of the operator $D^{(\alpha)}$ and

$$\varepsilon_D^{\alpha} = \begin{pmatrix} \varepsilon_{D,0}^{\alpha} \\ \varepsilon_{D,1}^{\alpha} \\ \vdots \\ \varepsilon_{D,M}^{\alpha} \end{pmatrix}. \tag{39}$$

As mentioned in [18], for each element of ε_D^{α}, an upper bound for the error related to $D^{(\alpha)}$ can be written as follows:

$$\|\varepsilon_{D,k}^{\alpha}\|_2 \leqslant \sqrt{2k+1} \sum_{i=1}^{k} \left| \frac{(k+i)!}{(k-i)!\, i!\, \Gamma(i-\alpha+1)} \right| \times \left(\frac{G(t^{i-1}, \widehat{p}_0(t), \ldots, \widehat{p}_M(t))}{G(\widehat{p}_0(t), \ldots, \widehat{p}_M(t))} \right)^{\frac{1}{2}}, \quad 0 \leqslant k \leqslant M. \tag{40}$$

By Theorem 2 and (40), we conclude that ε_D^{α} tends to zero as the number of the shifted Legendre orthonormal basis increases [5].

Lemma 1. Let $x(t)$ be a continuously differentiable function, and let $\overline{x}_M(t)$ denote the approximation of $x(t)$ by the *SLOPs*. Then,

$$\|x(t) - \overline{x}_M(t)\| \to 0 \quad as \quad M \to \infty. \tag{41}$$

Proof. See [15]. □

Lemma 2. For $x(t)$ and $\overline{x}_M(t)$ as in Lemma 1, when $M \to \infty$,

$$\|D^{\alpha}\, x(t) - D^{\alpha}\, \overline{x}_M(t)\| \to 0, \tag{42}$$

$$|D^k\, \overline{x}_M(t_0) - x_k| = 0, \quad k = 0, 1, \ldots, n-1, \tag{43}$$

$$\|\dot{x}(t) - \dot{x}_m(t)\| \to 0. \tag{44}$$

Proof. See [5]. □

We define $J1\,[C^T]$ as follows:

$$J1\,[C^T] = \int_{t_0}^{t_1} f\big(t, x(t), \frac{1}{b(t)}\left(D_{(\alpha)}\,x(t) - g\big(t, x(t)\big)\right) dt + \sum_{k=0}^{n-1} \left(D_{(k)}\,x(t_0) - x_k\right)\lambda_k. \tag{45}$$

Theorem 3. Consider problems (22)–(24), and let $x^*(t)$ be an optimal solution of $\min J1\,[C^T]$. Then,

$$\left|J_M\,[C^T] - J1\,[C^T]\right| \to 0 \quad as \quad M \to \infty. \tag{46}$$

Proof. Using (27) and (30) we obtain

$$\begin{aligned}\left|J_M\,[C^T] - J1\,[C^T]\right| = \Big| & \int_{t_0}^{t_1} f\big(t, C^T\,\triangle_M(t), \frac{1}{b(t)}\left(C^T\,D_{(\alpha)}\,\triangle_M(t) - g(t, C^T\,\triangle_M(t))\right)\big)\,dt \\ & + \sum_{k=0}^{n-1}\left(C^T\,D_{(k)}\,\triangle_M(t_0) - x_k\right)\lambda_k \\ & - \int_{t_0}^{t_1} f(t, x^*(t), \frac{1}{b(t)}\left(D_{(\alpha)}\,x^*(t) - g(t, x^*(t))\right)\big)\,dt \\ & - \sum_{k=0}^{n-1}\left(D_{(k)}\,x^*(t_0) - x_k\right)\lambda_k\,\Big|.\end{aligned}$$

According to (24), (31), and Lemmas 1 and 2, we know that

$$\sum_{k=0}^{n-1}\left(C^T\,D_{(k)}\,\triangle_M(t_0) - x_k\right)\lambda_k = 0$$

and that $\sum_{k=0}^{n-1}\left(D_{(k)}\,x^*(t_0) - x_k\right)\lambda_k = 0$. So,

$$\begin{aligned}&\left|J_M\,[C^T] - J1\,[C^T]\right| \\ &\quad = \Big|\int_{t_0}^{t_1}\Big(f\big(t, C^T\,\triangle_M(t), \frac{1}{b(t)}\left(C^T\,D_{(\alpha)}\,\triangle_M(t) - g(t, C^T\,\triangle_M(t))\right)\big) \\ &\qquad - f\big(t, x(t), \frac{1}{b(t)}\left(D_{(\alpha)}\,x(t) - g(t, x(t))\right)\big)\Big)dt\Big|\end{aligned}$$

We know that f satisfies the Lipschitz condition. Therefore,

$$\begin{aligned}&\left|J_M\,[C^T] - J1\,[C^T]\right| \\ &\quad \le \int_{t_0}^{t_1}\Big(L\left(\|C^T\,\triangle_M(t) - x(t)\|\right) \\ &\qquad + \|\frac{1}{b(t)}\left(C^T\,D_{(\alpha)}\,\triangle_M(t) - g\big(t, C^T\,\triangle_M(t)\big) - D_{(\alpha)}\,x(t) + g(t, x(t))\right)\|\Big)dt.\end{aligned}$$

By the Schwartz inequality and separating integrals, we obtain

$$\begin{aligned}
&\left|J_M\left[C^T\right]-J1\left[C^T\right]\right| \\
&\quad\leq L\int_{t_0}^{t_1}\left(\left\|C^T\triangle_M(t)-x(t))\right\|\right)dt \\
&\qquad+\frac{1}{|b(t)|}\int_{t_0}^{t_1}\left(\left(\left\|C^T D_{(\alpha)}\triangle_M(t)-D_{(\alpha)}x(t)\right\|\right)dt\right. \\
&\qquad+\frac{1}{|b(t)|}\int_{t_0}^{t_1}\left(\left\|g(t,x(t))-g(t,C^T\triangle_M(t))\right\|\right)dt.
\end{aligned}$$

We write the upper bounds of integrals and note that g satisfies the Lipschitz condition. Then,

$$\begin{aligned}
\left|J_M\left[C^T\right]-J1\left[C^T\right]\right|\leq L(t_1-t_0)\big(&\left\|C^T\triangle_M(t)-x(t)\right\| \\
&+\frac{(t_1-t_0)}{|b(t)|}\big(\left\|C^T D_{(\alpha)}\triangle_M(t)-D_{(\alpha)}x(t)\right)\| \\
&+\frac{K\,(t_1-t_0)}{|b(t)|}\left\|x(t)-C^T\triangle_M(t)\right\|.
\end{aligned}$$

If $M\to\infty$, then Lemma 1 shows that the first and third terms tend to zero. Also, the second term tends to zero by Lemma 2. Consequently, $J_M\left[C^T\right]\to J1[C^T]$. □

Through Theorem 3, we observed that the difference between the value of the target function for any optimized solution of $\min J1\left[C^T\right]$ and that of the target function for the approximate value of Legendre tends to zero as $M\to\infty$. Having (27)–(32) in mind, $\min J1\left[C^T\right]$ is equivalent to (22). Hence, the difference between the value of target function (22) and that of the Legendre approximate target function tends to zero.

5 Numerical experiments

In this section, we prove the accuracy of the proposed technique by providing some examples and then comparing our achievements with the numerical results obtained in other papers by the computer with Intel Core i7 CPU up to 3.5 GHz, RAM 12GB, and the codes written with Wolfram Mathematica 11.

Example 1. Consider the problem

$$\min\ J=\int_0^1\left((x(t)-t^2)^2+(u(t)+t^4-\frac{20\,t^{\frac{9}{10}}}{9\,\Gamma(\frac{9}{10})})^2\right)dt, \tag{47}$$

subject to dynamic constraints

$$D^{1.1}\,x(t) = t^2\,x(t) + u(t), \tag{48}$$

$$x(0) = \dot{x}(0) = 0. \tag{49}$$

Due to (48), we obtain $u(t)$ and rewrite (47) as

$$u(t) = D^{1.1}\,x(t) - t^2 x(t),$$

$$\begin{aligned}\min\ J = \int_0^1 & \big((C^T \triangle_M(t) - t^2)^2 \\ & + (D^{1.1}\,C^T \triangle_M(t) - t^2\,C^T \triangle_M(t) + t^4 - \frac{20\,t^{\frac{9}{10}}}{9\,\Gamma(\frac{9}{10})})^2\big)\,dt \\ & + \big(C^T\,D_{(0)} \triangle_M(t_0) - x(0)\big)\lambda_0 + \big(C^T\,D_{(1)} \triangle_M(t_0) - \dot{x}(0)\big)\lambda_1.\end{aligned}$$

The functional J is minimized by $x^*(t) = t^2$ and $u^*(t) = \frac{20\,t^{\frac{9}{10}}}{9\,\Gamma(\frac{9}{10})} - t^4$, with minimum equal to zero. Table 2 presents the approximate values of J, which are obtained by the proposed method and the methods utilized in [21, 3], with different values of M. As the results indicate, our approach is better than the ones used in [21, 3].

Table 1: Approximations of J with different values of M

M	The method	The method used in [21]	The method used in [3]
4	1.66202×10^{-6}	6.07530×10^{-6}	4.76932×10^{-6}
6	2.44576×10^{-7}	5.91532×10^{-7}	5.37825×10^{-7}
8	5.90947×10^{-8}	1.21966×10^{-7}	1.06099×10^{-7}
9	3.26447×10^{-8}	7.03371×10^{-8}	5.44304×10^{-8}

Table 3 presents the absolute values of errors for the control and state variables for various values of t. Also, in Figure 6, the approximate and exact values of the control and state variables are plotted for $M = 6$.

Table 2: Absolute errors of $x(t)$ and $u(t)$ at $M = 6$

t	$\lvert x^*(t) - x(t)\rvert$	$\lvert u^*(t) - u(t)\rvert$
0.1	1.60241×10^{-7}	1.72334×10^{-5}
0.2	2.35607×10^{-7}	4.57424×10^{-4}
0.3	9.96796×10^{-8}	2.85637×10^{-4}
0.4	6.68032×10^{-8}	2.89849×10^{-4}
0.5	7.86075×10^{-8}	1.79588×10^{-4}
0.6	9.06389×10^{-8}	2.80773×10^{-4}
0.7	2.84397×10^{-7}	1.15197×10^{-4}
0.8	2.78471×10^{-7}	2.69036×10^{-4}
0.9	3.55721×10^{-8}	2.73064×10^{-4}

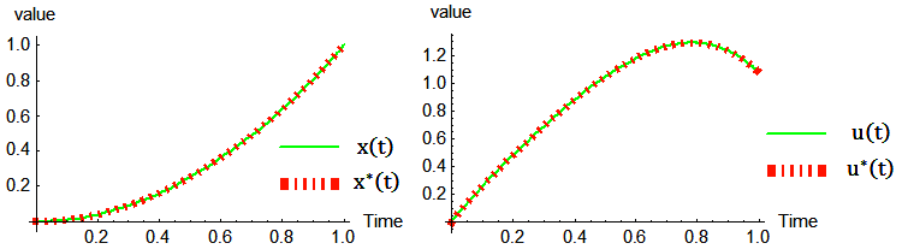


Figure 1: Approximate and exact values of the control and state variables for $M = 6$

Example 2. Consider the two-dimensional problem

$$\begin{aligned}\min J = \int_0^1 \big((x_1(t) - t^2)^2 + (x_2(t) - t^3)^2 \\ + (u_1(t) - t^4 + \frac{\Gamma(4)}{6\,\Gamma(2.9)}\, t^{1.9} - \frac{\Gamma(3)}{3\,\Gamma(1.9)}\, t^{0.9})^2 \\ + (u_2(t) - t^5 + \frac{\Gamma(4)}{2\,\Gamma(2.9)}\, t^{1.9})^2\big)\, dt, \end{aligned} \tag{50}$$

subject to dynamic constraints

$$D^{1.1}\, x_1(t) = 3\, u_1(t) - 3\, t^2 x_1(t) + t^2\, x_2(t) - u_2(t), \tag{51}$$

$$D^{1.1}\, x_2(t) = -2\, u_2(t) + (2\, t^2 - 1)\, x_2(t) + t\, x_1(t), \tag{52}$$

$$x_1(0) = \dot{x_1}(0) = 0, \tag{53}$$

and

$$x_2(0) = \dot{x_2}(0) = 0. \tag{54}$$

By (51) and (52), we obtain $u_1(t)$ and $u_2(t)$ as follows:

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & -\frac{1}{6} \\ 0 & -\frac{1}{2} \end{bmatrix} \Big(\begin{bmatrix} D^{1.1}\, x_1(t) \\ D^{1.1}\, x_2(t) \end{bmatrix} - \begin{bmatrix} -3\, t^2 x_1(t) + t^2\, x_2(t) \\ (2\, t^2 - 1)\, x_2(t) + t\, x_1(t) \end{bmatrix} \Big).$$

We define

$$x_1(t) = C_1^T \triangle_M(t), \quad C_1^T = \big(c_{10}\ c_{11}\ \cdots\ c_{1M}\big),$$
$$x_2(t) = C_2^T \triangle_M(t), \quad C_2^T = \big(c_{20}\ c_{21}\ \cdots\ c_{2M}\big),$$

and rewrite (50) as

$$\begin{aligned} \min J = \int_0^1 & \big((C_1^T \triangle_M(t) - t^2)^2 + (C_2^T \triangle_M(t) - t^3)^2 \\ & + \big(\frac{1}{3}(D^{1.1}\, C_1^T \triangle_M(t) + 3\, t^2\, (C_1^T \triangle_M(t)) - t^2\, (C_2^T \triangle_M(t))) \\ & - \frac{1}{6}(D^{1.1}\, C_2^T \triangle_M(t) - (2\, t^2 - 1)\, (C_2^T \triangle_M(t)) - t\, (C_1^T \triangle_M(t))) - t^4 \\ & + \frac{\Gamma(4)}{6\,\Gamma(2.9)}\, t^{1.9} - \frac{\Gamma(3)}{3\,\Gamma(1.9)}\, t^{0.9}\big)^2 + \big(-\frac{1}{2}(D^{1.1}\, C_2^T \triangle_M(t) \\ & - (2\, t^2 - 1)\, (C_2^T \triangle_M(t)) - t\, (C_1^T \triangle_M(t))) - t^5 + \frac{\Gamma(4)}{6\,\Gamma(2.9)}\, t^{1.9}\big)^2\big) dt \\ & + \big(C_1^T\, D_{(0)} \triangle_M(t_0) - x_1(0)\big)\lambda_0 + \big(C_1^T\, D_{(1)} \triangle_M(t_0) - \dot{x_1}(0)\big)\lambda_1 \\ & + \big(C_2^T\, D_{(0)} \triangle_M(t_0) - x_2(0)\big)\lambda_0 + \big(C_2^T\, D_{(1)} \triangle_M(t_0) - \dot{x_2}(0)\big)\lambda_1. \end{aligned}$$

The functions $x_1^*(t) = t^2, x_2^*(t) = t^3$ and $u_1^*(t) = t^4 - \frac{\Gamma(4)}{6\,\Gamma(2.9)}\, t^{1.9} + \frac{\Gamma(3)}{3\,\Gamma(1.9)}\, t^{0.9}, u_2^*(t) = t^5 - \frac{\Gamma(4)}{6\,\Gamma(2.9)}\, t^{1.9}$ minimize the functional J, and the minimum value is zero. In Table 4, we present the approximate values of J with different values of M.

Table 3: Approximate values of J with different values of M

M	J
4	2.39801×10^{-7}
6	3.03043×10^{-8}
8	6.97336×10^{-9}
9	6.97321×10^{-9}

Table 4 presents the absolute values of errors for the state and control variables for various values of t.

Also, in Figures 2 and 3, the approximate and exact values of the state and

Table 4: Absolute errors of $x_1(t)$, $x_2(t)$, $u_1(t)$, and $u_2(t)$ at $M = 6$

t	$\lvert x_1^*(t) - x_1(t)\rvert$	$\lvert x_2^*(t) - x_2(t)\rvert$	$\lvert u_1^*(t) - u_1(t)\rvert$	$\lvert u_2^*(t) - u_2(t)\rvert$
0.1	7.19262×10^{-7}	1.74666×10^{-7}	6.4603×10^{-6}	9.51622×10^{-6}
0.2	1.0357×10^{-6}	2.48769×10^{-7}	1.60228×10^{-4}	2.89678×10^{-5}
0.3	3.70976×10^{-7}	7.82014×10^{-8}	1.03983×10^{-4}	1.19302×10^{-5}
0.4	4.54804×10^{-7}	1.37132×10^{-7}	1.05124×10^{-4}	1.82481×10^{-5}
0.5	5.92208×10^{-7}	1.84041×10^{-7}	6.48507×10^{-5}	8.03613×10^{-6}
0.6	9.51419×10^{-8}	1.81842×10^{-8}	1.04023×10^{-4}	1.65065×10^{-5}
0.7	9.14941×10^{-7}	2.02377×10^{-7}	3.7991×10^{-5}	6.07151×10^{-6}
0.8	8.57316×10^{-7}	2.32216×10^{-7}	9.89654×10^{-5}	1.59221×10^{-5}
0.9	2.67307×10^{-7}	2.16467×10^{-9}	9.10531×10^{-5}	1.77034×10^{-5}

control variables are plotted at $M = 6$.

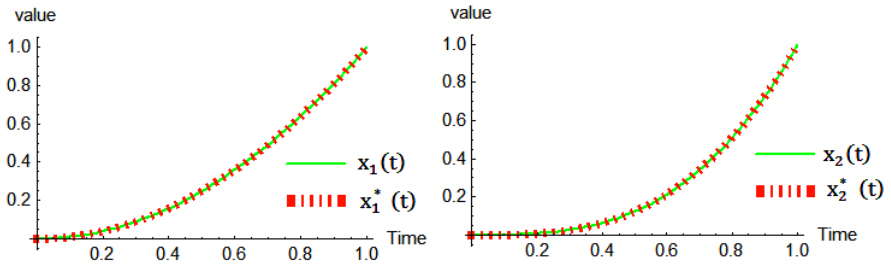


Figure 2: Approximate and exact values of the state variable at $M = 6$

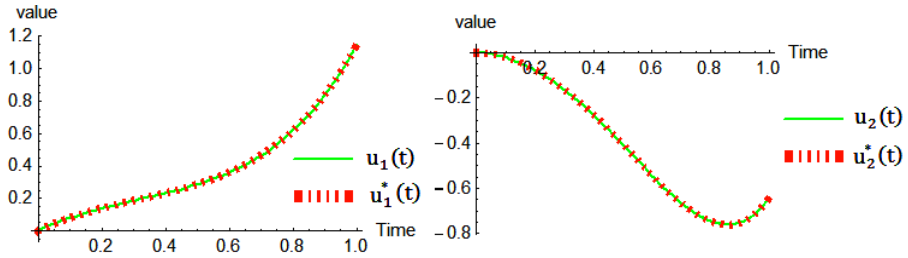


Figure 3: Approximate and exact values of the control variable at $M = 6$

We can apply this method to another category of problems. In fact, if in problems (22)–(24), we replace (23) by

$$\varphi\, D^{\alpha}\, x(t) + \psi\, \dot{x}(t) = g\big(t, x(t)\big) + b(t)\, u(t), \qquad (55)$$
$$n - 1 < \alpha \leqslant n,\ b(t) \neq 0,\ t \in [t_0, t_1],$$

then the method still converges according to (44), where φ and ψ are scalar coefficients. Let us present one example of this form.

Example 3. Recall from [28] the problem

$$\min J = \int_0^1 \big(u(t) - x(t)\big)^2\, dt, \tag{56}$$

subject to dynamic constraints

$$\dot{x}(t) + D^{\alpha}x(t) = u(t) - x(t) + \frac{6\, t^{\alpha+2}}{\Gamma(\alpha+3)} + t^3, \tag{57}$$

and

$$x(0) = 0. \tag{58}$$

By (57), we can find $u(t)$:

$$u(t) = \dot{x}(t) + D^{\alpha}x(t) + x(t) - \frac{6\, t^{\alpha+2}}{\Gamma(\alpha+3)} - t^3,$$

$$\min J = \int_0^1 \Big(C^T \dot{\triangle}_M(t) + D^{\alpha}(C^T \triangle_M(t)) - \frac{6\, t^{\alpha+2}}{\Gamma(\alpha+3)} - t^3\Big)^2 dt$$
$$+ \big(C^T D_{(0)} \triangle_M(t_0) - x(0)\big)\lambda_0.$$

The functions $x^*(t) = \frac{6\, t^{\alpha+3}}{\Gamma(\alpha+4)}$ and $u^*(t) = \frac{6\, t^{\alpha+3}}{\Gamma(\alpha+4)}$ minimize the functional J, and the minimum value is zero. In Table 5, we present the approximate values of J with different values of M.

Table 5: Approximate values of J at $\alpha = 0.9$ with different values of M

M	J
4	2.32302×10^{-7}
6	2.32786×10^{-10}
8	2.98816×10^{-12}

Table 6 presents the absolute values of errors for the control and state variables for various values of t.
Also, in Figure 3, the approximate and exact values of the control and state variables are plotted for $M = 6$. Tables 3 and 8 present the maximum errors of $u(t)$ and $x(t)$ with different values of M.
Also, in Figure 5, the control and state variables are plotted for $M = 5$ and different values of α.

Table 6: Absolute errors of $x(t)$ and $u(t)$ at $M = 6$

t	$\lvert x^*(t) - x(t)\rvert$	$\lvert u^*(t) - u(t)\rvert$
0.1	3.22688×10^{-7}	2.3951×10^{-5}
0.2	4.89573×10^{-7}	1.18457×10^{-5}
0.3	5.31838×10^{-7}	1.52362×10^{-5}
0.4	6.51328×10^{-7}	5.73914×10^{-6}
0.5	1.48297×10^{-7}	1.58438×10^{-5}
0.6	6.3336×10^{-7}	2.83551×10^{-5}
0.7	1.34478×10^{-7}	1.45402×10^{-5}
0.8	5.49314×10^{-7}	7.44278×10^{-6}
0.9	1.0371×10^{-7}	1.81787×10^{-5}

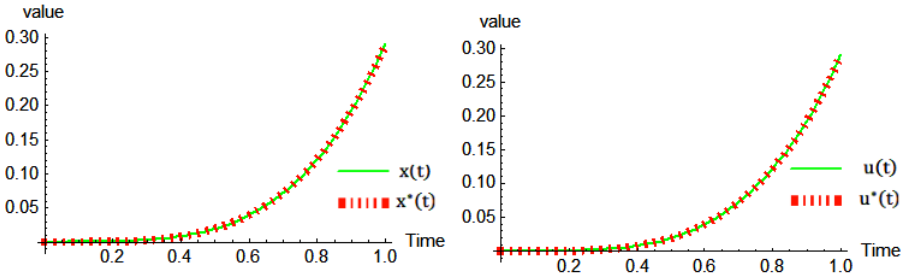


Figure 4: Approximate and exact values of the state and control variables at $M = 6$

Table 7: Maximum errors of $x(t)$ and $u(t)$ at $M = 3$.

$M = 3$	Maximum errors of $x(t)$	Maximum errors of $u(t)$
The method	2.36519×10^{-3}	2.30757×10^{-2}
Algorithm 1 in [28]	8.8025×10^{-3}	8.8025×10^{-3}
Algorithm 2 in [28]	5.1966×10^{-3}	4.3260×10^{-2}

Table 8: Maximum errors of $x(t)$ and $u(t)$ at $M = 5$.

$M = 5$	Maximum errors of $x(t)$	Maximum errors of $u(t)$
Our method	2.21121×10^{-5}	4.7773×10^{-4}
Algorithm 1 in [28]	1.0903×10^{-4}	1.0903×10^{-4}
Algorithm 2 in [28]	4.5321×10^{-5}	6.3134×10^{-4}

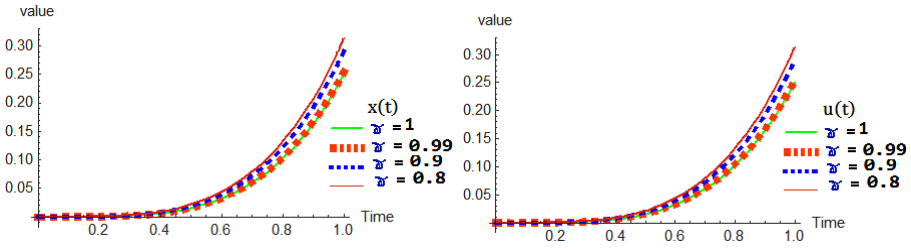


Figure 5: Control and state variables for $M = 5$ and different values of α

6 Conclusion

In this paper, we applied a numerical method to solve a class of fractional optimal control problems. We used the SLOPs and the operational matrix of fractional derivatives. Then, we used the Newton iterative technique to solve these problems. We obtained the error bound of the operational matrix in fractional derivatives and proved the convergence of the method. We focused on multidimensional problems, which have never been solved by this technique. To show the efficiency of the method for multidimensional problems, we provided some nonlinear examples. Comparison of our results with those obtained by other techniques in previous studies revealed the accuracy of the proposed technique for nonlinear and multidimensional problems.

References

1. Abdelhakem, M., Moussa, H., Baleanu, D.,and El-Kady, M. *Shifted Chebyshev schemes for solving fractional optimal control problems.* J. Vib. Control, 25 (2019) 1–8.

2. Arshad, S., Yıldız, T. A., Baleanu, D., and Tang, Y. *The role of obesity in fractional order tumor-immune model.* Politehn. Univ. Bucharest Sci. Bull. Ser. A Appl. Math. Phys. 82(2) (2020) 181–196.

3. Bhrawy, A. H., Doha, E. H., Baleanu, D., Ezz-Eldien, S. S., and Abdelkawy, M. A. *An accurate numerical technique for solving fractional optimal control problems.* Proc. Rom. Acad. Ser. A Math. Phys. Tech. Sci. Inf. Sci. 16(1) (2015) 47–54.

4. Bhrawy, A. H., Doha, E. H., Tenreiro Machado, J. A., and Ezz-Eldien, S. S. *An efficient numerical scheme for solving multi-dimensional fractional optimal control problems with a quadratic performance index.* Asian J. Control 17(6) (2015) 2389–2402.

5. Bhrawy, A. H., Ezz-Eldien, S. S., Doha, E. H., Abdelkawy, M. A., and Baleanu, D. *Solving fractional optimal control problems within a Chebyshev–Legendre operational technique.* Internat. J. Control 90(6) (2017) 1230–1244.

6. Caputo, M. *Mean fractional-order-derivatives differential equations and filters.* Ann. Univ. Ferrara Sez. VII (N.S.) 41 (1995), 73–84 (1997).

7. Daftardar-Gejji, V. (Ed.) *Fractional Calculus and Fractional Differential Equations.* Springer Singapore, 2019.

8. Ding, X., Cao, J., Zhao, X., and Alsaadi, F. E. *Mittag-Leffler synchronization of delayed fractional-order bidirectional associative memory neural networks with discontinuous activations: state feedback control and impulsive control schemes.* Proc. A. 473 (2017), no. 2204, 20170322, 21 pp.

9. El-Sayed, A. A., and Agaewal, P. *Numerical solution of multiterm variable-order fractional differential equations via shifted Legendre polynomials.* Math. Methods Appl. Sci. 42(11) (2019) 3978–3991.

10. Ezz-Eldien, S. S., Doha, E. H., Baleanu, D., and Bhrawy, A. H. *A numerical approach based on Legendre orthonormal polynomials for numerical solutions of fractional optimal control problems.* J. Vib. Control, 23(1) (2017) 16–30.

11. Hassani, H., Avazzadeh, Z., and Machado, J. A. T. *Solving two-dimensional variable-order fractional optimal control problems with transcendental Bernstein series.* Journal of Computational and Nonlinear Dynamics 14(6) (2019).

12. Hassani, H., Machado, J. T., and Naraghirad, E. *Generalized shifted Chebyshev polynomials for fractional optimal control problems.* Commun. Nonlinear Sci. Numer. Simul. 75 (2019), 50–61.

13. Heydari, M. H., and Avazzadeh, Z. *A computational method for solving two-dimensional nonlinear variable-order fractional optimal control problems.* Asian J. Control 22 (2020), no. 3, 1112–1126.

14. Kashkari, B. S., and Syam, M. I. *Fractional-order Legendre operational matrix of fractional integration for solving the Riccati equation with fractional order.* Appl. Math. Comput. 290 (2016), 281–291.

15. Khan, M. W., Abid, M., Khan, A. Q., and Mustafa, G. (2020). *Controller design for a fractional-order nonlinear glucose-insulin system using feedback linearization.* Transactions of the Institute of Measurement and Control. 42(13) (2020) 2372–2381.

16. Khan, R. A., and Khalil, H. *A new method based on legendre polynomials for solution of system of fractional order partial differential equations.* Int. J. Comput. Math. 91 (2014), no. 12, 2554–2567.

17. Kreyszing, E. *Introductory functional analysis with applications.* John Wiley & Sons, New York-London-Sydney, 1978.

18. Li, R., Cao, J., Alsaedi, A., and Alsaadi, F. *Stability analysis of fractional-order delayed neural networks.* Nonlinear Anal. Model. Control 22(4) (2017 505–520.

19. Lotfi, A., Dehghan, M., and Yousefi, S. A. *A numerical technique for solving fractional optimal control problems.* Comput. Math. Appl. 62(3) (2011) 1055–1067.

20. Lotfi, A., Yousefi, S. A., and Dehghan, M. *Numerical solution of a class of fractional optimal control problems via the Legendre orthonormal basis combined with the operational matrix and the Gauss quadrature rule.* JJ. Comput. Appl. Math. 250 (2013), 143–160.

21. Machado, J. T., Kiryakova, V., and Mainardi, F. Recent history of fractional calculus. Communications in nonlinear science and numerical simulation, Commun. Nonlinear Sci. Numer. Simul. 16(3) (2011) 1140–1153.

22. Miller, K. S., and Ross, B. *An introduction to the fractional calculus and fractional differential equations.* A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1993.

23. Możaryn, J., Petryszyn, J.,and Ozana, S. *PLC based fractional-order PID temperature control in pipeline: design procedure and experimental evaluation.* Meccanica 56(4) (2021) 855–871.

24. Naik, P. A., Zu, J., and Owolabi, K. M. *Global dynamics of a fractional order model for the transmission of HIV epidemic with optimal control.* Chaos Solitons Fractals 138 (2020), 109826, 24 pp.

25. Nemati, S., Lima, P. M., and Torres, D. F. *A numerical approach for solving fractional optimal control problems using modified hat functions.* Commun. Nonlinear Sci. Numer. Simul. 78 (2019), 104849, 14 pp.

26. Nemati, A., and Yousefi, S. A. *A numerical method for solving fractional optimal control problems using Ritz method.* : J. Comput. Nonlinear Dyn. 11(5) (2016) 1–7.

27. 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 [Harcourt Brace Jovanovich, Publishers], New York-London, 1974.

28. Sweilam, N. H., and Al-Ajami, T. M. *Legendre spectral-collocation method for solving some types of fractional optimal control problems.* J. Adv. Res., 6(3) (2015) 393–403.

29. Yari, A. *Numerical solution for fractional optimal control problems by Hermite polynomials.*J. Vib. Control, 27(5-6) (2021) 698–716.

How to cite this article
R. Naseri, A. Heydari and A.S. Bagherzadeh Using shifted Legendre orthonormal polynomials for solving fractional optimal control problems. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 513-532. doi: 10.22067/ijnao.2022.70466.1035.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 533–541
DOI:10.22067/ijnao.2022.73913.1081
https://ijnao.um.ac.ir/

How to cite this article
Research Article

On stagnation of the DGMRES method

F. Kyanfar

Abstract

Let A be an n-by-n matrix with index $\alpha > 0$ and $b \in \mathbb{C}^n$. In this paper, the problem of stagnation of the DGMRES method for the singular linear system $Ax = b$ is considered. We show that DGMRES(A, b, α) has partial stagnation of order at least k if and only if $(0, \ldots, 0)$ belongs to the the joint numerical range of matrices $\{B^{\alpha+1}, \ldots, B^{\alpha+k}\}$, where B is a compression of A to the range of A^α. Also, we characterize the nonsingular part of a matrices A such that DGMRES(A, b, α) does not stagnate for all $b \in \mathbb{C}^n$. Moreover, a sufficient condition for non-existence of real stagnation vectors $b \in \mathcal{R}(A^\alpha)$ for the DGMRES method is presented, and the DGMRES stagnation of special matrices are studied.

AMS subject classifications (2020): 65F10; 15A06; 15A60.

Keywords: Stagnation; DGMRES method; Singular systems.

1 Introduction

Let A be an n-by-n matrix with index α. The index is the size of the largest Jordan block of A corresponding to the zero eigenvalue. The Drazin inverse A^D of A is the unique n-by-n matrix that satisfies

$$AA^D = A^D A, \quad A^{\alpha+1}A^D = A^\alpha, \quad A^D A A^D = A^D.$$

Since A^D can be written as a polynomial in A [2, p. 186], there is a possibility of using Krylov subspace methods to find the Drazin inverse solution $A^D b$ to a possibly inconsistent linear system $Ax = b$. Such an algorithm, called DGMRES, developed by Sidi [7]. DGMRES has been considered in

* Corresponding author

Received 8 December 2021; revised 1 March 2022; accepted 15 March 2022

Faranges Kyanfar
Department of Applied Mathematics, Shahid Bahonar University of Kerman, Iran. e-mail: kyanfar@uk.ac.ir

several studies; see [1, 8]. This algorithm is similar to the GMRES algorithm developed by Saad and Schultz [6] for solving nonsingular linear systems. The stagnation of GMRES was studied in [3, 5, 10] and the stagnation of DGMRES was studied in [11].

Note that while the linear system $Ax = b$ may have no solution, if we multiply each side by A^α, then the linear system $A^{\alpha+1}x = A^\alpha b$ is consistent and has $x = A^D b$ as a solution. The DGMRES algorithm works as follows. Given an initial guess x_0, compute the initial residual $r_0 = b - Ax_0$. We will choose approximate solutions x_k, $k = 1, 2, \ldots, n - \alpha$, to be of the form x_0 plus a linear combination of vectors from the kth Krylov subspace

$$\mathcal{K}_k(A, A^\alpha r_0) = span\{A^\alpha r_0, \ldots, A^{\alpha+k-1} r_0\}, \tag{1}$$

such that the residual vector $r_k = b - Ax_k$ satisfies

$$\begin{aligned} \|A^\alpha r_k\| &= \min_{x \in \mathcal{K}_k(A, A^\alpha r_0)} \|A^\alpha (b - A(x_0 + x)\| \\ &= \min_{c_1, \ldots, c_k} \|A^\alpha (b - A(x_0 + c_1 A^\alpha r_0 + \cdots + c_k A^{\alpha+k-1} r_0))\| \\ &= \min_{c_1, \ldots, c_k} \|A^\alpha r_0 - c_1 A^{2\alpha+1} r_0 - \cdots - c_k A^{2\alpha+k} r_0))\|. \end{aligned} \tag{2}$$

The DGMRES terminates with the exact Drazin-inverse solution in at most $n - \alpha$ iterations (i.e., $\|A^\alpha r_{n-\alpha}\| = 0$) [7]. Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm for vectors and the spectral norm for matrices. Without loss of generality, we assume that $x_0 = 0$ and $\|A^\alpha r_0\| = \|A^\alpha b\| = 1$, because if $A^\alpha r_0 = 0$, then the DGMRES algorithm has the solution x_0 at the initial step, in other words, the DGMRES algorithm has no progress.

Definition 1. Let $\{A_1, A_2, \ldots, A_k\}$ be $n \times n$ matrices. The joint numerical range for $(A_1, A_2, \ldots, A_k)$ is defined and denoted by

$$W(A_1, A_2, \ldots, A_k) := \{(x^* A_1 x, x^* A_2 x, \ldots, x^* A_k x) : x \in \mathbb{C}^n, x^* x = 1\}.$$

Note that in Definition 1, if $k = 1$, then the joint numerical range coincide with the standard numerical range.

2 Partial stagnation of DGMRES

In this section, the problem of stagnation of the DGMRES algorithm for singular linear system $Ax = b$ is studied.

Definition 2. Let A be an n-by-n matrix with index α and a right-hand side vector $b \in \mathbb{C}^n$. We say that DGMRES (A, b, α) has partial stagnation of order k, if

$$\|A^\alpha r_0\| = \cdots = \|A^\alpha r_k\| > \|A^\alpha r_{k+1}\| \geq \cdots \geq \|A^\alpha r_{n-\alpha}\| = 0. \tag{3}$$

Also, if DGMRES (A, b, α) has partial stagnation of order $k = n - \alpha - 1$, then DGMRES (A, b, α) has complete stagnation. DGMRES (A, b, α) does not stagnate, if DGMRES (A, b, α) has not partial stagnation of any order.

In the following result, we state an equivalent definition for partial stagnation [11].

Lemma 1. Let A be an n-by-n matrix with index α and a right-hand side vector $b \in \mathbb{C}^n$. Then DGMRES (A, b, α) has partial stagnation of order at least k if and only if $A^\alpha b$ is perpendicular to $\text{span}\{A^{2\alpha+1}b, \ldots, A^{2\alpha+k}b\}$.

Proof. By using (2), we obtain that for all $1 \leq i \leq k$,

$$\|A^\alpha b\| = \min_{c_1,\ldots,c_i} \|A^\alpha b - c_1 A^{2\alpha+1} b - \cdots - c_i A^{2\alpha+i} b))\|.$$

Therefore, $A^\alpha b$ should be perpendicular to $\text{span}\{A^{2\alpha+1}b, \ldots, A^{2\alpha+k}b\}$. □

By using the Core-Nilpotent decomposition and QR decomposition, we obtain the following decomposition [1].

Let $A \in \mathbb{C}^{n\times n}$ with $\alpha = ind(A) > 0$. Then there exists a unitary matrix $Q \in \mathbb{C}^{n\times n}$ such that

$$A = Q \begin{bmatrix} B & * \\ 0 & N \end{bmatrix} Q^*, \tag{4}$$

where $B \in \mathbb{C}^{m\times m}$ is the compression of A to $\mathcal{R}(A^\alpha)$ and N is nilpotent with index α.

Theorem 1. Let $A \in \mathbb{C}^{n\times n}$ with index α be as in (4). Then there exists a vector $b \in \mathbb{C}^n$ such that DGMRES(A, b, α) has partial stagnation of order at least k if and only if $(0, \ldots, 0) \in W(B^{\alpha+1}, \ldots, B^{\alpha+k})$.

Proof. By Lemma 1, we know that the DGMRES(A, b, α) has partial stagnation of order at least k, if and only if $(A^\alpha b)^* A^{2\alpha+i} b = 0,\ i = 1, \ldots, k$. Then

$$(A^\alpha b)^*(A^{\alpha+i})(A^\alpha b) = 0, \qquad i = 1, \ldots, k. \tag{5}$$

By using (4) and (5), for $i = 1, \ldots, k$,

$$\begin{aligned}(A^\alpha b)^*(A^{\alpha+i})(A^\alpha b) &= (A^\alpha b)^* Q \begin{bmatrix} B^{\alpha+i} & * \\ 0 & N^{\alpha+i} \end{bmatrix} Q^*(A^\alpha b) \\ &= (Q^*(A^\alpha b))^* \begin{bmatrix} B^{\alpha+i} & * \\ 0 & 0 \end{bmatrix} Q^*(A^\alpha b) = 0. \end{aligned} \tag{6}$$

Define $z = \binom{z_1}{z_2} = Q^*(A^\alpha b)$, where $z_1 \in \mathbb{C}^m$. Since $0 \neq A^\alpha b \in \mathcal{R}(A^\alpha)$ and the last $n - m$ columns of Q form an orthonormal basis for the $\mathcal{R}(A^\alpha)^\perp$, we obtain that $z_2 = 0$ and hence $\|z_1\| = \|z\| = \|Q^*(A^\alpha b)\| = 1$. Therefore,

$$z^* \begin{bmatrix} B^{\alpha+i} & * \\ 0 & 0 \end{bmatrix} z = z_1^* B^{\alpha+i} z_1 = 0,\ i = 1, \ldots, k. \tag{7}$$

This means that $(0,\dots,0) \in W(B^{\alpha+1},\dots,B^{\alpha+k})$.

Conversely, assume that $(0,\dots,0) \in W(B^{\alpha+1},\dots,B^{\alpha+k})$. Then there exists a unit vector $z_1 \in \mathbb{C}^m$ such that $z_1^* B^{\alpha+i} z_1 = 0,\ i = 1,\dots,k$. Define $z = \binom{z_1}{0} \in \mathbb{C}^n$. Then (7) holds. We know that the first m columns of Q form an orthonormal basis for the range of A^α. Then $Qz = Q\binom{z_1}{0} \in \mathcal{R}(A^\alpha)$, and hence the equation $A^\alpha x = Qz$ has a solution $x = b$. Since $z = Q^*(A^\alpha b)$, by using (7)

$$(Q^*(A^\alpha b))^* \begin{bmatrix} B^{\alpha+i} & * \\ 0 & 0 \end{bmatrix} (Q^*(A^\alpha b)) = z_1^* B^{\alpha+i} z_1 = 0,\ i = 1,\dots,k.$$

Therefore, $(A^\alpha b)^*(A^{\alpha+i})(A^\alpha b) = (A^\alpha b)^*(A^{2\alpha+i} b) = 0,\ i = 1,\dots k$. This shows that $A^\alpha b$ is perpendicular to $A^{2\alpha+i} b,\ i = 1,\dots,k$. Then by Lemma 1, DGMRES(A, b, α) has partial stagnation of order at least k. □

3 Complete stagnation of DGMRES

Let A be an n-by-n matrix with index α and let $b \in \mathbb{C}^n$. By Definition 2, we know that DGMRES(A, b, α) has complete stagnation if

$$\|A^\alpha r_0\| = \cdots = \|A^\alpha r_{n-\alpha-1}\| > \|A^\alpha r_{n-\alpha}\| = 0. \tag{8}$$

In the following result, we show that $\|A^\alpha r_m\| = 0$.

Theorem 2. Let $A \in M_n(\mathbb{C})$ with index α be as in (4) and let $b \in \mathbb{C}^n$. Then $A^\alpha r_m = 0$, where m is the dimension of $\mathcal{R}(A^\alpha)$, the range of A^α.

Proof. The matrix $B \in M_m(\mathbb{C})$ is nonsingular, so by using the Cayley–Hamilton theorem, there exists a polynomial of degree at most $m-1$ say $p(x) = a_{m-1}x^{m-1} + \cdots + a_1 x + a_0$ such that $(B^{-1})^{\alpha+1} = p(B)$. Then by [2, p. 186] the Drazin inverse $A^D = A^\alpha p(A)$. Then

$$\begin{aligned} \|A^\alpha r_m\| =& \min_{x \in \mathcal{K}_m(A, A^\alpha b)} \|A^\alpha (b - Ax)\| \\ =& \min_{t_0,\dots,t_{m-1}} \|A^\alpha b - A^{2\alpha+1}(t_0 b + \cdots + t_{m-1} A^{m-1} b)\| \\ \le& \|A^\alpha b - A^{2\alpha+1}(a_0 b + \cdots + a_{m-1} A^{m-1} b)\| \\ =& \|A^\alpha b - A^{\alpha+1}[A^\alpha p(A)]b\| = \|(A^\alpha - A^{\alpha+1} A^D) b)\|. \end{aligned} \tag{9}$$

Since $A^{\alpha+1} A^D = A^\alpha$, we obtain that $\|A^\alpha r_m\| = 0$. □

Remark 1. Theorem 2 shows that the DGMRES method terminates at most after m iterations. Then the complete stagnation occurs if $m = n - \alpha$. This means that the nilpotent part N in (4) must be equal to the Jordan block of size α corresponding to zero eigenvalue, $N = J_\alpha(0)$.

4 Stagnation of real matrices

Let $A \in \mathbb{R}^{n\times n}$ with $\alpha = ind(A) > 0$. Then by the core-nilpotent and QR decompositions for real matrices, there exist an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$, an invertible matrix $B \in \mathbb{R}^{m\times m}$, and a nilpotent matrix $N \in \mathbb{R}^{n-m\times n-m}$ such that (4) holds. Let $A \in \mathbb{R}^{n\times n}$ and let $e \in \mathbb{R}^n$. Then easy computation shows that

$$e^T Ae = 0 \text{ if and only if } e^T(A + A^T)e = 0.$$

Let $A \in \mathbb{R}^{n\times n}$ be as in (4) with $\alpha = ind(A) > 0$. If we are looking for a real stagnation vector $e \in \mathcal{R}(A^\alpha)$, it is enough to consider the following polynomial system:

$$e^T(A^{\alpha+i} + (A^{\alpha+i})^T)e = 0, \qquad i = 1, 2, \ldots, k, \quad e^T e = 1. \tag{10}$$

Meurant [4, Theorem 2.2] presented a sufficient condition for non-existence of real stagnation vectors $b \in \mathbb{R}^n$ for the GMRES method. In the following result, we state a sufficient condition for non-existence of real stagnation vectors $b \in \mathcal{R}(A^\alpha)$ for DGMRES method.

Theorem 3. Let $A \in \mathbb{R}^{n\times n}$ with $\alpha = ind(A) > 0$ be as in (4) and let $B_i := B^i + (B^i)^T, \quad i = \alpha+1, \alpha+2, \ldots, \alpha+k$, where $k \leq m$ is a natural number. If there exist real scalars $\mu_i, \quad i = 1, 2, \ldots, k$ such that the matrix $\mu_1 B_{\alpha+1} + \cdots + \mu_k B_{\alpha+k}$ is a (positive or negative) definite matrix, then there is no real stagnation vector $e \in \mathcal{R}(A^\alpha)$.

Proof. Assume if possible there exist a real stagnation vector $e \in \mathcal{R}(A^\alpha)$. Then there exists $b \in \mathbb{R}^n$ such that $e = A^\alpha b$ and (5) holds. By using the notations $z = \binom{z_1}{z_2} = Q^T(A^\alpha b)$ with $\|z_1\| = 1$ in Theorem 1, we obtain that $z_1^T B^{\alpha+i} z_1 = 0, \ i = 1, \ldots, k$. By (10), $z_1^T(B^{\alpha+i} + (B^{\alpha+i})^T)z_1 = z_1^T B_{\alpha+i} z_1 = 0, \ i = 1, \ldots, k$, and hence $z_1^T(\mu_1 B_{\alpha+1} + \cdots + \mu_k B_{\alpha+k})z_1 = 0$. Since $\mu_1 B_{\alpha+1} + \cdots + \mu_k B_{\alpha+k}$ is (positive or negative) definite, we obtain that $z_1 = 0$, a contradiction with $\|z_1\| = 1$. □

Example 1. Let A be as in (4), where $B = \begin{bmatrix} 1 & 2 & 1 \\ 1 & -1 & 2 \\ 1 & 0 & -1 \end{bmatrix}$ and $N = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$. It is readily seen that $10B_2 + B_3 = \begin{bmatrix} 96 & 30 & 44 \\ 30 & 62 & -1 \\ 44 & -1 & 44 \end{bmatrix}$ is positive definite, where $B_2 = B^2 + (B^2)^T$ and $B_3 = B^3 + (B^3)^T$. Then by Theorem 3, there is no real stagnation vector.

5 Stagnation of special matrices

Let A be as in (4). If $m = 0$, then A is nilpotent with index α, which means that $A^\alpha = 0$, and hence $A^\alpha b = 0$ for all $b \in \mathbb{C}^n$. Then without loss of generality, we assume that $\|A^\alpha b\| = 1$ throughout this paper. Also, we assume that $m > 0$, which means that $B \in M_m(\mathbb{C})$ is invertible and A is not nilpotent. In this section, we are going to characterize all matrices $B \in M_m(\mathbb{C})$ such that DGMRES(A, b, α) does not stagnate, for all $b \in \mathbb{C}^n$ and unitary matrices $Q \in M_n(\mathbb{C})$.

The decomposition (4) is known as the core-nilpotent decomposition of A. Moreover, the matrix B is nonsingular. On the other hand, this decomposition is shown by $A = B \oplus N$.

Theorem 4. Let $B \in M_m(\mathbb{C})$ be an invertible matrix and let $N \in M_{n-m}(\mathbb{C})$ be a nilpotent matrix with index α. Then $B^{\alpha+1}$ is a scalar matrix if and only if DGMRES(A, b, α) does not stagnate for any $b \in \mathbb{C}^n$ and invertible $V \in M_n(\mathbb{C})$, where $A = V \begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix} V^{-1}$.

Proof. Assume that$B^{\alpha+1} = \lambda I_m$ is a scalar matrix, where $\lambda \neq 0$. Let $b \in \mathbb{C}^n$ be an arbitrary vector and let $V \in M_n(\mathbb{C})$ be an arbitrary invertible matrix. Assume that $V = QR$ is the QR decomposition of V. Then

$$\begin{aligned} A = V \begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix} V^{-1} =& Q \begin{bmatrix} R_1 & * \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} R_1^{-1} & * \\ 0 & R_2^{-1} \end{bmatrix} Q^* \\ =& Q \begin{bmatrix} R_1 B R_1^{-1} & * \\ 0 & R_2 N R_2^{-1} \end{bmatrix} Q^*. \end{aligned}$$

Note that $R_2 N R_2^{-1}$ is again a nilpotent matrix with index $\alpha > 0$ and that $R_1 B R_1^{-1} = \lambda I_m$ is a scalar matrix. Since $0 \notin W((R_1 B R_1^{-1})^{\alpha+1}) = \{\lambda^{\alpha+1}\}$, by Theorem 1, DGMRES$(A, b, \alpha)$ does not stagnate, for any $b \in \mathbb{C}^n$ and $V \in M_n(\mathbb{C})$.

Conversely, let DGMRES(A, b, α) do not stagnate for any $b \in \mathbb{C}^n$ and let $V \in M_n(\mathbb{C})$. Assume if possible $B^{\alpha+1}$ is not a scalar matrix. Then by [9, Theorem 3], there exists an invertible matrix $V_1 \in M_m(\mathbb{C})$ such that $0 \in W(V_1 B^{\alpha+1} V_1^{-1})$. Let $V_1 = Q_1 R_1$ be the QR decomposition of V_1. Define the matrix $V := \begin{bmatrix} V_1 & 0 \\ 0 & I_{n-m} \end{bmatrix}$ and the unitary matrix $Q := \begin{bmatrix} Q_1 & 0 \\ 0 & I_{n-m} \end{bmatrix}$. Then

$$A = V \begin{bmatrix} B & 0 \\ 0 & N \end{bmatrix} V^{-1} = Q \begin{bmatrix} R_1 B R_1^{-1} & 0 \\ 0 & N \end{bmatrix} Q^*.$$

Since $0 \in W(V_1 B^{\alpha+1} V_1^{-1}) = W(R_1 B^{\alpha+1} R_1^{-1})$, by Theorem 1, DGMRES$(A, b, \alpha)$ has a partial stagnation of order at least one, a contradiction. Then $B^{\alpha+1}$ is a scalar matrix. □

Zhou and Wei [11, Section 3] showed that for 2×2 matrices, the stagnation system has no relation with condition number of V and that the stagnation system always has a real root, where V is the Jordan transformation matrix of A. Indeed, in the following result, we show that for any 2×2 matrix A, DGMRES(A, b, α) does not stagnate for any Jordan transformation matrix $V \in M_2(\mathbb{C})$ and $b \in \mathbb{C}^2$.

Proposition 1. Let A be a nonzero singular 2×2 matrix with index $\alpha = 1$ and let $b \in \mathbb{C}^2$ be an arbitrary vector. Then DGMRES(A, b, α) does not stagnate.

Proof. The Jordan decomposition of 2-by-2 matrix A has the following form:

$$A = V \begin{bmatrix} \lambda & 0 \\ 0 & 0 \end{bmatrix} V^{-1}.$$

Then $B^2 = [\lambda^2]$ is a scalar matrix, and hence by Theorem 4, DGMRES(A, b, α) does not stagnate for any $b \in \mathbb{C}^2$. □

In the following example, we show that by changing the right-hand side vector b, the stagnation of DGMRES(A, b, α) will be removed.

Example 2. Let $A = B \oplus N$, where

$$B = \begin{bmatrix} 2.5300 & -0.4147 & -0.6717 & -0.3570 \\ -0.4147 & 1.7306 & 0.8017 & -0.4718 \\ -0.6717 & 0.8017 & -0.5233 & 0.5021 \\ -0.3570 & -0.4718 & 0.5021 & 1.2627 \end{bmatrix}, \quad \text{and} \quad N = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

By choosing the vector $b = \begin{bmatrix} -0.5291 & -0.1187 & -1.2012 & -0.5129 & 0 & 0 \end{bmatrix}^T$ as the right-hand side vector, DGMRES$(A, b, 2)$ has partial stagnation of order one (see Figure 1 (a)).

By choosing $\hat{b} = \begin{bmatrix} 0.2277 & 0.4357 & 0.3111 & 0.9234 & 0.4302 & 0.1848 \end{bmatrix}^T$, as a random vector, DGMRES$(A, \hat{b}, 2)$ does not stagnate (see Figure 1 (b)).

6 Conclusion

Let A be an n-by-n matrix with index $\alpha > 0$ and let $b \in \mathbb{C}^n$. A necessary and sufficient condition for partial stagnation of DGMRES(A, b, α) is obtained, and also for $A \in M_n(\mathbb{R})$, a sufficient condition for the non-existence of real stagnation vector $b \in \mathcal{R}(A^\alpha)$ is studied. Moreover, a characterize for matrices $A \in M_n(\mathbb{C})$ such that DGMRES(A, b, α) does not stagnate for every $b \in \mathbb{C}^n$ are considered.

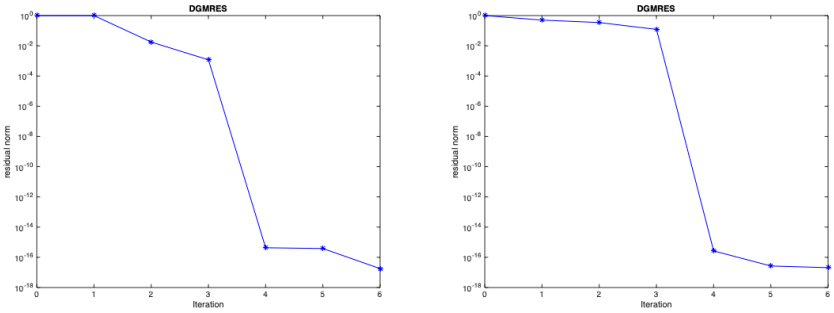


Figure 1: (a) DGMRES$(A, b, 2)$ (b) DGMRES$(A, \hat{b}, 2)$

Acknowledgement

The author would like to thank the anonymous referees for the careful reading and helpful comments to improve this paper.

References

1. Greenbaum, A., Kyanfar F. and Salemi, A. *On the convergence rate of DGMRES, Linear Algebra Appl.*, 552 (2018), 219–238.

2. Horn R.A. and Johnson, C.R. *Matrix analysis*, Second edition. Cambridge University Press, Cambridge, 2013.

3. Kyanfar, F., Mohseni Moghadam, M. and Salemi, A. *Complete stagnation of GMRES for normal matrices*, Comput. Appl. Math., 263 (2014), 417–422.

4. G. Meurant, *The complete stagnation of GMRES for* $n \leq 4$, Electron. Trans. Numer. Anal. 39 (2012), 75–101.

5. Meurant, G. *Necessary and sufficient conditions for GMRES complete and partial stagnation*, Appl. Numer. Math., 75 (2014), 100–107

6. Saad Y. and Schultz, M.H. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput. 7 (1986), 856–869.

7. Sidi, A. *DGMRES: A GMRES-type algorithm for Drazin-inverse solution of singular nonsymmetric linear systems*, Linear Algebra Appl., 335 (2001), 189–204.

8. Toutounian, F. and Buzhabadi, R. *New methods for computing the Drazin-inverse solution of singular linear systems*, Appl. Math. Comput. 294 (2017), 343–352.

9. Williams, J.P. *Similarity and the Numerical Range*, J. Math. Anal. Appl. 26 (1969), 307–314.

10. Zavorin, I., O'Leary, D.P. and Elman, H. *Complete stagnation of GMRES*, Linear Algebra Appl. 367 (2003), 165–183.

11. Zhou J. and Wei, Y. *Stagnation analysis of DGMRES*, Appl. Math. Comput. 151 (2004), 27–39.

How to cite this article
F. Kyanfar On stagnation of the DGMRES method. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 533-541. doi: 10.22067/ijnao.2022.73913.1081.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 542–566
DOI:10.22067/ijnao.2022.71302.1046
https://ijnao.um.ac.ir/

Research Article

Deception in multi-attacker security game with nonfuzzy and fuzzy payoffs

S. Esmaeeli, H. Hassanpour*, and H. Bigdeli

Abstract

There is significant interest in studying security games for defense optimization and reducing the effects of attacks on various security systems involving vital infrastructures, financial systems, security, and urban safeguarding centers. Game theory can be used as a mathematical tool to maximize the efficiency of limited security resources. In a game, players are smart, and it is natural for each player (defender or attacker) to try to deceive the opponent using various strategies in order to increase his payoff. Defenders can use deception as an effective means of enhancing security protection by giving incorrect information, hiding specific security resources, or using fake resources. However, despite the importance of deception in security issues, there is no considerable research on this field, and most of the works focus on deception in cyber environments. In this paper, a mixed-integer linear programming problem is proposed to allocate forces efficiently in a security game with multiple attackers using game theory analysis. The important subjects of information are their credibility and reliability. Especially when the defender uses deceptive defense forces, there are more ambiguity and uncertainty. Security game with Z-number payoffs is considered to apply both ambiguities in the payoffs and the reliability of earning these payoffs. Finally, the proposed method is illustrated by some numerical examples.

* Corresponding author

Received 5 July 2021; revised 15 April 2022; accepted 1 May 2022

S. Esmaeeli
Department of Mathematics, University of Birjand, Birjand, I.R. of Iran. e-mail: s.esmaeely@birjand.ac.ir

H. Hassanpour
Department of Mathematics, University of Birjand, Birjand, I.R. of Iran. e-mail: hhassanpour@birjand.ac.ir

H. Bigdeli
Researcher, Institute for the Study of War, Command and Staff University, Tehran, I.R. of Iran army. e-mail: hamidbigdeli92@gmail.com

AMS subject classifications (2020): 91Axx; 90C70; 90C29.

Keywords: Security game; Deceptive resource; Mixed-integer programming; Fuzzy theory; Z-number.

1 Introduction

Game theory has many applications in real-world problems, in many fields such as economics, military, politics, and so on (e.g., see [6, 2, 37]). In real-world game problems, we may encounter various types of uncertainty or inaccuracy in information (payoffs). Many researchers have studied game theory with different types of information ambiguity [3, 38, 39]. Seikh, Dutta, and Li [36] studied matrix games with rough interval payoffs and investigated two different solution methodologies to solve such a game. Karmakar, Seikh, and Castillo [24] developed a matrix game in a type-2 intuitionistic fuzzy environment. Bigdeli, Hassanpour, and Tayyebi [5] introduced two multiobjective linear programming problems to compute the optimistic and pessimistic values of fuzzy multiobjective games and their corresponding Pareto optimal strategies for each of the players by considering the concept of nearest interval approximation.

Security in maintaining military order and defense has always been a significant concern in human societies. In recent years, economic and political security has also become important. Limitations of resources such as money, personnel, and equipment have made it necessary to optimize the allocation of security resources. Security games have been successfully applied to solve many real-world security problems [1, 18, 26, 41]. They are also effective tools for arguing about the allocation of limited security resources and patrolling problems [1, 13, 25].

There has been a great deal of interest in research on game theory for security in airports, ports, transportation, and other infrastructures. Over the past decade, game theory has been used in various military sectors, computer network security applications, anti-ballistic missile defense systems, wildlife protection, and so on. Lye and Wing [27] proposed a game-theoretic method for analyzing security in computer networks. Brown et al. [8] described a new two-sided optimization model for planning the pre-positioning of defensive missile interceptors to counter an attacking threat. Conitzer and Sandholm [11] proposed a method to perform optimal random strategies in security games. Tarjom, Clempner, and Poznyak [42] used a method to calculate the Nash equilibrium in the case of one defender and several attackers. With respect to wildlife protection, Fang et al. [18] used repetitive interactions between rangers and hunters in protected areas to plan a patrol strategy that allowed rangers to collect hunting signals over time. Bigdeli, Hassanpour, and Tayyebi [7] proposed a model for solving a multiobjective security game with fuzzy payoffs and its application in a metro security system.

Most security games use the Stackelberg game because security forces typically commit to specific security policies to arrange their forces. Thus attackers are empowered to model their attacks under surveillance to take advantage of any potential weakness of the defender. Furthermore, the main assumption in Stackelberg security games is that limited security resources must be deployed strategically, considering differences in priorities of targets requiring security coverage and the responses of the adversaries to the security position (e.g., see [7, 4, 42, 43, 44]).

Previous studies assume the perfect surveillance of the defender's strategies despite the deceptions, while it is natural that if one of the players can deceive another, he will not hesitate. Defender's deceptive actions can affect the attacker's view of the defender's strategy, thus on the attacker's best response, and vice versa. Despite being relatively ignored in academia, in the military, deception is as old as war or politics. There are many examples of military deception in history. The story of the Trojan horse in Ancient Greece is perhaps the most famous ancient military deception. Also, in ancient China, many generals used to resort to deception ruses [30].

As a more recent example, World War II armies deceived their enemies by designing and building air tanks and wooden artillery. Thus, enemy forces would overestimate the enemy's defense capabilities and waste their ammunition or endanger their equipment. In another example, on a Japanese island in the Pacific Ocean, wicker planes deceived many American pilots. They spent a significant portion of their ammunition attacking unreal models by thinking only that the planes were real. For further study, in [12, 21, 19], there are numerous examples of deception in the First and Second World Wars.

Although research on deception in security games has increased in recent years, there is no noteworthy research in this field. Moreover, most authors focus on deception in cyber environments (e.g., see [14, 20, 28, 32, 40, 47]). Recently, deceptive methods have also been used to defend information systems. Cohen and Koike [10] provided a comprehensive discussion of deception to increase the security of information systems and concluded that "deception" is a positive factor for the defender and a negative factor for the attacker. In the security-military sector, Yin et al. [45] examined how fake resources and concealing the real resources of the defender might affect the attacker's beliefs and thus affect his best response. The authors [17] proposed a mathematical model to solve a security game in a fuzzy environment, in which the defender uses unrealistic resources when confronted with only one attacker. In the real world, it is important for players to have complete confidence in their information. Especially in situations where the defender uses deceptive defense forces, there are more ambiguity and uncertainty. Therefore, not only players can not accurately estimate their payoffs, but also they cannot be 100 % sure of these approximated estimates. Therefore, using fuzzy set theory in such games is necessary. There is no research on multi-attacker

security games with deceptive resources and fuzzy payoffs based on the best knowledge of the authors.

The security game has also been studied in [7, 17, 41]. In the multi-attacker security game solved in [41], the players' payoffs are considered to be crisp numbers. In [17], a security game problem in the fuzzy environment having only one attacker was solved. The fuzzy order used in [17] increases the number of constraints. In addition, the proposed method cannot be generalized to the case of multiple attackers. In [7], a multi-attacker security game with triangular fuzzy payoffs was solved, in which the authors considered the pessimistic situation and obtained an efficient solution for a cautious defender. In this paper, a security game problem with different types of attackers and different types of defense forces, such as real, secret, and fake, in a fuzzy environment is considered.

The remainder of the paper is organized as follows: In Section 2, some required concepts of fuzzy set theory are given. In Section 3, Stackelberg games are introduced, and the concept of efficient strategy in these games with multi-follower is defined. A security game with different types of attackers is introduced in Section 4. In Section 5, a security game problem is considered in which the defender's strategies can include deceptive protection covers, and a multiobjective optimization problem is proposed to obtain an efficient strategy for the defender. In Section 6, the players' payoffs are considered as Z-numbers, and a multiobjective optimization problem is proposed to get the efficient coverage of the defender when he uses deceptive resources. In Section 7, four numerical examples are provided to illustrate the proposed method. Finally, the conclusion is made in Section 8.

2 Basic concepts and definitions

In this Section, some concepts that are used in the paper are given.

Definition 1. A fuzzy set $\tilde{A}$ defined on a universe X is given as $\tilde{A} = \{(x, \mu_{\tilde{A}}(x))|x \in X\}$, where $\mu_{\tilde{A}} : X \longrightarrow [0,1]$ is the continuous membership function of $\tilde{A}$. The membership value $\mu_{\tilde{A}}(x)$ describes the degree of belongingness of $x \in X$ in $\tilde{A}$.

The support of a fuzzy set $\tilde{A}$ on X is defined by

$$supp(\tilde{A}) = \{x \in X \mid \mu_{\tilde{A}}(x) > 0\}.$$

A fuzzy number is a fuzzy set $\tilde{A}$ on the real line $\mathbb{R}$ with a continuous membership function $\mu_{\tilde{A}}$ that can be described as follows [15, 22]:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & \text{for all } x \in (-\infty, a_1], \\ f_A(x) & \text{for all } x \in [a_1, a_2], \\ 1 & \text{for all } x \in [a_2, a_3], \\ g_A(x) & \text{for all } x \in [a_3, a_4], \\ 0 & \text{for all } x \in [a_4, \infty), \end{cases} \tag{1}$$

where f_A represents a continuous and monotonically increasing function on $[a_1, a_2]$ and g_A is a continuous and monotonically decreasing function on $[a_3, a_4]$.

The α-level set of a fuzzy number $\tilde{A}$ is defined by the ordinary set $\tilde{A}_\alpha = \{x \in X |\, \mu_{\tilde{A}}(x) \geq \alpha\}$ for $\alpha \in (0, 1]$, and for $\alpha = 0$, $\tilde{A}_\alpha = cl\{x \in X | \mu_{\tilde{A}}(x) > 0\}$ where cl means closure of the set [9]. For $\alpha \in (0, 1]$, the α-level set of a fuzzy number is a closed and bounded interval, denoted as $\tilde{A}_\alpha = [f_A^{-1}(\alpha), g_A^{-1}(\alpha)]$, where $f_A^{-1}(\alpha) = inf\{x |\, \mu_{\tilde{A}}(x) \geq \alpha\}$ and $g_A^{-1}(\alpha) = sup\{x |\, \mu_{\tilde{A}}(x) \geq \alpha\}$.

Definition 2. [22] The expected interval of a fuzzy number $\tilde{A}$, denoted by $EI(\tilde{A})$, is defined as follows:

$$EI(\tilde{A}) = [\int_0^1 f_A^{-1}(\alpha) d\alpha, \int_0^1 g_A^{-1}(\alpha) d\alpha].$$

A fuzzy number $\tilde{A}$ on $\mathbb{R}$ is said to be a triangular fuzzy number if its membership function $\mu_{\tilde{A}} : \mathbb{R} \longrightarrow [0, 1]$ is

$$\mu_{\tilde{A}}(x) = \begin{cases} (x - a^1)/(a^2 - a^1), & a^1 \leq x \leq a^2, \\ (a^3 - x)/(a^3 - a^2), & a^2 \leq x \leq a^3, \\ 0, & otherwise, \end{cases} \tag{2}$$

where a^1 and a^3 represent the beginning and end points of the support of $\tilde{A}$, respectively, and a^2 is the median value (center).

The triangular fuzzy number defined above, is denoted by $\tilde{A} = (a^1, a^2, a^3)$. The addition of two triangular fuzzy numbers $\tilde{A} = (a^1, a^2, a^3)$ and $\tilde{B} = (b^1, b^2, b^3)$, and the multiplication of the triangular fuzzy number $\tilde{A}$ by $k \in \mathbb{R}$ using the extension principle of Zadeh [34] are obtained as follows:

$$\tilde{A} + \tilde{B} = (a^1, a^2, a^3) + (b^1, b^2, b^3) = (a^1 + b^1, a^2 + b^2, a^3 + b^3). \tag{3}$$

$$k\tilde{A} = \begin{cases} (ka^1, ka^2, ka^3), & k \geq 0, \\ (ka^3, ka^2, ka^1), & k < 0. \end{cases} \tag{4}$$

Proposition 1. [31] If $\tilde{A}$ is a triangular fuzzy number, then its expected interval can be computed as follows:

$$EI(\tilde{A}) = [\frac{1}{2}(a^1 + a^2), \frac{1}{2}(a^2 + a^3)].$$

Let $A = [A^L, A^R]$ and $B = [B^L, B^R]$ be two intervals. Then,

$$A + B = [A^L + B^L, A^R + B^R],\ A - B = [A^L - B^R, A^R - B^L], \quad (5)$$

$$\lambda A = \begin{cases} [\lambda A^L, \lambda A^R], & \lambda \geq 0, \\ [\lambda A^R, \lambda A^L], & \lambda < 0, \end{cases} \quad (6)$$

where λ is a real scalar.

Traditional fuzzy sets were developed to model the uncertainty made by human doubt when extracting information. However, the classical fuzzy sets do not account for the reliability of the obtained information. To overcome this limitation, Zadeh [46] proposed Z-numbers.

Definition 3. [23] A Z-number is an ordered pair of fuzzy numbers denoted as $Z = (\tilde{A}, \tilde{R})$. The first component $\tilde{A}$ is a restriction on the values which a real-valued uncertain variable Y can take. The second component $\tilde{R}$ is a measure of reliability for the first component.

In above definition, the membership function of the first component $\tilde{A}$, is $\mu_{\tilde{A}} : X \to [0, 1]$, where X is an arbitrary universal set and the membership function of the second component is $\mu_{\tilde{R}} : [0, 1] \to [0, 1]$.

In this paper, both parts of Z-numbers are considered to be triangular fuzzy numbers. To manipulate the problem involving Z-numbers, first, we convert Z-numbers to triangular fuzzy numbers in three steps, using the method presented by Kang et al. [23]. Consider a Z-number $Z = (\tilde{A}, \tilde{R})$.

Step 1. Convert the second component to a crisp number α as follows:

$$\alpha = \frac{\int_0^1 x\mu_{\tilde{R}}(x)dx}{\int_0^1 \mu_{\tilde{R}}(x)dx}. \quad (7)$$

Step 2. Use α as the weight of the first part (restriction). The weighted Z-number can be denoted as $\tilde{Z}^\alpha = \{(x, \mu_{\tilde{Z}^\alpha}(x))|\mu_{\tilde{Z}^\alpha}(x) = \alpha\mu_{\tilde{A}}(x), x \in X\}$.

Step 3. Convert the irregular fuzzy number (weighted restriction) to regular fuzzy number. The regular fuzzy set can be denoted as

$$\tilde{Z}' = \{(x, \mu_{\tilde{Z}'}(x))|\mu_{\tilde{Z}'}(x) = \mu_{\tilde{A}}(\frac{x}{\sqrt{\alpha}}), x \in \sqrt{\alpha}X\}.$$

Example 1. For the triangular fuzzy number $\tilde{A} = (a^1, a^2, a^3)$ by some simple calculations, one can see from (7) that

$$\alpha = \frac{a^1 + a^2 + a^3}{3}.$$

Let we have an uncertain variable, which takes the value of "almost 3" with the reliability of "almost 0.9". One can represents "almost 3" by the triangular fuzzy number $(2, 3, 4)$ (e.g.), and its reliability by the triangular fuzzy number $(0.8, 0.9, 1)$. Then we have the Z-number $Z = ((2, 3, 4), (0.8, 0.9, 1))$ to represent such an uncertainty. To handle such a Z-number payoff in our game problem, first we convert its reliability to a crisp number as follows:

$$\alpha = \frac{a^1 + a^2 + a^3}{3} = 0.9.$$

Then, we convert the weighted Z-number to triangular fuzzy number according to the proposed approach. So we have

$$\tilde{Z}' = (2\sqrt{0.9}, 3\sqrt{0.9}, 4\sqrt{0.9}) = (1.8974, 2.8461, 3.7948).$$

3 Stackelberg game

Stackelberg games, also known as the leader-follower games, were first introduced in 1952 by the German economist Van Stackelberg to model leadership and commitment. In Stackelberg games, the first player is the leader who chooses a strategy first, then the second player, called the follower, observes the leader's strategy and selects a counter-strategy accordingly. In other words, the game has two players and two stages. In stage 1, the leader's action set is $[0, \infty)$, whereas the follower's only available action is to "do nothing". In stage 2, the follower's action set is $[0, \infty)$, and the leader's only available action is to "do nothing". The problem in this game is to find the optimal strategy for the leader, assuming that the follower optimizes his payoff according to the logical observations that depend on the chosen strategy of the leader. The leader is committed to his decision, which means that if he selects a strategy, then he cannot change it. Therefore, to obtain Stacklerberg's solution, first, the maximum value of the follower's payoff is obtained for the various strategies of the leader. The payoff of the leader is optimized on the best response of the follower. The solution from the above process is called the Stackelberg solution, which can be calculated by the following bilevel linear programming problem[29]:

$$\begin{aligned} \max_{x} \quad & z_1(x, y) = c_1 x + d_1 y \\ & \text{where } y \text{ solves} \\ & \max_{y} \quad z_2(x, y) = c_2 x + d_2 y \\ & \text{subject to } Ax + By \leq b, \\ & \qquad\qquad x \geq 0,\ y \geq 0, \end{aligned} \tag{8}$$

where c_1 and c_2 are n_1-dimensional row coefficient vectors, d_1 and d_2 are n_2-dimensional row coefficient vectors, A is an $m \times n_1$, B is an $m \times n_2$ coefficient matrix, and b is an m-dimensional column constant vector. Moreover, $z_1(x, y)$ and $z_2(x, y)$, respectively, represent the payoff functions of the leader and follower, and x and y represent the strategy of the leader and the follower, respectively.

If the leader commits to the strategy x, the optimal solution $y^*(x)$ is obtained as the logical solution of the follower, by solving the low-level problem

of (8). Assuming that the follower gives a logical solution $y^*(x)$, the leader maximizes his objective function $z(x, y^*(x))$. In this case, the obtained solution is called the Stackelberg solution. This problem can be solved using bilevel programming method (see, e.g., [29]). In this paper, we use the Karush–Kuhn–Tucker (KKT) optimality conditions.

In a Stackelberg game with multi-followers, the leader has to maximize his payoff in the face of several types of followers. He has to choose a strategy to get the most payoff against all of the followers.

First, the followers choose their strategies, so each of them plays his best response. The leader must decide how to play against all of them in order to earn the highest possible payoff. He cannot play his best response against all the followers. Because if he plays his best against one of the followers, he may suffer a significant loss against another, which will reduce his final payoff. Therefore, to obtain Stacklerberg's solution, a multiobjective problem must be solved. Let us call this solution an efficient strategy, defined mathematically here.

Definition 4. Consider a Stackelberg game with p followers. Suppose that y^j is the chosen strategy of the follower type j and that x^j is the chosen strategy of the leader against the follower j. Let $U_l^j(x^j, y^j)$ and $U_f^j(x^j, y^j)$ be the payoffs of leader and follower type j, respectively, for the selected strategies. We call the strategy $x^* = (x^{1*}, x^{2*}, \ldots, x^{p*})$ the efficient strategy for leader, whenever (x^*, y^*) is an efficient solution of the following multiobjective programming problem

$$U_l(x^*, y^*) = \max_x (U_l^1(x^1, y^{1*}), \ldots, U_l^p(x^p, y^{p*})),$$

where y^{j*} represents the best response of the follower type j to the leader's x^j strategy.

4 Security game with multi-attackers

The security game precisely matches the Stackelberg game if we consider the defender as the leader and the attacker(s) as the follower(s). Thus, in this game, the defender commits to a strategy first. Then, the attackers optimize their payoffs, considering the action chosed by the defender. The defender must first commit to a strategy for placing his resources (manpower, equipment, ammunition, etc.) on targets. Then the attackers decide which targets they attack.

Let $T = \{1, \ldots, n\}$ be a set of targets, which may be attacked by p attackers, and assume that the defender has m security forces to protect the targets. The defender and each of the attackers, as the players of this game, try to earn the most payoffs. The attackers select targets that cause the most damage to the defender. On the other hand, the defender aims

to optimize resource assignments to minimize damage. Thus, each player has different strategies for achieving his goal. Each pure strategy of each attacker is to select a target to attack. The mixed strategy of attacker type j is $A^j = (a_1^j, \ldots, a_n^j)$, defined as follows:

$$a_t^j \geq 0, \quad \text{for all } t \in T, \quad \sum_{t=1}^{n} a_t^j = 1, \quad j = 1, \ldots p,$$

where a_t^j is the portion of the force of attacker type j used in attacking to the target t.

Each pure strategy of the defender is choosing a set of targets that have to be protected. If the defender considers only pure strategies, some targets may not be covered, and the attackers may use this weakness to attack them. Note that security resources are limited, and the defender may not be able to cover all the targets fully. Given the limited resources, we define the defender's mixed strategy as $C = (c_1, \ldots, c_n)$, where

$$0 \leq c_t \leq 1, \quad \text{for all } t \in T, \quad \sum_{t=1}^{n} c_t \leq m.$$

In fact, c_t is the amount of coverage of the target $t \in T$ and indicates the probability of the defender succeeding in preventing an attack on the target t. The constraint $0 \leq c_t \leq 1$ ensures that the amount of coverage of the target t have to be less than or equal to one unit of force required for the target t and to prevent force loss. The constraint $\sum_{t=1}^{n} c_t \leq m$ ensures that all of the allocated covers have not to be more than the number of available covering forces.

Suppose that defender and the attacker type j choose strategies C and A^j, respectively. The expected payoffs of the defender and the attacker type j, are

$$\begin{aligned} U_d^j(C, A^j) &= \sum_{t=1}^{n} a_t^j U_d^j(C, t), \quad j = 1, \ldots, p, \\ U_a^j(C, A^j) &= \sum_{t=1}^{n} a_t^j U_a^j(C, t), \quad j = 1, \ldots, p, \end{aligned} \tag{9}$$

if the target t is attacked by a_t^j unit of the force of attacker type j and covered by cover c_t, where

$$\begin{aligned} U_d^j(C, t) &= c_t U_d^{c,j}(t) + (1 - c_t) U_d^{u,j}(t), \\ U_a^j(C, t) &= c_t U_a^{c,j}(t) + (1 - c_t) U_a^{u,j}(t). \end{aligned} \tag{10}$$

In (10), $U_d^{c,j}(t)$ ($U_d^{u,j}(t)$) is defender's payoff when the target t is selected by attacker type j and covered (uncovered) by the defender. Similarly, $U_a^{c,j}(t)$ and $U_a^{u,j}(t)$ are defined for the attacker type j.

This security game, as a Stackelberg game, has several followers (attackers), wherein the defender first selects a strategy, and the attackers surveil the defender's actions. Each attacker tries to maximize his payoff by choosing a strategy that is the best response to the defender's fixed strategy. This is while the defender has to maximize his payoff against several types of attackers. He has to decide how to cover the various targets to get the most payoff. In other words, we are looking for an efficient strategy for the defender. The defender has to consider the set of best responses of attackers to each of his strategies.

An efficient strategy is obtained by solving the following bilevel multiobjective program:

$$
\begin{array}{lll}
(P_1) & Max & (U_d^1(C,A^1), U_d^2(C,A^2), \ldots, U_d^p(C,A^p)) \\
& s.t. & \sum_{t=1}^n c_t \le m, \\
& & 0 \le c_t \le 1, \quad t = 1, \ldots, n, \\
& & \left.\begin{array}{ll} & \textit{where } A^j \textit{ solves} \\ Max & U_a^j(C,A^j) \\ s.t. & \sum_{t=1}^n a_t^j = 1, \\ & a_t^j \ge 0,\ t = 1, \ldots, n, \end{array}\right\} j = 1, \ldots, p,
\end{array}
$$

where $U_d^j(C,A^j)$ and $U_a^j(C,A^j)$ for $j = 1, \ldots, p$ are given by (9).

Theorem 1. The bilevel multiobjective program (P_1) can be solved by solving the following multiobjective optimization problem:

$$
\begin{array}{lll}
(P_2) & Max & (U_d^1(C,A^1), U_d^2(C,A^2), \ldots, U_d^p(C,A^p)) \\
& s.t. & \sum_{t=1}^n c_t \le m, \qquad (11) \\
& & 0 \le c_t \le 1, \quad t = 1, \ldots, n, \qquad (12) \\
& & \left.\begin{array}{l} a_t^j \ge 0, \\ a_t^j \le M\delta_t^j, \\ \sum_{t=1}^n a_t^j = 1, \\ 0 \le k^j - (c_t U_a^{c,j}(t) + (1-c_t)U_a^{u,j}(t)) \le (1-\delta_t^j)M, \\ k^j \in \mathbb{R}, \delta_t^j \in \{0,1\}, \end{array}\right\} \begin{array}{l} j = 1, \ldots, p, \\ t = 1, \ldots, n, \end{array} \\
& & \qquad (13)
\end{array}
$$

where M is a large positive number, and $U_d^1(C,A^1), U_d^2(C,A^2), \ldots, U_d^p(C,A^p)$ are given by (9).

Proof. We prove that the constraints (13) are equivalent to the low-level problem of (P_1). By keeping C, the optimal policy of the defender fixed, the

optimization problem of attacker type j, which gives his best response to the defender's strategy C, is

$$\begin{array}{ll} Max & U_a^j(C, A^j) \\ s.t. & \sum_{t=1}^n a_t^j = 1, \\ & a_t^j \geq 0, \quad t = 1, \ldots, n, \end{array} \tag{14}$$

There is a scalar k^j that satisfies together with a_t^j the following KKT optimality conditions (Note that keeping C fixed, each low-level problem is a linear programming problem, for which the KKT conditions are necessary and sufficient for optimality):

$$\begin{aligned} & k^j \geq c_t U_a^{c,j}(t) + (1 - c_t) U_a^{u,j}(t), \qquad t = 1, \ldots, n, \\ & a_t^j (k^j - (c_t U_a^{c,j}(t) + (1 - c_t) U_a^{u,j}(t))) = 0, \qquad t = 1, \ldots, n, \\ & \sum_{t=1}^n a_t^j = 1, \\ & a_t^j \geq 0, \qquad t = 1, \ldots, n. \end{aligned} \tag{15}$$

By introducing the binary variables δ_t^j for $t = 1, \ldots, n$, and M as a large positive number, the constraints (15) are equivalently written as follows:

$$\begin{aligned} & a_t^j \leq M \delta_t^j, \qquad t = 1, \ldots, n, \\ & 0 \leq k^j - (c_t U_a^{c,j}(t) + (1 - c_t) U_a^{u,j}(t)) \leq (1 - \delta_t^j) M, \qquad t = 1, \ldots, n, \\ & \sum_{t=1}^n a_t^j = 1, \\ & a_t^j \geq 0, \qquad t = 1, \ldots, n. \end{aligned} \tag{16}$$

□

If the defender knows that each attacker attacks at most one target, the constraints (13) can be equivalently replaced by the following constraints:

$$\left.\begin{array}{l} a_t^j \in \{0, 1\}, \\ \sum_{t=1}^n a_t^j = 1, \\ 0 \leq k^j - (c_t U_a^{c,j}(t) + (1 - c_t) U_a^{u,j}(t)) \leq (1 - a_t^j) M, \end{array}\right\} \begin{array}{l} j = 1, \ldots, p, \\ t = 1, \ldots, n. \end{array} \tag{17}$$

A solution to the multiobjective programming problem (P_2) is an efficient strategy for the defender in the security game with multiple attackers.

There are several methods to get an efficient solution to problem (P_2) (e.g., see [16, 34]). In Section 7, we use the weighted sum method. The weights

of the objective functions in problem (P_2), depending on the importance of them for the defender, can be determined by consultation with experts or using methods such as AHP* and TOPSIS**.

5 Deception in multi-attacker security game

In a security game, depending on the available budget, a defender can use deceptive resources to increase his payoff or to reduce the attackers' desire to attack targets. For example, in military operations, the security of various urban or regional centers, different political ceremonies, and so on, defense forces use some types of covert resources and some types of fake ones. Depending on the type of protected targets, the defender uses different deceptive resources, with different probability of deception failure. For example, hidden cameras for protected targets, secret police forces, air marshals on the flight lines, and fake resources are some deceptive resources. The defense force must be able to have the best arrangement of these resources against the attackers according to the available budget. In this section, we consider m real forces and two kinds of deceptive resources: the first kind has a positive effect on the defender's payoff. For example, secret forces have positive effects on the defender's payoff because they have defensive power. The second kind has no effect on the defender's payoff and only reduces the attackers' desire to attack targets. These kinds of deceptive resources do not affect the defense of a target, but they can at least disturb the view of the attackers, and they can reduce the intensity of their attacks. For example, fake resources cause errors in the attacker's observations but do not have defensive power. Therefore they do not increase the defender's payoff. We denote the set of the first (second) kind of deceptive resources by D_1 (D_2). Accordingly, the defender's payoff for a target $t \in T$ is

$$U_d^j(C,t) = c_t U_d^{c,j}(t) + (1-c_t)U_d^{u,j}(t) + \sum_{i \in D1} (c_{t,i} U_d^{cij}(t) + (1-c_{t,i})U_d^{uij}(t)). \tag{18}$$

In (18), for $i \in D_1$ and $t \in T$, $c_{t,i}$ is the amount of deceptive resource coverage, and $U_d^{cij}(t)$ ($U_d^{uij}(t)$) is the defender's payoff from deceptive resource coverage (uncoverage) i against the attacker type j. Note that, obviously, the defender's payoff from using a deceptive resource $i \in D_1$ and a real resource are not the same necessarily. Also, obviously $c_t + \sum_{i \in D_1} c_{t,i} \leq 1$ because more coverage for the target t is useless for the defender. If the importance of a real cover unit differs from a deceptive cover unit, then the mentioned constraint is changed to $wc_t + \sum_{i \in D_1} w_i c_{t,i} \leq 1$, where w and w_i are the weights of real cover and each unit cover type i, respectively.

* Analytic Hierarchy Process

** Technique for Order of Preference by Similarity to Ideal Solution

To determine the attackers' payoffs, we look at the amount of coverage that they observe and their reaction. Using deceptive coverage resources by the defender is not always 100 percent successful in deceiving the attackers. It is natural that each of them has a failure probability. Suppose that r_i is the probability of deceptive resource's failure for $i \in D_1 \cup D_2$. If the vector of defender choices is $C = (c_1 + \sum_{i\in D_1\cup D_2} c_{1,i}, \ldots, c_n + \sum_{i\in D_1\cup D_2} c_{n,i})$, then the attacker's observation is $E = (e_1, \ldots, e_n)$ in which

$$e_t = c_t + \sum_{i\in D_1\cup D_2} r_i c_{t,i}, \qquad t = 1, \ldots, n. \tag{19}$$

It is assumed that the failure probability of the deceptive resource depends only on its structure. Therefore, the failure probability of one type of deceptive resource is the same for all attackers. Then the payoff of attacker type j is

$$\begin{aligned} U_a^j(E,t) =& c_t U_a^{c,j}(t) + (1-c_t)U_a^{u,j}(t) \\ &+ \sum_{i\in D_1\cup D_2} (r_i c_{t,i} U_a^{cij}(t) + (1-r_i)(1-c_{t,i})U_a^{uij}(t)), \quad t = 1, \ldots, n. \end{aligned} \tag{20}$$

In (20), $U_a^{cij}(t)$ ($U_a^{uij}(t)$) is the payoff of attacker type j in attacking to the target t with (without any) deceptive resource coverage i.

Now, suppose that the defender's budget to create deceptive resources is B, and that he can purchase deceptive resource type i at the cost of B_i per unit. Then to obtain an efficient defense strategy, he has to consider the following constraints:

$$\sum_{t=1}^{n} \sum_{i\in D_1\cup D_2} B_i c_{t,i} \le B. \tag{21}$$

Based on the above discussion, the efficient strategy of the defender is obtained by solving the following multiobjective mixed-integer linear program:

$$\begin{array}{lll} (P_3) & Max & (U_d^1(C,A^1), U_d^2(C,A^2), \ldots, U_d^p(C,A^p)) \\ & s.t. & \sum_{t=1}^{n} c_t \le m, \\ & & 0 \le c_t \le 1 \qquad t = 1, \ldots, n, \\ & & c_t + \sum_{i\in D_1} c_{t,i} \le 1, \qquad t = 1, \ldots, n, \\ & & \sum_{t=1}^{n} \sum_{i\in D_1\cup D_2} B_i c_{t,i} \le B, \\ & & \left.\begin{array}{l} \sum_{t=1}^{n} a_t^j = 1, \\ a_t^j \ge 0, \\ a_t^j \le M\delta_t^j, \\ 0 \le k^j - U_a^j(E,t)) \le (1-\delta_t^j)M, \\ k^j \in \mathbb{R}, \delta_t^j \in \{0,1\}, \end{array}\right\} \begin{array}{l} j = 1, \ldots, p, \\ t = 1, \ldots, n, \end{array} \end{array}$$

where M is a large positive number.

6 Deception in multi-attacker security game in a fuzzy environment

In the real world, the information in a security game is often vague due to the lack of sufficient evidence. For example, a defender may not accurately identify any type of attacker, and attackers may not recognize and/or control different kinds of deceptive resources. Even if they know to some extent what the deceptive resource is, they cannot be 100% sure of what they have seen. In this situation, showing the payoffs in the form of Z-numbers is an appropriate suggestion for expressing ambiguity. The first component represents the player's payoff from selecting a strategy, and the second component shows the measure of the reliability of this selection. In our study, both components of Z-numbers are considered to be triangular fuzzy numbers. For a strategy profile (C, A), the payoff of attacker type j is $\overline{U}_a^j = (\tilde{U}_a^j, \tilde{R}_a^j)$, wherein $\tilde{U}_a^j$ and $\tilde{R}_a^j$ represent the payoff of attacker type j and the reliability of earning this payoff, respectively. The same definition is applied to the defender, and his payoff against attacker type j is denoted by $\overline{U}_d^j = (\tilde{U}_d^j, \tilde{R}_d^j)$. To solve the problem, we convert the Z-numbers to triangular fuzzy numbers by the procedure described in Section 2. Finally, considering the described conversion, we have a triangular fuzzy number for each player's payoff.

Now we have the following programming problem, in which some parameters are triangular fuzzy numbers:

$$
\begin{aligned}
(P_4)\quad Max \quad & (\tilde{U}_d^1(C, A^1), \tilde{U}_d^2(C, A^2), \ldots, \tilde{U}_d^p(C, A^p)) \\
s.t. \quad & \sum_{t=1}^{n} c_t \leq m, \\
& 0 \leq c_t \leq 1, \qquad t = 1, \ldots, n, \\
& c_t + \sum_{i \in D_1} c_{t,i} \leq 1, \qquad t = 1, \ldots, n, \\
& \sum_{t=1}^{n} \sum_{i \in D_1 \cup D_2} B_i c_{t,i} \leq B, \\
& \left.\begin{array}{l} \sum_{t=1}^{n} a_t^j = 1, \\ 0 \leq a_t^j \leq M\delta_t^j, \\ \tilde{U}_a^j(E, t) \leq \tilde{k}^j, \\ \tilde{k}^j \leq (1 - \delta_t^j)\tilde{M} + \tilde{U}_a^j(E, t), \\ k^j \in \mathbb{R}, \qquad \delta_t^j \in \{0, 1\}, \end{array}\right\} \begin{array}{l} j = 1, \ldots, p, \\ t = 1, \ldots, n. \end{array}
\end{aligned}
$$

To solve the problem $(P4)$, let for $s = a, d$ and $j = 1, \ldots, p$, $EI(\tilde{U}_s^j(C, A^j)) = [U_s^{jL}(C, A^j), U_s^{jR}(C, A^j)]$ and $EI(\tilde{k}) = [k^{jL}, k^{jR}]$ be the expected intervals corresponding to fuzzy numbers $\tilde{U}_s^j(C, A^j)$ and $\tilde{k}$, which are calculated according to Proposition 1. Then problem (P_4) is transformed into the following

interval programming problem:

$$
\begin{aligned}
(P_5)\ Max \quad & ([U_d^{1L}(C,A^1), U_d^{1R}(C,A^1)], \ldots, [U_d^{pL}(C,A^p), U_d^{pR}(C,A^p)]) \\
& s.t. \\
& \textstyle\sum_{t=1}^{n} c_t \le m, \\
& 0 \le c_t \le 1, && t = 1, \ldots, n, \\
& c_t + \textstyle\sum_{i \in D_1} c_{t,i} \le 1, && t = 1 \cdots, n, \\
& \textstyle\sum_{t=1}^{n} \sum_{i \in D_1 \cup D_2} B_i c_{t,i} \le B, \\
& \textstyle\sum_{t=1}^{n} a_t^j = 1, && j = 1, \ldots, p, \\
& 0 \le a_t^j \le M\delta_t^j, && j = 1, \ldots, p, \ t = 1, \ldots, n, \\
& [U_a^{jL}(E,t)), U_a^{jR}(E,t))] \le [k^{jL}, k^{jR}], && j = 1, \ldots, p, \ t = 1, \ldots, n, \\
& [k^{jL}, k^{jR}] \le (1 - \delta_t^j)[M, M] + [U_a^{jL}(E,t)), U_a^{jR}(E,t))], && j = 1, \ldots, p, \ t = 1, \ldots, n, \\
& k^{jL}, k^{jU} \in \mathbb{R}, \qquad \delta_t^j \in \{0,1\}, && j = 1, \ldots, p.
\end{aligned}
$$

There are several methods for solving (P_5). In most of them, the main idea is based on intervals' comparison. Instead, Saati, Memariani, and Jahanshahloo [33] proposed a new approach in which a variable is defined corresponding to each interval so that it maximizes the objective functions while satisfying the constraints. More clearly, to solve problem (P_5), we solve the following problem:

$$
\begin{aligned}
(P_6)\ Max\ & (u_1, \ldots, u_p) \\
& s.t.\ \textstyle\sum_{t=1}^{n} c_t \le m, \\
& 0 \le c_t \le 1, \qquad t = 1, \ldots, n, \\
& c_t + \textstyle\sum_{i \in D_1} c_{t,i} \le 1, \qquad t = 1, \ldots, n, \\
& \textstyle\sum_{t=1}^{n} \sum_{i \in D_1 \cup D_2} B_i c_{t,i} \le B, \\
& U_d^{jL}(C,A^j) \le u_j \le U_d^{jR}(C,A^j), \qquad j = 1, \ldots, p, \\
& U_a^{jL}(E,t)) \le v_j \le U_a^{jk}(E,t)), \qquad j = 1, \ldots, p, \\
& k^{jL} \le k_j \le k^{jU}, \qquad j = 1, \ldots, p \\
& \left.\begin{aligned}
\textstyle\sum_{t=1}^{n} a_t^j &= 1, \\
0 \le a_t^j &\le M\delta_t^j, \\
v_j &\le k_j, \\
k_j &\le (1 - \delta_t^j)M + v_j, \\
k^{jL}, k^{jU} \in \mathbb{R}, &\qquad \delta_t^j \in \{0,1\},
\end{aligned}\right\} \begin{array}{l} j = 1, \ldots, p, \\ t = 1, \ldots, n, \end{array}
\end{aligned}
$$

in which

$$
\begin{aligned}
& u_j \in [U_d^{jL}(C,A^1), U_d^{1R}(C,A^j)], && j = 1, \ldots, p, \\
& v_j \in [U_a^{jL}(E,t)), U_a^{jk}(E,t))], && j = 1, \ldots, p, \\
& k_j \in [k^{jL}, k^{jU}], && j = 1, \ldots, p.
\end{aligned}
$$

In fact, by solving problem (P_6), the best choices of the variables u_j, v_j, and k_j are determined from their corresponding intervals so that both maximize the objective functions and satisfy the constraints.

Now, once again, we have a multiobjective problem with crisp parameters. There are several methods to get an efficient solution to this problem (e.g., see [16, 34]). In the solved examples in Section 7, we use the weighted sum method.

Remark 1. The proposed method was extended to solve a multi-attacker security game having Z-numbers payoffs. However, it can be used if the payoffs are triangular fuzzy numbers or real numbers as well. In the first case, steps 1-3 in Section 2 to convert Z-numbers to triangular fuzzy numbers are removed, and in the second case, we have to solve the problem (P_3).

Remark 2 (Comparison with similar works)**.** As mentioned in Remark 1, our method can also be used to solve a security game with triangular fuzzy payoffs. Such a problem was also considered in [7]. Bigdeli, Hassanpour, and Tayyebi [7] have used a pessimistic approach to solve the problem, but our method solves the problem without considering a pessimistic or optimistic point of view. Therefore it is natural to obtain different solutions by the two methods. Furthermore, there is no significant difference between the two methods in view of computational complexity. Therefore, in a security game with triangular fuzzy payoffs, a pessimistic decision-maker can use the method of [7]. The special feature of our work is that we have considered a security game with Z-numbers payoffs and deceptive resources, but in [7], it did not cover these issues.

7 Numerical examples

In this section, we give four examples. In the first example, the defender uses only real resources. In the second example, the defender uses three types of deceptive resources: one fake and two types of secret resources. In both examples, the players' payoffs are considered to be real numbers. In the third example, the defender uses two types of deceptive resources, and the players' payoffs are Z-numbers. The final example is an example solved in [7]. We solve it by our method and compare the solutions obtained from the two methods. All of the optimization problems in examples were solved by Lingo software.

Example 2. In a security game, suppose that three attackers intend to attack four targets and that a defender has $m = 2$ forces to protect these targets. The players' payoffs are given in Tables 2–4. The weights assigned to the tables are 0.2, 0.3, and 0.5, respectively.

By solving the problem (P_2) by the weighted sum method, the following efficient strategy is obtained:

Table 1: Game matrix of defender and attacker type 1 in Example 2

	target 1		target 2		target 3		target 4	
	covered (c)	uncovered (u)	c	u	c	u	c	u
defender	1.5	-0.5	5	-6	2	-1	9	-8
attacker	-1.5	2	-4	5	-2	3	-4	9

Table 2: Game matrix of defender and attacker type 2 in Example 2

	target 1		target 2		target 3		target 4	
	c	u	c	u	c	u	c	u
defender	2	-0.5	6	-5	3	-2	11	-10
attacker	-1	1	-3	4	-2	3	-4	8

$$C = (0.34, 0.55, 0.40, 0.62).$$

Since the defender has two covering resources (two defense forces), it is concluded that 17, 27.5, 20, and 31 percent of the forces should be assigned to the targets t_1, t_2, t_3, and t_4, respectively, and 4.5 % of defense forces are not assigned.

As the tables show, the target t_4 has greater payoffs for the defender than the other targets. Also, for all three attackers, this target has greater payoffs than the other targets. Therefore, it is more likely to attack this target. In the case of the target t_1 is the opposite. In the solution obtained by our method, the highest coverage was obtained for the target t_4, and the lowest coverage was achieved for the target t_1.

Example 3. Consider a security game in which three attackers intend to attack four targets. The defender has $m = 1$ real security force to protect the targets. Decision-makers (experts) have provided the following information: The defender uses three types of deceptive resources. He uses an experienced and trained secret force with a 0.2 probability of being exposed. At the same time, a real force acts as a covert force with a lower cost and 0.4 probability of being exposed (secret normal force). The payoff of an experienced secret force is 1.3 times more than that of a real security force. The probability that the attacker will not distinguish these fake resources is 0.4 (i.e., his failure probability is 0.6). The required budget for each deceptive force unit is 1,

Table 3: Game matrix of defender and attacker type 3 in Example 2

	target 1		target 2		target 3		target 4	
	c	u	c	u	c	u	c	u
defender	1	-0.5	6	-4.5	3	-1	10	-9
attacker	-1	0.5	-4	5	-3	4	-6	10

Table 4: Game matrix of defender and attacker type 1 in Example 3

	cover's type	target 1		target 2		target 3		target 4	
		c	u	c	u	c	u	c	u
defender	real	5	-3	8	-9	2	-2.5	3	-5
attacker	real/secret normal force	-2	3	-4	6	-3	5	-4	5
	experienced secret force	-3	3	-5	6	-4	5	-5	5
	fake	3	-3	6	-6	5	-5	3	-5

Table 5: Game matrix of defender and attacker type 2 in Example 3

	cover's type	target 1		target 2		target 3		target 4	
		c	u	c	u	c	u	c	u
defender	real	4	-1	10	-7	1.5	-1	2	-2.5
attacker	real/secret normal force	-3	2.5	-2	1.5	-2	1	-3	1
	experienced secret force	-4	2.5	-3	1.5	-2	1	-1	1
	fake	2.5	-2.5	1.5	-1.5	1	1	2	-1

3, and 7, respectively, for fake, secret normal, and experienced secret force, and the defender's available budget is 12. The players' payoffs are given in Tables 4–6.

Solving the problem (P_3) by weighted sum method with equal weights for the objective functions yields the solution given in Table 3.

This means that in order to protect four targets with the mentioned security resources, the defender must plan the presence of the real security resource with 42% in the target 1, 50% in the target 3, and 7% in the target 4. The target 2 does not require a real resource, and it is sufficient to be protected by an experienced secret force unit and 0.79 fake force unit. Likewise, the defender must deploy other deceptive security resources according to Table 3.

Example 4. Consider a security game with three targets and two attackers. The defender uses $m = 1$ real security force and two secret sources to protect the targets. Secret forces are exposed to 0.3 probability. The required budget for a secret force unit is 5, and the defender's available budget is 9. The value of each unit of secret force is 1.5 times that of a real force unit. The players' payoffs are Z-numbers given in Tables 8 and 9.

Table 6: Game matrix of defender and attacker type 3 in Example 3

	cover's type	target 1		target 2		target 3		target 4	
		c	u	c	u	c	u	c	u
defender	real	5	-2	6	-4	3	-1	4	-3
attacker	real/secret normal force	-3	3	-2	5	-2	4	-1	1
	experienced secret force	-3	3	-2	5	-2	4	-1	1.5
	fake	3	-3	5	-4	4	-5	1	-2

Table 7: Amounts of targets coverages in Example 3

	i=real	i=experienced secret force	i= secret normal force	i= fake
t=1	0.42	0	0	0
t=2	0	1	0	0.79
t=3	0.5	0	0.5	0.2
t=4	0.07	0	0.5	0

Table 8: Game matrix of defender and attacker type 1 in Example 4

		defender		attacker type 1	
		c	u	c	u
t_1	real	((6,6,7), (0.8,.0.9,1))	((-3,-2,-2), (0.8,.0.9,1))	((-3,-3,-2), (0.8,.0.9,1))	((2,3,4), (0.7,0.8,.0.9))
	secret	((3,3,4), (0.6,0.7,0.8))	((-2,-2,-1), (0.6,0.7,0.8))	((-4,-3,-2), ,(0.6,0.7,0.8))	((1,2,3), (0.6,0.7,0.8))
t_2	real	((6,6,7), (0.7,0.8,0.9))	((-2,-1.5,-1), ,(0.7,0.8,0.9))	((-5,-4,-3), (0.7,0.8,0.9))	((2,3,3), (0.7,0.8,0.9))
	secret	((3,4,5), (0.6,0.7,0.8))	((-2,-1,-1), (0.6,0.7,0.8))	((-2,-2-1), (0.6,0.7,0.8))	((1,2,3), (0.6,0.7,0.8))
t_3	real	((2,4,4), (0.8,.0.9,1))	((-1.5,-1,-1), (0.8,.0.9,1))	((-3,-2,-1), (0.8,.0.9,1))	((1,2,2), (0.8,.0.9,1))
	secret	((2,2,3), (0.6,0.7,0.8))	((-3,-2,-1), (0.6,0.7,0.8))	((-2,-2,-1), (0.6,0.7,0.8))	((1,2,3), (0.6,0.7,0.8))

Table 9: Game matrix of defender and attacker type 2 in Example 4

		defender		attacker type 2	
		c	u	c	u
t_1	real	((5,5,6), (0.8,.0.9,1))	((-2,-2,-1), (0.8,.0.9,1))	((-2,-2,-1), (0.8,.0.9,1))	((1,2,3), (0.8,.0.9,1))
	secret	((5,6,6), (0.6,0.7,0.8))	((-3,-2,-1), (0.6,0.7,0.8))	((-4,-3,-2), (0.6,0.7,0.8))	((1,2,3), ,(0.6,0.7,0.8))
t_2	real	((4,4,5), (0.7,0.8,0.9))	((-1,-0.5,0), (0.7,0.8,0.9))	((-2,-1,-1), (0.7,0.8,0.9))	((2,2,3), (0.7,0.8,0.9))
	secret	((5,6,6), (0.6,0.7,0.8))	((-2,-2,-1), (0.6,0.7,0.8))	((-2,-2,-1), (0.6,0.7,0.8))	((2,3,4), (0.6,0.7,0.8))
t_3	real	((3,3,4), (0.7,0.8,0.9))	((-3,-2,-2), (0.7,0.8,0.9))	((-4,-3,-3), (0.7,0.8,0.9))	((1,2,4), (0.7,0.8,0.9))
	secret	((3,3,4), (0.6,0.7,0.8))	((-1.5,-1,-0.5), (0.6,0.7,0.8))	((-3,-2,-1), (0.6,0.7,0.8))	((1,3,4), (0.6,0.7,0.8))

Table 10: Game matrix of defender and attacker type 1 as triangular fuzzy numbers in Example 4

		defender		attacker type 1	
		c	u	c	u
t_1	real	(5.75,5.75,6.6)	(-2.8,-1.8,-1.8)	(-2.8,-2.8,-1.9)	(1.8,2.8,3.8)
	secret	(2.5,2.5,3.3)	(-1.6,-1.6,-0.8)	(-3.3,-2.5,-1.67)	(0.8,1.6,2.5)
t_2	real	(5.3,5.3,6.2)	(-1.78,-1.2,-0.8)	(-4.4,-3.5,-2.6)	(1.78,2.68,2.68)
	secret	(3.3,4.1,4.1)	(-1.6,-0.8,-0.8)	(-1.67,-1.67,-0.8)	(0.8,1.67,2.5)
t_3	real	(1.8,3.8,3.8)	(-1.2,-0.8,-0.8)	(-2.8,-1.9,-0.9)	(0.94,1.9,1.9)
	secret	(1.6,1.6,2.5)	(-2.5,-1.6,-0.8)	(-1.67,-1.67,-0.83)	(0.8,1.67,2.5)

Table 11: Game matrix of defender and attacker type 2 as triangular fuzzy numbers in Example 4

		defender		attacker type 2	
		c	u	c	u
t_1	real	(4.7,4.7,5.6)	(-1.7,-1.7,-0.8)	(-2.84,-1.89,-0.94)	(0,1.89,2.84)
	secret	(4.2,5,5)	(-2.5,-1.6,-0.8)	(-1.67 ,-0.83,-0.83)	(0.83,1.67,2.5)
t_2	real	(3.5,3.5,4.4)	(-0.8,-0.4,0)	(-1.7,-0.8,-0.8)	(1.78,2.68,2.68)
	secret	(4.1,4.1,5)	(-1.7,-0.8,-0.8)	(-1.67,-1.67,-0.83)	(1.67,2.5,3.34)
t_3	real	(2.5,2.5,3.3)	(-2.6,- 1.8,-1.8)	(-3.5,-2.68,-2.68)	(0.89,1.78,3.57)
	secret	(2.5,2.5,3.3)	(-1.2,-0.8,-0.4)	(-2.5,-2.5,-1.67)	(0.83,2.5,3.3)

Now, for the given player's payoffs, we calculate the $\sqrt{\alpha}$ values from (7), and apply them as the weights of payoffs. Then we have triangular fuzzy payoffs given in Tables 10 and 1.

Solving the problem (P_6) by the weighted sum method (with equal weights for the objective functions) for these data yields the following solution:

$$C_{real} = (0.52, 0.23, 0.04),\ C_{secret} = (0, 0.83, 0.97).$$

This means that the defender should allocate 52%, 23%, and 4% of his real forces to the targets 1, 2, and 3, respectively. Because of the constraint $\sum_{t=1}^{n} c_t \leq m$, not all resources will necessarily be allocated in the optimal solution. In this example, 79% of the real resources are used and 21% of them remain unused. Also, with the available budget, he can allocate 41.5% of the two secret forces (i.e., 0.83 of the two units) to the target 2 and 48.5% (i.e., 0.97 of the two units) to the target 3.

Example 5. Consider the security game with two targets and three attackers solved in [7]. The players' payoffs are given in Tables 12–14.

Solving this example by our method (Problem P_6, without deceptive resources) yields the payoff 4.7 and the cover $C = C_{real} = (0.05, 0.95)$. This example was solved in [7] with a pessimistic viewpoint and the defender's payoff was obtained 3.19 and $C = (0.29, 0.79)$, which is not better than our

Table 12: Game matrix of defender and attacker type 1 in Example 5

	target 1		target 2	
	c	u	c	u
defender	(3,5,6)	(-3,-2,-1)	(9,10,11)	(2,3,5)
attacker	(-2,-1,0)	(2,4,5)	(-2,-1,0)	(9,10,11)

Table 13: Game matrix of defender and attacker type 2 in Example 5

	target 1		target 2	
	c	u	c	u
defender	(0,1,2)	(0,0,0)	(1,2,4)	(-3,-2,-1)
attacker	(-2,-1,0)	(0,1,2)	(0,0,0)	(3,5,6)

solution. Such a result was expected because the solution of [7] is a pessimistic solution.

8 Conclusions

Optimization of force allocation is an important issue in war situations for enemy points of attack, and in any situation (whether war or not), for sensitive centers and infrastructure. A motivated attacker monitors defense forces and takes advantage of the pattern of forces. Defenders must be able to predict the attacker's reaction to different defensive strategies with the highest probability. On the other hand, resource limitation is a major problem in many security areas. Game theory can be used as a valuable tool to analyze these issues and especially to determine the optimal strategy in case of a conflict of interests. Security games are used to solve various security issues according to the type and number of attackers and defenders.

In this paper, a mathematical model was proposed to allocate defense forces in a security game with several attackers. Defenders can use deceptive resources to reduce attack, intensity, productivity, or costs. Applying these resources can fail with certain probabilities. Given these probabilities and budget constraints, a mathematical model was introduced to optimize the allocation of these deceptive resources. In the proposed model, the available

Table 14: Game matrix of defender and attacker type 1 in Example 5

	target 1		target 2	
	c	u	c	u
defender	(1,2,4)	(-2,-1,0)	(2,3,5)	(-3,-2,-1)
attacker	(-3,-2,-1)	(0,1,2)	(-5,-3,-2)	(2,4,5)

budget, the importance of targets for attackers and defenders, and their possible strategies were considered to optimize the allocation of forces. Also, when the defender uses deceptive resources, the ambiguity in the amount of players' payoffs for both players increases. Hence, the players' payoffs were considered as Z-numbers. Then, the problem was solved in a two-stage procedure. In the first stage, the Z-numbers were converted to triangular fuzzy numbers, and in the second stage, the triangular fuzzy numbers were converted to intervals using their expected intervals. Then the interval programming problem was solved by an available method in the literature. Finally, the applicability of the proposed methods was illustrated by some numerical examples.

There are various types of uncertain data, for example, intuitive fuzzy numbers, type-2 fuzzy numbers, and so on. The introduced model handles the payoffs of real, triangular fuzzy numbers, and Z-numbers. However, it cannot be used for other types of fuzzy numbers (or types of uncertainty). As a suggestion, security games with multi-attacker can be solved with other kinds of uncertainty in payoffs.

References

1. Basilico, N., Gatti, N., and Amigoni, F. *Leader-follower strategies for robotic patrolling in environments with arbitrary topologies*, 8th International Conference on Autonomous Agents and Multi-Agent Systems, (2009) 57–64.

2. Bigdeli, H. and Hassanpour, H. *Modeling and solving multiobjective security game problem using multiobjective bilevel problem and its application in metro security system*, Journal of Electronical and Cyber Defence, Special Issue of the International Conference on Combinatorics, Cryptography and Computation (In Persian), (2017) 31–38.

3. Bigdeli, H. and Hassanpour, H. *An approach to solve multi-objective linear production planning games with fuzzy parameters*, Yugosl. J. Oper. Res. 28(2), (2018) 237–248.

4. Bigdeli, H. and Hassanpour, H. *Solving defender-attacker game with multiple decision makers using expected-value Model*, Casp. J. Math. Sci. (CJMS) (2020).

5. Bigdeli, H., Hassanpour, H. and Tayyebi, J. *Optimistic and pessimistic solutions of single and multi-objective matrix games with fuzzy payoffs and analysis of some military cases*, Scientific Journal of Advanced Defense Science and Technology (In Persian), (2017) 133–145.

6. Bigdeli, H., Hassanpour, H. and Tayyebi, J. *Constrained bimatrix games with fuzzy goals and its application in nuclear negotiations*, Iran. J. Numer. Anal. Optim., 8(1), (2018) 81–110.

7. Bigdeli, H., Hassanpour, H. and Tayyebi, J. *Multiobjective security game with fuzzy payoffs*, Iran. J. Fuzzy Syst. 16(1), (2019) 89–101.

8. Brown, G., Carlyle, M., Diehl, D., Kline, J. and Wood, K. *A two-sided optimization for theater ballistic missile defense*, Oper. Res. 53(5), (2005) 745–763.

9. Buckley, J.J. *Joint solution to fuzzy programming problems*, Fuzzy Sets Syst. 72(2), (1995) 215–220.

10. Cohen, F. and Koike, D. *Misleading attackers with deception*, In Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, (2004) 30–37.

11. Conitzer, V. and Sandholm, T. *Computing the optimal strategy to commit to*, 7th ACM conference on Electronic commerce, (2006) 82–90.

12. Daniel, D.C. and Herbig, K.L. *Strategic military deception*, New York: Pergamon Press, 1981.

13. Dickerson, J.P., Simari, G.I., Subrahmanian, V.S. and Kraus, S. *A graph-theoretic approach to protect static and moving targets from adversaries*, 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, (2010) 299–306.

14. Do, C.T., Tran, N.H., Hong, C., Kamhoua, C.A., Kwiat, K.A., Blasch, E., Ren, S., Pissinou, N. and Iyengar, S.S. *Game theory for cyber security and privacy*, ACM Comput. Surv. (CSUR), 50(2), (2017) 1–37.

15. Dubois, D. and Prade, H. *Fuzzy sets and statistical data*, Eur. J. Oper. Res. 25(3), (1986) 345–356.

16. Ehrgott, M. *Multicriteria optimization*, Springer Science & Business Media, 2005.

17. Esmaieli, S., Hassanpour, H. and Bigdeli, H. *Lexicographic programming for solving security game with fuzzy payoffs and computing optimal deception strategy*, Defensive Future Study Researches Journal (In Persian), 5(16), (2020) 89–108.

18. Fang, F., Nguyen, T.H., Pickles, R., Lam, W.Y., Clements, G.R., An, B., Singh, A., Tambe, M. and Lemieux, A. *Deploying PAWS: field optimization of the protection assistant for wildlife security*, In Twenty-Eighth IAAI Conference, (2016).

19. Frank. Jr. and Willard C. *Politico military deception at sea in the Spanish civil war, 1936-39.*, Intell. Natl. Secur. 5(3), (1990) 84–112.

20. Fugate, S. and Ferguson-Walter, K. *Artificial intelligence and game theory models for defending critical networks with cyber deception*, AI Mag. 40(1), (2019) 49–62.

21. Hamilton, D.L. *Deception in Soviet military doctrine and operations*, NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 1986.

22. Heilpern, S. *The expected valued of a fuzzy number*, Fuzzy sets Syst. 47, (1992) 81–86.

23. Kang, B., Wei, D., Li, Y. and Deng, Y. *A method of converting Z-number to classical fuzzy number*, J. Inf. Comput. Sci. 9(3), (2012) 703–709.

24. Karmakar, S., Seikh, M.R. and Castillo, O. *Type-2 intuitionistic fuzzy matrix games based on a new distance measure: Application to biogas-plant implementation problem*, Appl. Soft Comput. 106, (2021) p.107357.

25. Korzhyk, D., Conitzer, V. and Parr, R. *Complexity of computing optimal Stackelberg strategies in security resource allocation games*, 24th AAAI Conference on Artificial Intelligence, (2010) 805–810.

26. Letchford, J. and Vorobeychik, Y. *Computing randomized security strategies in networked domains*, Applied adversarial Reasoning and Risk Modeling, In Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

27. Lye, K. and Wing, J.M. *Game strategies in network security*, Int. J. Inf. Secur. 4(1), (2005) 71–86.

28. McQueen, M.A. and Boyer, W.F. *Deception used for cyber defense of control systems*, 2nd Conference on Human System Interactions, (2009) 624–631.

29. Nishizaki, I. and Sakawa, M. *Stackelberg solutions to multiobjective two-level linear programming problems*, J. Optim. Theory Appl. 103(1), (1999) 161–182.

30. Oikonomakis, P. *Strategic military deception prerequisites of success in technological environment*, 2016.

31. Ren, A., Wang, Y. and Xue, X. *Interactive programming approach for solving the fully fuzzy bilevel linear programming problem*, Knowl Based Syst. 99, (2016) 103–111.

32. Rowe, N.C., Custy, E.J. and Duong, B.T. *Defending cyberspace with fake honeypots*, J. Comput. 2(2), (2007) 25–36.

33. Saati, S.M., Memariani, A. and Jahanshahloo, G.R. *Efficiency analysis and ranking of DMUs with fuzzy data*, Fuzzy Optim. Decis. Mak. 1(3) (2002) 255-267.

34. Sakawa, M. *Fuzzy sets and interactive multiobjective optimization*, Plenumpress, New York and London, 1993.

35. Sakawa, M. and Nishizaki, I. *Cooperative and noncooperative multi-level programming*, Springer, New York and London, 2009.

36. Seikh, M.R., Dutta, S. and Li, D.F. *Solution of matrix games with rough interval pay-offs and its application in the telecom market share problem*, Int. J. Intell. Syst. 36(10), (2021) 6066–6100.

37. Seikh, M.R., Karmakar, S. and Castillo, O. *A novel defuzzification approach of Type-2 fuzzy variable to solving matrix games, An application to plastic ban problem*, Iran. J. Fuzzy Syst. 18(5), (2021) 155–172.

38. Seikh, M.R., Karmakar, S. and Nayak, P.K. *Matrix games with dense fuzzy payoffs*, Int. J. Intell. Syst. 36(4), (2021) 1770–1799.

39. Seikh, M.R., Karmakar, S. and Xia, M. *Solving matrix games with hesitant fuzzy pay-offs*, Iran. J. Fuzzy Syst. 17(4), (2020) 25–40.

40. Sokri, A. *Optimal resource allocation in cyber-security: A game theoretic approach*, Procedia Comput. Sci. 134, (2018) 283–288.

41. Tambe, M. *Security and game theory: algorithms, deployed systems, lessons learned*, Cambridge University Press, 2011.

42. Trejo, K.K., Clempner, J.B. and Poznyak, A.S. *A Stackelberg security game with random strategies based on the extraproximal theoretic approach*, Eng. Appl. Artif. Intell. 37, (2015) 145–153.

43. Trejo, K.K., Kristal K., Clempner, J.B. and Poznyak, A.S. *Adapting strategies to dynamic environments in controllable Stackelberg security games*, IEEE 55th Conference on Decision and Control (CDC), (2016) 5484–5489.

44. Wang, A., Cai, Y., Yang, W. and Hou, Z. *A Stackelberg security game with cooperative jamming over a multiuser OFDMA network*, IEEE Wireless Communications and Networking Conference (WCNC), (2013) 4169–4174.

45. Yin, Y., An, B., Vorobeychik, Y. and Zhuang, J., *Optimal deceptive strategies in security games: A preliminary study*, In Proc. of AAAI, 2013.

46. Zadeh, L.A. *A note on Z-numbers,* Inform. Sci. 181(14) (2011) 2923–2932.

47. Zhu, Q. *Game theory for cyber deception: A tutorial*, 6th Annual Symposium on Hot Topics in the Science of Security, (2019) 1–3.

How to cite this article
S. Esmaeeli, H. Hassanpour and H. Bigdeli . *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 542-566. doi: 10.22067/ijnao.2022.71302.1046.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 567–584
DOI:10.22067/ijnao.2022.70644.1096
https://ijnao.um.ac.ir/

How to cite this article
Research Article

A two-phase method for solving continuous rank-one quadratic knapsack problems

S.E. Monabbati

Abstract

We propose a two-phase algorithm for solving continuous rank-one quadratic knapsack problems (R1QKPs). In particular, we study the solution structure of the problem without the knapsack constraint. In fact an $O(n \log n)$ algorithm is suggested in this case. We then use the solution structure to propose an $O(n^2 \log n)$ algorithm that finds an interval containing the optimal value of the Lagrangian dual of R1QKP. In the second phase, we solve the Lagrangian dual problem using a traditional single-variable optimization method. We perform a computational test on random instances and compare our algorithm with the general solver CPLEX.

AMS subject classifications (2020): 90C20; 90C06; 90C25.

Keywords: Quadratic Knapsack Problem; Line-Sweep Algorithm

1 Introduction

The quadratic knapsack problem (QKP) deals with minimizing a quadratic function over one allocation constraint together with simple bounds on decision variables. Formally, this problem can be written as

$$\text{minimize} \quad \frac{1}{2}\bar{x}^\top Q y - \bar{c}^\top \bar{x}, \tag{1a}$$

* Corresponding author

Received 23 January 2022; revised 29 June 2022; accepted 14 July 2022

Sayyed Ehsan Monabbati
Department of Mathematics, Faculty of Mathematical Sciences, Alzahra University, Tehran, Iran. e-mail: e.monabbati@alzahra.ac.ir

$$\text{subject to} \quad \bar{a}^\top \bar{x} = b, \tag{1b}$$

$$0 \le \bar{x} \le \bar{u}, \tag{1c}$$

where Q is a symmetric $n \times n$ matrix, $\bar{a}, \bar{c}, \bar{u} \in \mathbb{R}^n$, and $b \in \mathbb{R}$. The QKP as a quadratic optimization problem is polynomially solvable when Q is positive definite matrix [12].

When Q is diagonal with strictly positive diagonal entries, then QKP can be viewed as a strictly convex separable optimization problem that has many applications (e.g., resource allocation [1, 13, 14] and multicommodity network flows [9]). The solution methods for solving this type of QKPs usually rely on the fact that the optimal solution to the Lagrangian dual subproblems can be explicitly obtained in terms of the Lagrange multiplier λ of (1b). Therefore, the problem reduces to find a value for λ such that the solution to the corresponding Lagrangian subproblem is satisfied equality constraint (1b).

Helgason, Kennington, and Lall [9] proposed an $O(n \log n)$ algorithm for solving the equation based on searching breakpoints of the Lagrangian dual problem. Brucker [2] found an $O(n)$ bisection algorithm based on the properties of the Lagrangian dual function. Dai and Fletcher [4] proposed a two-phase method. A bracketing phase determines an interval containing the solution followed by the secant phase that approximates the solution within the promising interval. This method is modified by Comminetti, Mascarenhas, and Silva [3] by ignoring the bracketing phase and using a semi-smooth Newton method instead of the secant method. Liu and Liu [11] considered a special case of the strictly convex form of the problem. They found the solution structure of the subproblems and used it in a modified secant algorithm.

Robinson, Jiang, and Lerme [15] used the geometric interpretation of the problem and proposed an algorithm that works in the primal space rather than the dual space. This algorithm iteratively fixes variables and terminates after at most n iterations.

In a more general case, when Q is positive semidefinite in (1), Dussault, Ferland, and Lemaire [7] proposed an iterative algorithm in which a QKP with diagonal Q should be solved in each iteration. Paradalos, Ye, and Han [12] suggested a potential reduction algorithm to solve this class of QKP. di Serafino et al. [6] proposed a two-phase gradient projection with acceptable numerical performance compared to similar gradient-based methods.

QKPs with positive definite Q are also solved by a gradient projection method [4] and an augmented-Lagrangian approach [8].

In this paper, we suppose that Q is a rank-one symmetric matrix, that is, $Q = qq^\top$ for some $q \in \mathbb{R}^n$. Moreover, we assume that $0 < \bar{u}$. Without loss of generality, we assume that $q_i \neq 0$ for each i. By the changing variables

$$x_i \leftarrow q_i \bar{x}_i, \quad c_i \leftarrow \frac{\bar{c}_i}{q_i}, \quad a_i \leftarrow \frac{\bar{a}_i}{q_i}, \quad u_i \leftarrow \max\{0, q_i \bar{u}_i\},$$

problem (1) is reduced to

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}(\mathbf{1}^\top x)^2 - c^\top x, && \text{(2a)} \\ \text{subject to} \quad & a^\top x = b, && \text{(2b)} \\ & 0 \le x \le u. && \text{(2c)} \end{aligned}$$

Sharkey and Romeijn [16] studied a class of nonseparable nonlinear knapsack problems in which one has to

$$\begin{aligned} \text{minimize} \quad & g(s^\top x) - c^\top x, \\ \text{subject to} \quad & a^\top x = b, \\ & l \le x \le u, \end{aligned} \qquad (3)$$

where $g : \mathbb{R} \to \mathbb{R}$ is an arbitrary real-valued function, and $s \in \mathbb{R}^n$ is given. They introduced an algorithm for solving (3) that runs in $O(n^2 \max\{\log n, \phi\})$, where ϕ is the time required to solve a single-variable optimization problem $\min\{g(S) - \alpha S \ : \ L \le S \le U\}$ for given $\alpha, L, U \in \mathbb{R}$. With $g(t) = t^2$ and s equal to the all-one vector, problem (2) is a special case of problem (3). That is, there exists an $O(n^2 \max\{\log n, 1\}) = O(n^2 \log n)$ algorithm for solving problem (2).

In this paper, we consider a two-phase algorithm for solving problem (2). In Section 2, we study the solution structure of the relaxed version of the problem in which the equality constraint (2b) is excluded. We show that the relaxed version could be solved in $O(n \log n)$ time. In Section 3, in phase I, we use the solution structure of the relaxed version to find an interval that may contain the optimal value of the Lagrangian dual function. This is done in $O(n^2 \log n)$ time in the worst case. Then, we perform phase II, in which we explore the interval by a single-variable optimization method to find the optimal Lagrangian multiplier with the desired precision. In Section 4, we perform a computational test. In particular, we compare the algorithm with the general quadratic programming solver CPLEX.

2 Solution structure of the relaxed version

In this section, we consider the following relaxed version of the problem (2):

$$\begin{aligned} \text{minimize} \quad & f(x) = \frac{1}{2}(\mathbf{1}^\top x)^2 - c^\top x, && \text{(4a)} \\ \text{subject to} \quad & 0 \le x \le u. && \text{(4b)} \end{aligned}$$

We propose a characterization of the solution in the primal space. Note that most of algorithms for such problems use the so-called KKT conditions to study the solution structure.

Without loss of generality, we assume that $c_1 \geq c_2 \geq \cdots \geq c_n \geq 0$, and that $l_i = 0$, $i = 1, \ldots, n$. Given two vectors $a, b \in \mathbb{R}^n$, we denote the set $\{x \ : \ a \leq x \leq b\}$ by $[a, b]$. Finally, given a vector $u \in \mathbb{R}^n$, we define $U_k := \sum_{i=1}^{k} u_i$ for $k = 1, \ldots, n$, and $U_0 := 0$.

Now consider the following preliminary lemmas:

Lemma 1. For $k = 1, \ldots, n$ define $x^{(k)}$ as

$$x_i^{(k)} = \begin{cases} u_i, & i = 1, \ldots, k, \\ 0, & i = k+1, \ldots, n, \end{cases}$$

and $x^{(0)}$ as the all-zero vector, and define G_k as

$$G_k = \frac{1}{2}\left(U_k + U_{k-1}\right) - c_k = U_{k-1} + \frac{1}{2}u_k - c_k.$$

Then the following assertions hold:

(i) If $\bar{n}$ is the smallest index in $\{1, \ldots, n\}$ such that $G_{\bar{n}} \geq 0$, then $\min_{i=1,\ldots,n} f(x^{(i)}) = f(x^{(\bar{n}-1)})$.

(ii) If $G_k < 0$ for all $k = 1, \ldots, n-1$, then $\min_{i=1,\ldots,n} f(x^{(i)}) = f(x^{(n)})$.

Proof. (i) For $1 \leq k \leq n-1$, we have

$$\begin{aligned} G_k - G_{k+1} &= \frac{1}{2}\left(U_k + U_{k-1}\right) - c_k - \frac{1}{2}\left(U_{k+1} + U_k\right) + c_{k+1} \\ &= -\frac{1}{2}\left(u_k + u_{k+1}\right) + \left(c_{k+1} - c_k\right) \\ &< 0. \end{aligned}$$

Thus

$$G_1 < G_2 < \cdots < G_{\bar{n}-1} < 0 \leq G_{\bar{n}} < G_{\bar{n}+1} < \cdots < G_n.$$

On the other hand, for $1 \leq k \leq n-1$, we have

$$\begin{aligned} f(x^{(k+1)}) - f(x^{(k)}) &= \frac{1}{2}U_{k+1}^2 - \sum_{i=1}^{k+1} u_i c_i - \frac{1}{2}U_k^2 + \sum_{i=1}^{k} u_i c_i \\ &= \frac{1}{2}U_{k+1}^2 - u_{k+1}c_{k+1} - \frac{1}{2}U_k^2 \\ &= \frac{1}{2}(U_{k+1}^2 - U_k^2) - u_{k+1}c_{k+1} \\ &= \frac{1}{2}(U_{k+1} - U_k)(U_{k+1} + U_k) - u_{k+1}c_{k+1} \\ &= u_{k+1}\left(\frac{1}{2}(U_{k+1} + U_k) - c_{k+1}\right) \\ &= u_{k+1}G_{k+1}. \end{aligned} \tag{5}$$

Now let $m > \bar{n} - 1$. Then

$$\begin{aligned} f(x^{(m)}) - f(x^{(\bar{n}-1)}) &= f(x^{(m)}) - f(x^{(m-1)}) + f(x^{(m-1)}) + \cdots + f(x^{(\bar{n})}) - f(x^{(\bar{n}-1)}) \\ &= u_m G_m + \cdots + u_{\bar{n}} G_{\bar{n}} > G_{\bar{n}}(u_m + \cdots + u_{\bar{n}+1}) \\ &\geq 0. \end{aligned}$$

Similarly, if $m < \bar{n} - 1$, then $f(x^{(m)}) - f(x^{(\bar{n}-1)}) \geq 0$.

(ii) The second part can be easily proved by considering (5). □

We need the following result for two-dimensional version of problem (4).

Lemma 2. Consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & f(x_1, x_2) = \frac{1}{2}(x_1 + x_2)^2 - c_1 x_1 - c_2 x_2, \\ \text{subject to} \quad & 0 \leq x_1 \leq u_1, \\ & 0 \leq x_2 \leq u_2, \end{aligned} \tag{6}$$

where $c_1 \geq c_2 \geq 0$ and u_1 and u_2 are real positive constants. Define set $I := I_1 \cup I_2$, where $I_1 = \{(u_1, x_2) \ : \ 0 \leq x_2 \leq u_2\}$, and $I_2 = \{(x_1, 0) \ : \ 0 \leq x_1 \leq u_1\}$. Then, problem (6) has no optimal solution outside of I.

Proof. If $c_1 = c_2$, then $f(x_1, x_2) = \frac{1}{2}(x_1 + x_2)^2 - c_1(x_1 + x_2) = \frac{1}{2}z^2 - c_1 z = g(z)$, where $z = x_1 + x_2$. It is easy to see that $x^* = (x_1^*, x_2^*)$ with $x_1^* = \min\{c_1, u_1\}$ and $x_2^* = \min\{c_1 - x_1^*, u_2\}$, is the optimal solution to the problem, and we have $x^* \in I$. Assume that $c_1 \neq c_2$. The feasible region of (6) is equal to $I_1 \cup I_2 \cup I_3 \cup I_4$, where $I_3 = \{(x_1, x_2) \ : \ 0 < x_1 < u_1, 0 < x_2 < u_2\}$ and $I_4 = \{(0, x_2) \ : \ 0 < x_2 < u_2\} \cup \{(x_1, u_2) \ : \ 0 < x_1 < u_1\}$. We show that there is no optimal solution in I_3 and I_4. Indeed, we write the KKT optimality conditions as follows:

$$x_1 + x_2 - c_1 + \alpha_1 - \alpha_2 = 0, \tag{7}$$

$$x_1 + x_2 - c_2 + \beta_1 - \beta_2 = 0, \tag{8}$$

$$\alpha_1(x_1 - u_1) = 0, \quad \alpha_2 x_1 = 0, \tag{9}$$

$$\beta_1(x_2 - u_2) = 0, \quad \beta_2 x_2 = 0, \tag{10}$$

$$0 \leq x_1 \leq u_1, \tag{11}$$

$$0 \leq x_2 \leq u_2, \tag{12}$$

$$\alpha_1, \alpha_2, \beta_1, \beta_2 \geq 0, \tag{13}$$

where α_i and β_i, $i = 1, 2$ are KKT multipliers corresponding to the bound constraints. If $(x_1, x_2) \in I_3$, then from (9) and (10), we have $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 0$. Substituting these values in (7) and (8) implies that $c_1 = c_2$, which contradicts our assumption. On the other hand, if $(x_1, x_2) \in I_4$ and $x_1 = 0$, then $\alpha_1 = 0$. Now, (7) implies that $x_2 = c_1 + \alpha_2 > 0$. Thus $\beta_2 = 0$. From (8), we have $x_2 = c_2 - \beta_1$. Therefore, $c_2 = c_1 + \alpha_2 + \beta_1 \geq c_1$. This contradicts

our assumption on c_i's. That is, problem (6) has no optimal solution with $x_1 = 0$. Now, if $x_2 = u_2$, then $\alpha_1 = 0$ and β_2. It implies from (7) and (8) that $0 \leq \alpha_2 + \beta_1 = c_2 - c_1 \leq 0$. That is, $c_2 = c_1$, a contradiction. □

Theorem 1. Suppose that $x^{(k)}$ and G_k, $k = 1, \ldots, n$, and $\bar{n}$ are defined as in Lemma 1. Then the following assertions hold:

(i) For $\bar{n} > 1$, define δ_1 and δ_2 as $\delta_1 := \min\{c_{\bar{n}-1} - U_{\bar{n}-2}, u_{\bar{n}-1}\}$ and $\delta_2 := \max\{c_{\bar{n}} - U_{\bar{n}-1}, 0\}$. Also, define $\bar{x}$ and $\tilde{x}$ as

$$\bar{x} = x^{(\bar{n}-2)} + \delta_1 e_{\bar{n}-1}, \quad \tilde{x} = x^{(\bar{n}-1)} + \delta_2 e_{\bar{n}},$$

where e_i is the ith column of the identity matrix of dimension n. Then $\min\{f(\bar{x}), f(\tilde{x})\}$ is the optimal value of the following optimization problem:

$$\begin{aligned} &\text{minimize} && f(x), \\ &\text{subject to} && x^{(\bar{n}-2)} \leq x \leq x^{(\bar{n})}. \end{aligned} \tag{14}$$

(ii) For $\bar{n} = 1$, define $\delta := \min\{c_1, u_1\}$ and $\tilde{x} := \delta e_1$. Then $f(\tilde{x})$ is the optimal value of the following optimization problem:

$$\begin{aligned} &\text{minimize} && f(x), \\ &\text{subject to} && x^{(0)} \leq x \leq x^{(1)}. \end{aligned}$$

(iii) For $G_k < 0$ for all $k = 1, \ldots, n$, define $\delta := \min\{c_n - U_{n-1}, u_n\}$ and $\tilde{x} := x^{(n-1)} + \delta e_n$. Then $f(\tilde{x})$ is the optimal value of the following optimization problem:

$$\begin{aligned} &\text{minimize} && f(x), \\ &\text{subject to} && x^{(n-1)} \leq x \leq x^{(n)}. \end{aligned}$$

Proof. (i) By Lemma 2, we can partition the optimal solution set as $I_1 \cup I_2$, where $I_1 = [x^{(\bar{n}-2)}, x^{(\bar{n}-1)}]$ and $I_2 = [x^{(\bar{n}-1)}, x^{(\bar{n})}]$. We show that $f(\bar{x}) = \min\{f(x) : x \in I_1\}$ and $f(\tilde{x}) = \min\{f(x) : x \in I_2\}$. Indeed, we use a simple technique of single-variable calculus. Let $x \in I_1$. Then $x = x(\delta)$, for some $\delta \in [0, u_{\bar{n}-1}]$, where $x(\delta) = x^{(\bar{n}-2)} + \delta e_{\bar{n}-1}$. Thus the problem $\min\{f(x) : x \in I_1\}$ reduces to $\min\{f(x(\delta)) : 0 \leq \delta \leq u_{\bar{n}-1}\}$. On the other hand, one can write

$$f(x(\delta)) = \frac{1}{2}\left(U_{\bar{n}-2} + \delta\right)^2 - \sum_{i=1}^{\bar{n}-2} c_i u_i - c_{\bar{n}-1}\delta.$$

We have $df(x(\delta))/d\delta = U_{\bar{n}-2} + \delta - c_{\bar{n}-1}$. Thus $df(x(\delta))/d\delta = 0$ only if $\delta = \delta' = c_{\bar{n}-1} - U_{\bar{n}-2}$. Since $d^2 f(x(\delta))/d\delta^2 > 0$ and $\delta' > \frac{1}{2}u_{\bar{n}-1}$ the optimal value is achieved at δ_1.

To prove $f(\tilde{x}) = \min\{f(x) : x \in I_2\}$, by the same argument as the previous paragraph, it suffices to solve single optimization problem $\min\{f(x(\delta)) : 0 \leq \delta \leq u_{\bar{n}}\}$, where $x(\delta) = x^{(\bar{n}-1)} + \delta e_{\bar{n}}$. It is easy to see that if $\delta = \delta' = c_{\bar{n}} - U_{\bar{n}-1}$, then $df(x(\delta))/d\delta = 0$. Since $\delta' \leq \frac{1}{2}u_{\bar{n}}$, by the definition of $\bar{n}$, then $f(\tilde{x})$ is the optimal value of $\min\{f(x) : x \in I_2\}$.

The proof of parts (ii) and (iii) is similar. □

The following Corollary 1 presents simple conditions under which the optimal solution to the problem in Theorem 1(i) is $\bar{x}$ or $\tilde{x}$.

Corollary 1. In Theorem 1(i),

(i) if $\delta_1 = u_{\bar{n}-1}$, then $\min\{f(\bar{x}), f(\tilde{x})\} = f(\tilde{x})$, and

(ii) if $\delta_2 = 0$, then $\min\{f(\bar{x}), f(\tilde{x})\} = f(\bar{x})$.

Proof. For brevity, we just prove part (i). The proof of the second part is similar. Under the assumption of part (i), we have

$$f(\bar{x}) - f(\tilde{x}) = \frac{1}{2}U_{\bar{n}-1}^2 - \sum_{i=1}^{\bar{n}-1} c_i u_i - \frac{1}{2}c_{\bar{n}}^2 + \sum_{i=1}^{\bar{n}-1} c_i u_i + c_{\bar{n}}(c_{\bar{n}} - U_{\bar{n}-1}) = \frac{1}{2}\left(U_{\bar{n}-1} - c_{\bar{n}}\right)^2 \geq 0.$$

□

Theorem 1 solves a relaxed version of problem (2). In Theorem 2, we show that the solution to the relaxed version is the solution to the original problem.

Theorem 2. Define G_k's, $x^{(k)}$'s, $\bar{n}$, $\bar{x}$, and $\tilde{x}$ as in Theorem 1. Then, the following assertions hold:

(i) If $1 < \bar{n} \leq n$, then $\min\{f(\bar{x}), f(\tilde{x})\}$ is the optimal value of (4), where $\tilde{x}$ and $\bar{x}$ are defined as in Theorem 1(i).

(ii) If $\bar{n} = 1$, then $f(\tilde{x})$ is the optimal value of (4), where $\tilde{x}$ is defined as in Theorem 1(ii).

(iii) If $G_k < 0$ for all $k = 1, \ldots, n$, then $f(\tilde{x})$ is the optimal solution to (4), where $\tilde{x} = x^{(n-1)} + \delta' e_n$ and $\delta' = \min\{c_n - U_{n-1}, u_n\}$.

Proof. For two vectors $x, z \in \mathbb{R}^n$, we have

$$f(z) - f(x) = \frac{1}{2}(\mathbf{1}^\top z + \mathbf{1}^\top x)(\mathbf{1}^\top z - \mathbf{1}^\top x) - c^\top(z - x). \tag{15}$$

Let x be a feasible solution to (4). If $x = u = x^{(n)}$, then from the definition of $\bar{n}$, we have $f(x^{\bar{n}-1}) \leq f(x)$, and the result follows from Theorem 1. Suppose $x \neq u$. We show there exists a specially structured feasible solution x' that is better than x. Indeed, let k be such that

$$U_k \le \mathbf{1}^\top x < U_{k+1}.$$

Define vector x' by

$$x'_i = \begin{cases} u_i, & i = 1, \dots, k, \\ \mathbf{1}^\top x - U_k, & i = k+1, \\ 0, & i = k+2, \dots, n. \end{cases}$$

Then, clearly x' is feasible for (4) and

$$\mathbf{1}^\top x' = \sum_{i=1}^{k} x'_i + x'_{k+1} + \sum_{i=k+2}^{n} x'_i = \sum_{i=1}^{k} u_i + \sum_{i=1}^{n} x_i - \sum_{i=1}^{k} u_i = \mathbf{1}^\top x.$$

Moreover, we obtain

$$\begin{aligned} c^\top x' &= \sum_{i=1}^{k} u_i c_i + c_{k+1} x'_{k+1} = \sum_{i=1}^{k} u_i c_i + c_{k+1} \sum_{i=1}^{n} x_i - c_{k+1} \sum_{i=1}^{k} u_i \\ &= \sum_{i=1}^{k} u_i c_i + c_{k+1} \sum_{i=1}^{k} x_i + c_{k+1} \sum_{i=k+1}^{n} x_i - c_{k+1} \sum_{i=1}^{k} u_i \\ &\ge \sum_{i=1}^{k} u_i c_i + \sum_{i=1}^{k} (x_i - u_i) c_i + \sum_{i=k+1}^{n} x_i c_i \quad \text{(by the monotonicity of } c_i\text{'s)} \\ &= c^\top x. \end{aligned}$$

Therefore, (15) implies that $f(x') - f(x) = -c^\top (x' - x) \le 0$, that is, $f(x') \le f(x)$.

(i) Now, we consider three cases for the index k introduced in the definition of x': $k \ge \bar{n}$, $k < \bar{n} - 2$, and $k = \bar{n} - 1, \bar{n} - 2$. In the latter case, we have $x^{(\bar{n}-2)} \le x' \le x^{(\bar{n})}$, so the assertion is true by Theorem 1, since

$$\min\{f(\bar{x}), f(\tilde{x})\} = \min\{f(x) \ : \ x \in [x^{(\bar{n}-2)}, x^{(\bar{n})}]\} \le f(x') \le f(x).$$

We show in both the other cases, there is a point in the set $\{x^{(i)}\}_{i=1,\dots,n}$ better than x', that is, $f(x^{(i)}) \le f(x')$ for some $i = 1, \dots, n$. By Lemma 1, $f(x^{(\bar{n}-1)}) \le f(x^{(i)})$ and the result follows by Theorem 1.

First, let $k \ge \bar{n}$. Then

$$\begin{aligned} f(x^{(k)}) - f(x') &= \frac{1}{2}(\mathbf{1}^\top x^{(k)} - \mathbf{1}^\top x')(\mathbf{1}^\top x^{(k)} + \mathbf{1}^\top x') - c^\top (x^{(k)} - x') \\ &= -\frac{1}{2} x'_{k+1}(2U_k + x'_{k+1}) + c_{k+1} x'_{k+1} \\ &= -x'_{k+1} \left(\frac{1}{2}(2U_k + x'_{k+1}) - c_{k+1} \right). \end{aligned}$$

On the other hand, we have

$$2U_k + x'_{k+1} = 2\sum_{i=1}^{\bar{n}-1} u_i + \sum_{i=\bar{n}}^{k} u_i + x'_{k+1} \geq U_{\bar{n}} + U_{\bar{n}-1}.$$

Therefore,

$$\frac{1}{2}(2U_k + x'_{k+1}) - c_{k+1} \geq \frac{1}{2}(U_{\bar{n}} + U_{\bar{n}-1}) - c_{\bar{n}} = G_{\bar{n}} \geq 0.$$

Thus $f(x^{(k)}) \leq f(x')$.

Now, let $k < \bar{n} - 2$. Then

$$\begin{aligned} f(x^{(k+1)}) - f(x') &= \frac{1}{2}(\mathbf{1}^\top x^{(k+1)} - \mathbf{1}^\top x')(\mathbf{1}^\top x^{(k+1)} + \mathbf{1}^\top x') - c^\top (x^{(k+1)} - x') \\ &= \frac{1}{2}(u_{k+1} - x'_{k+1})(U_{k+1} + U_k + x'_{k+1}) - c_{k+1}(u_{k+1} - x'_{k+1}) \\ &= (u_{k+1} - x'_{k+1})\left(\frac{1}{2}(2U_k + x'_{k+1} + u_{k+1}) - c_{k+1}\right). \end{aligned}$$

On the other hand, we have

$$2U_k + x'_{k+1} + u_{k+1} \leq 2U_k + 2u_{k+1} \leq 2U_k + 2\sum_{i=k+1}^{\bar{n}-2} u_i + u_{\bar{n}-1} = U_{\bar{n}-2} + U_{\bar{n}-1}.$$

Hence,

$$\frac{1}{2}(2U_k + x'_{k+1} + u_{k+1}) - c_{k+1} \leq \frac{1}{2}(U_{\bar{n}-2} + U_{\bar{n}-1}) - c_{\bar{n}-1} = G_{\bar{n}-1} < 0.$$

That is, $f(x^{(k+1)}) < f(x')$. Thus in both cases, there exist a point, say $x^{(t)}$, such that $f(x^{(t)}) \leq f(x') \leq f(x)$. Now, by Lemma 1, $f(x^{(\bar{n}-1)}) \leq f(x^{(t)}) \leq f(x)$ and the result follows by Theorem 1.

Proof of (ii). Consider the possible values of k at the beginning of the proof of part (i). Here, we just have $k \geq \bar{n} = 1$. Now, similar argument for this case proves (ii).

Proof of (iii). Again consider the possible values of k at the beginning of the proof of part (i). Similar argument with case $k < \bar{n} - 2$ for $\bar{n} = n + 1$ proves part (iii).

□

We conclude the following result on the time needed to solve problem (4).

Theorem 3. There exists an $O(n \log n)$ time algorithm for problem (4).

Proof. When the index $\bar{n}$ is determined, the solution can be determined in $O(n)$ time. We need $O(n \log n)$ to sort the vector c, $O(n)$ to compute vector

G, and $O(\log n)$ to find the index $\bar{n}$. That is, problem (4) can be solved in $O(n \log n)$. □

3 The algorithm

In this section, we propose our algorithm for solving problem (2). The algorithm consists of two phases: bounding the optimal Lagrangian multiplier and computing the optimal solution to the desired precision. The bounding phase is based on the Lagrangian dual of (2) and the solution structure of the relaxed version has been described in section 2.

3.1 Lagrangian dual

Let λ be the Lagrange multiplier of equality constraint in (2). Then, the Lagrangian function is given by

$$\begin{aligned} \phi(\lambda) &:= \min \left\{ \frac{1}{2}(\mathbf{1}^\top x)^2 - c^\top x + \lambda(b - a^\top x) \ : \ 0 \le x \le u \right\} \\ &= \lambda b + \min \left\{ \frac{1}{2}(\mathbf{1}^\top x)^2 - (c + \lambda a)^\top x \ : \ 0 \le x \le u \right\}. \end{aligned} \tag{16}$$

We have the following statement about the structure of the Lagrangian function ϕ.

Theorem 4. For a given Lagrange multiplier λ, define $\bar{n}$ as in Theorem 2. If $\bar{n} > 1$, then

$$\phi(\lambda) = \lambda b + f_\lambda(x^{(\bar{n}-1)}), \text{if } c_{\bar{n}}(\lambda) \le U_{\bar{n}-1} \le c_{\bar{n}-1}(\lambda), \qquad \text{(Type A)}$$
$$\phi(\lambda) = \lambda b + p_{\bar{n}}(\lambda), \text{if } U_{\bar{n}-1} < c_{\bar{n}}(\lambda), \qquad \text{(Type B)}$$
$$\phi(\lambda) = \lambda b + p_{\bar{n}-1}(\lambda), \text{if } U_{\bar{n}-1} > c_{\bar{n}-1}(\lambda), \qquad \text{(Type C)}$$

where f_λ is the objective function of the optimization part of (16), and

$$p_k(\lambda) = -\frac{1}{2}a_k^2\lambda^2 - a^\top d_k\lambda + \frac{1}{2}c_k^2 - c^\top d_k,$$
$$d_k = x^{(k-1)} + (c_k - U_{k-1})e_k.$$

Proof. The proof is based on the four possible cases for δ_1 and δ_2 in Theorem 2. We just prove (Type A) and, for the sake of brevity, we omit the remaining parts.

Suppose that $c_{\bar{n}}(\lambda) \le U_{\bar{n}-1} \le c_{\bar{n}-1}(\lambda)$. Then we have $c_{\bar{n}-1}(\lambda) - U_{\bar{n}-2} \ge u_{\bar{n}-1}$ and $c_{\bar{n}}(\lambda) - U_{\bar{n}-1} \le 0$. Therefore, the values of δ_1 and δ_2 in Theorem 2 can be determined as

$$\delta_1 = \min\{c_{\bar{n}-1}(\lambda) - U_{\bar{n}-2}, u_{\bar{n}-1}\} = u_{\bar{n}-1},$$
$$\delta_2 = \max\{c_{\bar{n}}(\lambda) - U_{\bar{n}-1}, 0\} = c_{\bar{n}}(\lambda) - U_{\bar{n}-1}.$$

Thus we have $\bar{x} = x^{(\bar{n}-2)} + \delta_1 e_{\bar{n}-1} = x^{(\bar{n}-1)}$. By some simplifications, we have $f_\lambda(\tilde{x}) - f_\lambda(\bar{x}) = \frac{1}{2}(c_{\bar{n}}(\lambda) - U_{\bar{n}-1})^2 \geq 0$. Now, Theorem 2 implies that $\min\{f_\lambda(x) \ : \ 0 \leq x \leq u\} = \min\{f(\tilde{x}), f(\bar{x})\} = f_\lambda(\bar{x}) = f_\lambda(x^{(\bar{n}-1)})$. □

Now, one may conclude that if $\bar{n}$ is fixed on an interval $[\lambda_a, \lambda_b]$, then $\phi(\lambda)$ is a piecewise function that contains exactly three pieces. However, the following simple example shows that this is not true.

Example 1. Consider problem (2) with the following parameters:

$$a^\top = \begin{bmatrix} -7 & -5 & 7 & -5 & 7 \end{bmatrix}, \quad c^\top = \begin{bmatrix} 54 & 44 & 15 & -8 & -70 \end{bmatrix},$$
$$u^\top = \begin{bmatrix} 62 & 48 & 36 & 84 & 59 \end{bmatrix}.$$

In Figure 1, we plot $\phi(\lambda)$ for $\lambda \in [-8.36, 7.00]$. We distinct three cases in (Type A), (Type B), and (Type C) in blue, red, and green, respectively. As it can be seen in Figure 1, $\phi(\lambda)$ consists of four pieces.

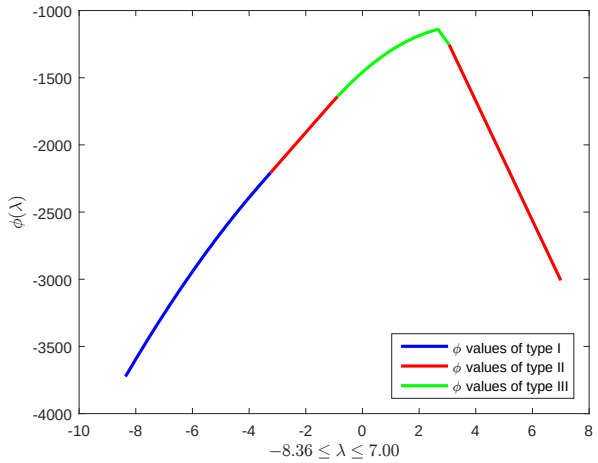


Figure 1: Plot of the Lagrangian function $\phi(\lambda)$ for Example 1.

The inner optimization problem in (16) is a special case of problem (4) that can be solved by Theorem 2. In Theorem 2, it is assumed that coefficients of the linear term in the objective function are sorted in decreasing order. In problem (16), the order of coefficients of the linear term depends on λ. From now on, we denote by $c_i(\lambda)$ the coefficient of x_i, that is, $c_i(\lambda) = c_i + \lambda a_i$. Moreover, we denote the line $\{c_i(\lambda) \ : \ \lambda \in \mathbb{R}\}$ by ℓ_i. It is easy to see that

when λ becomes greater than the intersection of ℓ_i and ℓ_j, $c_i(\lambda)$ and $c_j(\lambda)$ change position in the ordered list of coefficients.

We use a modification of the well-known plane sweep algorithm to find the ordered intersection points of lines $\{\ell_i \ : \ i = 1, \ldots, n\}$. Now, let λ_a, λ', and λ_b be three consecutive intersection points. Then, because the Lagrangian function is unimodal, the optimal Lagrange multiplier λ^* lies in the interval $[\lambda_a, \lambda_b]$ if $\phi(\lambda') > \phi(\lambda_a)$ and $\phi(\lambda') > \phi(\lambda_b)$.

We modify the implementation of the line-sweep algorithm proposed in [5]. In this algorithm, a vertical line ℓ sweeps the plane from left to right. The *status* of the sweep line is the ordered sequence of lines that intersect it. The status initially contains all lines in the order of decreasing slope, that is, the order of lines when they intersect with the sweep line at $\lambda = -\infty$. The status is updated when ℓ reaches an intersecting point. For example, suppose that the sequence of four lines ℓ_l, ℓ_i, ℓ_j, and ℓ_m appears in the status when ℓ reaches the intersection point of ℓ_i and ℓ_j. Then, ℓ_i and ℓ_j switch the position and intersection of lines ℓ_i and ℓ_m, and the intersection of ℓ_j and ℓ_l are to be checked. The new detected intersection points are stored to proceed. The order of cost coefficient of the linear term in $\phi(\lambda)$ is unchanged between two consecutive intersection points.

If $c_i(\lambda) < 0$ for some i, then $x_i = 0$ in the optimal solution to the $\phi(\lambda)$ subproblem. We introduce a set Z to store the non-vanished variables. To do so, we add a dummy line $\ell_0 : c_0(\lambda) = 0$. In each intersection of the dummy line and the other lines, the set Z should be updated. In fact, if ℓ_i intersect ℓ_0 and $i \notin Z$, then we add i to Z; otherwise, if $i \in Z$, then it should be removed from Z. In other words, since we sweep the plane from left to right, if ℓ_i intersect ℓ_0 and $a_i < 0$, then we add i to Z. If ℓ_i intersect ℓ_0 and $a_i > 0$, then it means that i should be removed from Z. Moreover, Z initially contains the set of all lines with a positive slope. With this modification, we guarantee that between two consecutive intersection points, the set of zero-valued variables is unchanged. It should be noted here that lines with equal slopes are sorted based on increasing order of c_i's. We summarize the approach in Algorithm 1. This algorithm is used as the first phase in the main algorithm.

Theorem 5. Algorithm 1 runs in $O(n^2 \log n)$ time.

Proof. Initializing state array ℓ, line indices array p and the queue Q in steps 3–7 needs $O(n \log n)$ time. In each iteration, we perform two main operations: computing the value of ϕ for a new intersect point λ^{new} and updating Q. The order of $c_i(\lambda^{\text{new}})$ and the vector G can be updated from the previous intersection point in $O(1)$ time. Finding $\bar{n}$ needs $O(\log n)$, using binary search. On the other hand, insertion and deletion on the priority queue Q takes $O(\log n)$ since one can implement the priority queue by a heap to store the intersection points. Therefore, each iteration of the main loop needs $O(\log n)$ time. Since there are $O(n^2)$ intersection points, the algorithm runs in $O(n^2 \log n)$. □

Algorithm 1 A plane sweep algorithm for finding an interval containing the optimal solution to the Lagrangian dual problem.

Input: vectors c, a, and u and scaler b.
Output: interval $[\lambda_a, \lambda_b]$ that contains the optimal solution to the problem $\max_{\lambda\in\mathbb{R}} \phi(\lambda)$ or the smallest and largest intersection points.
Initialize a state array, $\ell = [1, \ldots, n]$, with lines $\ell[1], \ldots, \ell[n]$ sorted in decreasing order of their slope.
Initialize queue $Q = \emptyset$.
Initialize line indices array $p = [1, \ldots, n]$.
FAIL $\leftarrow$ `true`
Insert intersection points of all adjacent lines into Q.
Set $\lambda^{\text{prev}} \leftarrow -\infty$, $\lambda^{\text{prev prev}} \leftarrow -\infty$
while Q is not empty **do**
    Pop from Q the current intersection point λ^{new} and the corresponding two adjacent lines $\ell[i]$ and $\ell[j]$.
    Update state array: $\ell[p[i]] \leftrightarrow \ell[p[j]]$.
    Update the line indices array: $p[i] \leftrightarrow p[j]$.
    Insert the intersection point of $\ell[p[i]]$ and $\ell[p[i]+1]$ and the intersection point of $\ell[p[j]]$ and $\ell[p[j]-1]$ into Q, if there exists any.
    if $\phi(\lambda^{\text{prev}}) > \phi(\lambda^{\text{prev prev}})$ and $\phi(\lambda^{\text{prev}}) > \phi(\lambda^{\text{new}})$ **then**
        Set FAIL $\leftarrow$ `false`
        return $[\lambda^{\text{prev prev}}, \lambda^{\text{new}}]$ as the promising interval.
    end if
    Set $\lambda^{\text{prev prev}} \leftarrow \lambda^{\text{prev}}$.
    Set $\lambda^{\text{prev}} \leftarrow \lambda^{\text{new}}$.
end while
if FAIL **then**
    return the smallest λ_{LB} and the largest λ_{UB} intersection points.
end if

Let λ_{LB} and λ_{UB} be the smallest and greatest intersection points of lines $\{\ell_i \ : \ i = 1, \ldots, n\}$, respectively. The optimal solution to the Lagrangian problem may lie out of the interval $[\lambda_{LB}, \lambda_{UB}]$. In this case, Algorithm 1 fails to find the optimal interval. So, we explore the outside of $[\lambda_{LB}, \lambda_{UB}]$ in a separate phase.

First, consider the exploration of $(-\infty, \lambda_{LB})$. Since the components of vector G in Theorem 2 are linear functions in term of λ, then there exists $\lambda'_{LB} < \lambda_{LB}$ such that the order of G_k's does not change for $\lambda < \lambda'_{LB}$. Indeed, a similar procedure for finding the smallest intersection of lines ℓ_i's can be used here to compute λ'_{LB}. Now, since $\phi(\lambda)$ is unimodal, one can conclude that

$$\max_{(-\infty, \lambda_{LB}]} \phi(\lambda) = \max_{[\lambda'_{LB}, \lambda_{LB}]} \phi(\lambda). \tag{17}$$

Similarly, for the values of $\lambda > \lambda_{UB}$, one can find a threshold, say λ'_{UB}, such that

$$\max_{[\lambda_{UB}, \infty)} \phi(\lambda) = \max_{[\lambda_{UB}, \lambda'_{UB}]} \phi(\lambda). \tag{18}$$

We summarize the main algorithm in Algorithm 2.

Algorithm 2 A two-phase algorithm for solving rank-one QKP (2).

1: Run Algorithm 1 to find a promising interval that contains the optimal Lagrange multiplier.
2: **if** Algorithm 1 returns an interval $[\lambda_a, \lambda_b]$ **then**
3: Solve the optimization problem $\max_{[\lambda_a, \lambda_b]} \phi(\lambda)$ and return the optimal solution.
4: **else**
5: Solve optimization problems (17) and (18) and store the optimal values.
6: **end if**
7: **return** the best λ found as an optimal Lagrange multiplier.

It is clear that Algorithm 2 converges to the optimal solution since the output interval of Algorithm 1 contains the optimal solution and $\phi(\lambda)$ is unimodal. In fact, the single variable optimization problem in step 3 can be solved efficiently by a classical root-finding algorithm.

4 Computational experiments

In this section, we compare the running time of Algorithm 2 with the general convex quadratic programming solver, CPLEX. We implement Algorithm 2 with MATLAB R2019b. All runs are performed on a system with a Core i7 2.00 GHz CPU and 8.00 GB of RAM equipped with a 64bit Windows operating system. We solve single variables optimization problems (17), (18),

Table 1: Parameters for two types of problem instances.

Type	a	c	l	$u-l$
TypeI	$U(-50,50)$	$U(-50,50)$	$U(0,20)$	$U(1,100)$
TypeII	$U(-100,10)$	$U(10,100)$	$U(0,20)$	$U(1,100)$

and step 3 in Algorithm 2, using MATLAB built-in function fminbnd, which is based on the golden section search and parabolic interpolation.

Our testbed contains two types of randomly generated rank-one knapsack problems up to $n = 100,000$ variables. In the first type, the vectors a and c are integral and generated uniformly from the same interval. We denote this type by TypeI. In the second type (TypeII), the vectors a and c are positive and negative randomly generated integral vectors, respectively. In Table 1, we summarize the parameter values for problem instances.

As a well-known general convex quadratic programming solver, we chose CPLEX (ver. 12.9) to compare our results.

Based on our numerical results, we set the quadratic programming solver of CPLEX (qpmethod option) to the barrier. The barrier convergence tolerance, convergetol, is set to $1e-12$ (The default value is convergetol $= 1e-6$). It should be noted here that this setting is applied after we found that the default value leads to the optimal solutions that have components with a "meaningful" distance to their correct values. Another point is that for other optimizers such as primal and dual, CPLEX found the optimal solution in correct precision, but the running time is too long for large instances. For brevity, we do not report details related to these experiments.

After completing our experimental tests, we found in [10] that the sparsity of the Hessian matrix influences the performance of CPLEX. To increase the performance, we reformulate our problem as

$$\min\left\{\frac{1}{2}y^2 - c^\top x \ : \ \mathbf{1}^\top x - y = 0,\ a^\top x = b,\ 0 \le x \le u\right\}.$$

We denote the results corresponding to running CPLEX on the original problem and the aforementioned modification, respectively, by CPLEX$_{org}$ and CPLEX$_{ref}$.

Table 2 shows the average running time for five runs of each algorithm/-solver for each problem size. Dash sign, "-", denoted the algorithm/solver encounters out-of-memory status.

In all cases, Algorithm 2 outperforms CPLEX$_{org}$. For instances up to $n = 5000$, our algorithm and CPLEX$_{ref}$ have the same running time, whereas for larger instances, CPLEX$_{ref}$ has smaller running time.

Table 2: A comparison of running times (in seconds) between our algorithm and CPLEX$_{org}$ and CPLEX$_{ref}$.

n		Our algorithm	CPLEX$_{org}$	CPLEX$_{ref}$
1000	TypeI	0.06	0.09	0.01
	TypeII	0.01	0.06	0.02
1500	TypeI	0.04	0.15	0.02
	TypeII	0.02	0.13	0.02
2000	TypeI	0.04	0.27	0.02
	TypeII	0.02	0.27	0.02
5000	TypeI	0.09	2.21	0.02
	TypeII	0.06	2.12	0.03
10000	TypeI	0.26	16.26	0.04
	TypeII	0.23	16.95	0.05
15000	TypeI	0.62	61.20	0.10
	TypeII	0.63	65.88	0.10
20000	TypeI	1.16	-	1.20
	TypeII	0.88	-	1.02
50000	TypeI	3.22	-	0.11
	TypeII	3.19	-	0.11
100000	TypeI	12.19	-	0.14
	TypeII	11.31	-	0.17

5 Conclusions

In this paper, we proposed a two-phase algorithm for rank-one QKPs. To this end, we studied the solution structure of the problem when it has no resource constraint. Indeed, we proposed an $O(n \log n)$ algorithm to solve this problem. We then used the solution structure to propose an $O(n^2 \log n)$ line-sweep algorithm that finds an interval that contains the optimal Lagrange multiplier. Then, the estimated optimal interval was explored for computing the optimal solution with the desired accuracy. Our computational tests showed that our algorithm has better running time than CPLEX when CPLEX is used to solve the original problem. After a reformulation of the problem, CPLEX outperforms our algorithm for instances with $n \geq 5000$.

Acknowledgements

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

References

1. Bitran, G.R. and Hax, A.C.*Disaggregation and resource allocation using convex knapsack problems with bounded variables*, Management Sci. 27(4) (1981) 431–441.

2. Brucker, P. *An $O(n)$ algorithm for quadratic knapsack problems*, Oper. Res. Lett. 3(3) (1984) 163–166.

3. Cominetti, R., Mascarenhas, W.F. and Silva, P.J.S. *A Newton's method for the continuous quadratic knapsack problem*, Math. Program. Comput. 6(2) (2014) 151–169.

4. Dai, Y-H. and Fletcher, R. *New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds*, Math. Program. 106(3) (2006) 403–421.

5. de Berg, M., Cheong, O., van Kreveld, M. and Overmars, M. *Computational geometry. Algorithms and applications.* Third edition. Springer-Verlag, Berlin, 2008.

6. di Serafino, D., Toraldo, G., Viola, M. and Barlow, J. *A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables*, SIAM J. Optim. 28(4) (2018) 2809–2838.

7. Dussault, J-P., Ferland, J.A. and Lemaire, B. *Convex quadratic programming with one constraint and bounded variables*, Math. Program. 36(1) (1986) 90–104.

8. Fletcher, R. *Augmented lagrangians, box constrained QP and extensions*, IMA J. Numer. Anal. 37(4) (2017) 1635–1656.

9. Helgason, R., Kennington, J. and Lall, H. *A polynomially bounded algorithm for a singly constrained quadratic program*, Math. Program. 18(3) (1980) 338–343.

10. IBM, *Cplex performance tuning for quadratic programs*, https://www.ibm.com/support/pages/node/397129?mhsrc=ibmsearch_a&mhq=CPLEXPerformanceTuningforQuadraticPrograms, June 2018, [Online; accessed 23-January-2022].

11. Liu, M. and Liu, Y-J. *Fast algorithm for singly linearly constrained quadratic programs with box-like constraints*, Comput. Optim. Appl. 66(2) (2017) 309–326.

12. Pardalos, P.M., Ye, Y., and Han, C-G. *Algorithms for the solution of quadratic knapsack problems*, Linear Algebra Appl. 152 (1991), 69–91.

13. Patriksson, M. *A survey on the continuous nonlinear resource allocation problem*, European J. Oper. Res. 185(1) (2008) 1–46.

14. Patriksson, M. and Strömberg, C. *Algorithms for the continuous nonlinear resource allocation problem—new implementations and numerical studies*, European J. Oper. Res. 243(3) (2015) 703–722.

15. Robinson, A.G., Jiang, N. and Lerme, C.S. *On the continuous quadratic knapsack problem*, Math. program. 55(1-3) (1992) 99–108.

16. Sharkey, T.C. and Romeijn, H.E. *A class of nonlinear nonseparable continuous knapsack and multiple-choice knapsack problems*, Math. Program. 126(1) (2011) 69–96.

How to cite this article
S.E. Monabbati . *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 567-584. doi: 10.22067/ijnao.2022.70644.1096.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 585–606
DOI:10.22067/ijnao.2022.76489.1129
https://ijnao.um.ac.ir/

Research Article

Numerical solution of nonlinear fractional Riccati differential equations using compact finite difference method

H. Porki, M. Arabameri*, and R. Gharechahi

Abstract

This paper aims to apply and investigate the compact finite difference methods for solving integer-order and fractional-order Riccati differential equations. The fractional derivative in the fractional case is described in the Caputo sense. In solving the Riccati equation, we first approximate first-order derivatives using the approach of compact finite difference. In this way, the system of nonlinear equations is obtained, which solves the Riccati equation. In addition, we examine the convergence analysis of the proposed approach for the fractional and nonfractional cases and prove that the methods are convergent under some suitable conditions. Examples are also given to illustrate the efficiency of our method compared to other methods.

AMS subject classifications (2020): 34B15, 33F05, 65D20, 74S20.

Keywords: Fractional Riccati equation; Caputo fractional derivative; Compact finite difference methods.

* Corresponding author

Received 1 May 2022; revised 26 June 2022; accepted 23 July 2022

Homayoon Porki
Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran. e-mail: homayoun.porki@gmail.com

Maryam Arabameri
Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran. e-mail: arabameri@math.usb.ac.ir

Raziyeh Gharechahi
Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran. e-mail: r.gharechahi_64@yahoo.com

1 Introduction

In recent years, there has been a growing interest in fractional computation [14, 26, 31, 33, 34]. Fractional differential equations have become increasingly important as they have applications in various fields of science and engineering [13]. Numerous phenomena in fluid mechanics, viscoelasticity, chemistry, physics, finance, and other sciences can be described successfully by models using mathematical tools of fractional calculation, that is, the theory of fractional-order derivatives and integrals. Much important work on theoretical analysis [38, 10] has been carried out, but the analytical solutions of most fractional differential equations cannot be achieved explicitly. Numerical solution strategies based on convergence and stability analysis were used by many authors [12, 11, 13, 16, 20, 35, 36, 39, 41, 22]. Liu has carried out extensive research on the finite difference method of fractional differential equations [22, 23, 24]. The two most frequently used are the Riemann–Liouville and Caputo type. The difference between the two definitions is in the order of evaluation [29].

In this paper, we consider the following Riccati equation:

$$\begin{cases} u'(x) = p(x) + q(x)u(x) + r(x)u^2(x), & 0 < x < T, \\ u(0) = 0. \end{cases} \tag{1}$$

Also, we consider the following fractional Riccati equation:

$$D^\alpha u(x) = p(x) + q(x)u(x) + r(x)u^2(x), \qquad 0 < \alpha \leq 1, \qquad 0 < x < T, \tag{2}$$

along with the initial condition

$$u(0) = 0, \tag{3}$$

where $x \in \mathbb{R}$ and $p(x)$, $q(x)$, and $r(x)$ are known functions. Moreover, D^α is the Caputo derivative operator of the fractional-order α, which is defined as below:

$$D^\alpha u(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x (x-s)^{-\alpha} u'(s)\, ds. \tag{4}$$

In the past, two scholars, Bernoulli (1654-1705) and Riccati (1676-1754) introduced and assessed a particular case of differential equations (2). The Riccati differential equations (RDEs) and fractional Riccati differential equations (FRDEs) are used in many physical phenomena. Such applications can include control systems, robust stabilization, diffusion problems, network synthesis, optimal filtering, stochastic theory, controls, financial mathematics, optimal control, river flows, robust stabilization, financial mathematics

dynamic games, linear systems with Markovian jumps, stochastic control, econometric models, and invariant embedding [32, 28, 4, 19, 15, 9, 21, 5, 30]. Many researchers have used numerical approaches to solve the RDEs and FRDEs. Some standard procedures can be referenced, including the differential transform method [7], series solutions Adomian's decomposition method [1], Homotopy perturbation method [1], variational iteration method [18], Homotopy analysis method [37], piecewise spectral-collocation method [6], and so on [8, 27, 25, 3].
This paper aims to obtain numerical solutions to (1)–(3) using a high-order compact finite difference approach.

Several researchers have employed the compact finite difference method to solve fractional differential equations. Du, Cao, and Sun [14] have used the compact finite difference method to solve the fractional diffusion-wave equation. Gao and Sun [17] have also employed the compact finite difference method to solve the fractional sub-diffusion equation. They have also proved the stability and convergence of their method. Cui [13] solved the one-dimensional fractional diffusion equation via a high-order compact finite difference scheme and obtained a fully discrete implicit system by Grunwald–Letnikov's discretization of the Riemann–Liouville derivative.

The present study is organized as follows: In Sections 2, 3, and 4, the compact finite difference methods are reviewed and applied to solve (1)–(3). Also, their convergence is discussed. In Section 5, the numerical results obtained by the proposed methods are presented. We also compare the results of our approach and those of the proposed methods in [2]. The conclusion and the advantages of the proposed technique are presented in Section 6.

2 Compact finite difference scheme

In this work, our primary goal is to apply the compact finite difference method to solve (1)–(3). For this, we first subdivide the range $0 \leq x \leq T$ to N equal partitions with step length h as follows:

$$x_0 = 0, \qquad x_i = ih, \qquad i = 0, 1, \ldots, N, \qquad h = \frac{T}{N}. \tag{5}$$

Set

$$u_i \approx u(x_i), \qquad u_i' \approx u'(x_i).$$

For the first derivatives, the following compact finite difference scheme was given in [40]:

$$\begin{cases} 4u'_1 + u'_2 = \frac{1}{h}(\frac{-11}{12}u_0 - 4u_1 + 6u_2 - \frac{4}{3}u_3 + \frac{1}{4}u_4), \\ u'_{i-1} + 4u'_i + u'_{i+1} = \frac{3}{h}(-u_{i-1} + u_{i+1}), \qquad i = 1, \ldots, N-1, \\ u'_{N-2} + 4u'_{N-1} = \frac{1}{h}(-\frac{1}{4}u_{N-4} + \frac{4}{3}u_{N-3} - 6u_{N-2} + 4u_{N-1} + \frac{11}{12}u_N). \end{cases} \tag{6}$$

All above relations have the accuracy of $O(h^4)$. The matrix form for (23) is

$$A_1 u' = \frac{1}{h} B_1 u, \tag{7}$$

where

$$A_1 = \begin{pmatrix} 0 & 4 & 1 & 0 & \ldots & 0 \\ 1 & 4 & 1 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ldots & 0 & 1 & 4 & 1 \\ 0 & \ldots & 0 & 1 & 4 & 0 \end{pmatrix}_{(N+1)\times(N+1)},$$

$$B_1 = \begin{pmatrix} -\frac{11}{12} & -4 & 6 & \frac{4}{3} & \frac{1}{4} & 0 & \ldots & 0 \\ -3 & 0 & 3 & 0 & 0 & 0 & \ldots & 0 \\ 0 & -3 & 0 & 3 & 0 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ldots & 0 & 0 & 0 & -3 & 0 & 3 \\ 0 & \ldots & 0 & -\frac{1}{4} & \frac{4}{3} & -6 & 4 & \frac{11}{12} \end{pmatrix}_{(N+1)\times(N+1)}.$$

Also, $u = [u_0, u_1, \ldots, u_N]^T$ and $u' = [u'_0, u'_1, \ldots, u'_N]^T$.

Lemma 1. The coefficient matrix A_1 is invertible.

Proof. Let us expand A_1 along the first column. Then

$$det(A_1) = -det \begin{pmatrix} 4 & 1 & 0 & \ldots & 0 \\ 1 & 4 & 1 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 1 & 4 & 1 \\ 0 & \ldots & 1 & 4 & 0 \end{pmatrix}_{N\times N}.$$

Now, by expanding along the last column, we have

$$det(A_1) = (-1)^N det \begin{pmatrix} 4 & 1 & 0 & \ldots & 0 \\ 1 & 4 & 1 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 1 & 4 & 1 \\ 0 & \ldots & 0 & 1 & 4 \end{pmatrix}_{(N-1)\times(N-1)} \neq 0.$$

□

According to Lemma 1, from (24), we have $u' = \frac{1}{h}A_1^{-1}B_1 u$. By defining $C = A_1^{-1}B_1$, the following relation holds for u':

$$u' = \frac{1}{h} C u, \tag{8}$$

and in the component form, we have

$$u'_i = \frac{1}{h} \sum_{j=0}^{N} c_{i+1,j+1} u_j, \qquad i = 0, \ldots, N. \tag{9}$$

Lemma 2. The coefficient matrix B_1 is invertible.

Proof. Let us expand B_1 along the first row. Then

$$det(B_1) = -det \begin{pmatrix} -3 & 0 & 3 & 0 & 0 & 0 & \ldots & 0 \\ 0 & -3 & 0 & 3 & 0 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ldots & 0 & 0 & 0 & -3 & 0 & 3 \\ 0 & \ldots & 0 & -\frac{1}{4} & \frac{4}{3} & -6 & 4 & \frac{11}{12} \end{pmatrix}_{N \times N}.$$

Now, by expanding along the last row, we have

$$det(B_1) = (-1)^N det \begin{pmatrix} -3 & 0 & 3 & 0 & 0 & 0 & \ldots & 0 \\ 0 & -3 & 0 & 3 & 0 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ldots & 0 & 0 & 0 & -3 & 0 & 3 \end{pmatrix}_{(N-1)\times(N-1)} \neq 0.$$

□

According to Lemma 2, it follows that the matrix C is invertible.

3 Compact finite difference scheme for Riccati problem in $\alpha = 1$ case and its convergence

This section uses the compact finite difference scheme for the nonfractional Riccati problem and investigates its convergence. Consider the subsequent classical Riccati initial value problem

$$u'(x) = p(x) + q(x)u(x) + r(x)u^2(x), \qquad 0 < x < T. \tag{10}$$

Its initial condition is

$$u(0) = 0. \tag{11}$$

Using (10), we have

$$u'_0 = p(x_0). \tag{12}$$

So, using (9), equation (12) can be written as

$$\frac{1}{h}\sum_{j=0}^{N} c_{1,j+1}u_j = p(x_0). \tag{13}$$

For $x = x_i$, one can write (10) as

$$u'(x_i) = p(x_i) + q(x_i)u(x_i) + r(x_i)u^2(x_i), \qquad i = 1, \ldots, N. \tag{14}$$

Thus from (9)

$$\frac{1}{h}\sum_{j=0}^{N} c_{i+1,j+1}u_j - p(x_i) - q(x_i)u_i - r(x_i)u_i^2 = 0, \qquad i = 1, \ldots, N. \tag{15}$$

Equations (13) and (15) form a system including $N+1$ equations and $N+1$ unknowns $u_0, u_1, \ldots, u_N$, that can be solved by Maple software.

Now, the convergence analysis of the proposed method for (10) along with initial conditions (11) is investigated.

Theorem 1. Let $U = [u(x_0), u(x_1), \ldots, u(x_N)]^T$ be the vector of exact solution to (1) along with its initial condition, and let $u = [u_0, u_1, \ldots, u_N]^T$ be the numerical solution at the same points obtained by (13) and (15). Then

$$\|E\| \leq O(h^2), \tag{16}$$

provided $h\|C^{-1}\|\|M\| \leq 1$, where $E = [e_0, e_1, \ldots, e_N]^T$ and $e_i = u(x_i) - u_i$, $i = 0, \ldots, N$ ($\|\cdot\|$ is the infinity norm).

Proof. According to (13) and (15), for a numerical solution, we have

$$\begin{cases} \frac{1}{h}\sum_{j=0}^{N} c_{1,j+1}u_j = p(x_0), \\ \frac{1}{h}\sum_{j=0}^{N} c_{i+1,j+1}u_j - p(x_i) - q(x_i)u_i - r(x_i)u_i^2 = 0, \qquad i = 1, \ldots, N, \end{cases} \tag{17}$$

and for an exact solution, we have

$$\begin{cases} \frac{1}{h}\sum_{j=0}^{N} c_{1,j+1}u(x_j) = p(x_0) + O(h^4), \\ \frac{1}{h}\sum_{j=0}^{N} c_{i+1,j+1}u(x_j) - p(x_i) - q(x_i)u(x_i) - r(x_i)u^2(x_i) = O(h^4), \quad i = 1, \ldots, N. \end{cases} \tag{18}$$

By subtracting (17) and (18), one concludes that

$$\begin{cases} \frac{1}{h}\sum_{j=0}^{N} c_{1,j+1}(u(x_j) - u_j) = O(h^4), \\ \frac{1}{h}\sum_{j=0}^{N} c_{i+1,j+1}(u(x_j) - u_j) - q(x_i)(u(x_i) - u_i) \\ \quad -r(x_i)(u^2(x_i) - u_i^2) = O(h^4), \qquad i = 1, \ldots, N. \end{cases} \tag{19}$$

Using the Taylor expansion, we have

$$u^2(x_i) - u_i^2 = \frac{\partial u^2}{\partial u}|_{x=x_i}(u(x_i) - u_i) + O(h^2), \qquad i = 1, \ldots, N. \tag{20}$$

In relation (19), we have

$$\begin{cases} \frac{1}{h}\sum_{j=0}^{N} c_{1,j+1}(u(x_j)-u_j) = O(h^4), \\ \frac{1}{h}\sum_{j=0}^{N} c_{i+1,j+1}(u(x_j)-u_j) - q(x_i)(u(x_i)-u_i) \\ \quad -2r(x_i)u(x_i)(u(x_i)-u_i) = O(h^4)+O(h^2), \qquad i=1,\ldots,N. \end{cases} \tag{21}$$

Thus, one concludes that

$$\begin{cases} \sum_{j=0}^{N} c_{1,j+1}e_j = O(h^5), \\ \sum_{j=0}^{N} c_{i+1,j+1}e_j - hq(x_i)e_i - 2hr(x_i)u(x_i)e_i = O(h^3), \quad i=1,\ldots,N, \end{cases} \tag{22}$$

where $e_j = u(x_j) - u_j$, $j = 0,\ldots,N$, and $u_i \approx u(x_i)$. Therefore, (22) can be written as

$$\begin{cases} c_{11}e_0 + c_{12}e_1 + c_{13}e_2 + \cdots + c_{1,N+1}e_N = O(h^5), \\ c_{21}e_0 + c_{22}e_1 + c_{23}e_2 + \cdots + c_{2,N+1}e_N - hq(x_1)e_1 - 2hr(x_1)u(x_1)e_1 = O(h^3), \\ c_{31}e_0 + c_{32}e_1 + c_{33}e_2 + \cdots + c_{3,N+1}e_N - hq(x_2)e_2 - 2hr(x_2)u(x_2)e_2 = O(h^3), \\ \vdots \\ c_{N+1,1}e_0 + c_{N+1,2}e_1 + c_{N+1,3}e_2 + \cdots + c_{N+1,N+1}e_N \\ -hq(x_N)e_N - 2hr(x_N)u(x_N)e_N = O(h^3). \end{cases} \tag{23}$$

The matrix form of the above equations is as follows:

$$[C - hQ - hRJ]E = T, \tag{24}$$

where $Q = diag(0, q(x_1), \ldots, q(x_N))$, $R = diag(0, r(x_1), \ldots, r(x_N))$, $J = diag(0, 2u(x_1), \ldots, 2u(x_N))$, and

$$T = \begin{pmatrix} O(h^5) \\ O(h^3) \\ O(h^3) \\ \vdots \\ O(h^3) \end{pmatrix}_{(N+1)\times 1} \qquad C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1,N+1} \\ c_{21} & c_{22} & \cdots & c_{2,N+1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N+1,1} & c_{N+1,2} & \cdots & c_{N+1,N+1} \end{pmatrix}_{(N+1)\times(N+1)}. \tag{25}$$

By replacing $M = Q + RJ$ in relation (24), we have $[C - hM]E = T$. Because C is invertible, we can write

$$(I - hC^{-1}M)E = C^{-1}T. \tag{26}$$

Now, if we assume $h\|C^{-1}\|\|M\| \leq 1$, then we conclude the matrix $I - hC^{-1}M$ is invertible. By the geometric series theorem, we have

$$\|(I - hC^{-1}M)^{-1}\| \leq \frac{1}{1 - h\|C^{-1}\|\|M\|}. \tag{27}$$

From (26), we have $E = (I - hC^{-1}M)^{-1}C^{-1}T$. Thus $\|E\| \leq \|(I - hC^{-1}M)^{-1}\|\|C^{-1}\|\|T\|$.

Now from relation (27), we can write $\|E\| \leq \frac{1}{1-h\|C^{-1}\|\|M\|}\|C^{-1}\|\|T\|$. Because $\|T\| \equiv O(h^3)$, we can derive $\|E\| \leq \frac{O(h^3)}{O(h)} \equiv O(h^2)$.

□

4 Implement the compact finite difference scheme for the fractional Riccati problem and its convergence

In this section, we introduce a compact finite difference scheme for the fractional Riccati problem of order $0 < \alpha < 1$. According to (2), we rewrite the Caputo derivative in $x = x_i$, $i = 1, \ldots, N$, as

$$D^{\alpha}u(x_i) = \frac{1}{\Gamma(1-\alpha)}\sum_{k=0}^{i-1}\int_{x_k}^{x_{k+1}}\frac{u'(s)}{(x_i - s)^{\alpha}}ds. \tag{28}$$

Now, the above equation can be written as

$$\begin{aligned} D^{\alpha}u(x_i) &\approx \frac{1}{\Gamma(1-\alpha)}\sum_{k=0}^{i-1}\int_{x_k}^{x_{k+1}} u_i'(x_i - s)^{-\alpha}ds \\ &= \frac{1}{\Gamma(1-\alpha)}u_i'\sum_{k=0}^{i-1}\int_{x_k}^{x_{k+1}}(x_i - s)^{-\alpha}ds \\ &= \frac{1}{\Gamma(1-\alpha)}u_i'\sum_{k=0}^{i-1}[\frac{(x_i - x_k)^{1-\alpha} - (x_i - x_{k+1})^{1-\alpha}}{1-\alpha}]. \end{aligned} \tag{29}$$

Substituting $x_i = ih$ in (29), we have

$$\begin{aligned} D^{\alpha}u(x_i) &\approx \frac{1}{\Gamma(1-\alpha)}u_i'\sum_{k=0}^{i-1}[\frac{h^{1-\alpha}((i-k)^{1-\alpha} - (i-k-1)^{1-\alpha})}{1-\alpha}] \\ &= \frac{u_i'}{h^{\alpha-1}\Gamma(2-\alpha)}\sum_{k=0}^{i-1}a_{i-k}, \end{aligned} \tag{30}$$

where $a_{i-k} = (i-k)^{1-\alpha} - (i-k-1)^{1-\alpha}$, $i = 1, \ldots, N$ and $k = 0, \ldots, i-1$.

Thus, the solution to (2) can be approximated using the following equations:

$$\frac{u_i'}{h^{\alpha-1}\Gamma(2-\alpha)}\sum_{k=0}^{i-1}a_{i-k}=p(x_i)+q(x_i)u_i+r(x_i)u_i^2,\quad 0<\alpha<1,\ i=1,\dots,N, \tag{31}$$

where $u_i'=\frac{1}{h}\sum\limits_{j=0}^{N}c_{i+1,j+1}u_j$, $i=1,\dots,N$. In the matrix form, (31) is equivalent to

$$Fu'=\rho(G+Qu+Ru^2), \tag{32}$$

where $\rho=h^{\alpha-1}\Gamma(2-\alpha)$, $Q=diag(q(x_1),\dots,q(x_N))$, $R=diag(r(x_1),\dots,r(x_N))$, $u=[u_1,\dots,u_N]^T$, $u'=[u_1',\dots,u_N']^T$, $G=\begin{pmatrix}p(x_1)\\p(x_2)\\\vdots\\p(x_N)\end{pmatrix}$, and

$$F=\begin{pmatrix}a_1 & 0 & 0 & 0 & \dots & 0\\ 0 & a_1+a_2 & 0 & 0 & \dots & 0\\ 0 & 0 & a_1+a_2+a_3 & 0 & \dots & 0\\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots\\ 0 & 0 & \dots & 0 & a_1+a_2+\dots+a_{N-1} & 0\\ 0 & 0 & \dots & 0 & 0 & a_1+a_2+\dots+a_N\end{pmatrix}. \tag{33}$$

For $i=1,\dots,N$, (31) can be used to form a system including N equations and N unknowns $u_1,\dots,u_N$, that can be solved by Maple software.

Now, we discuss the issue of convergence. For convergence analysis of the fractional case, we need the following Lemma.

Lemma 3. [35] Suppose $u\in C^2[0,x_i]$. Then

$$\begin{aligned}&\Big|\int_0^{x_i}\frac{u'(s)}{(x_i-s)^{\alpha}}ds-\sum_{k=0}^{i-1}u_i'\int_{x_k}^{x_{k+1}}(x_i-s)^{-\alpha}ds\Big|\\ &\quad\le\frac{1}{1-\alpha}\Big[\frac{1-\alpha}{12}+\frac{2^{2-\alpha}}{2-\alpha}-(1+2^{-\alpha})\Big]\max_{0\le s\le x_i}|u''(s)|h^{2-\alpha}.\end{aligned} \tag{34}$$

From (30), we have

$$D^{\alpha}u(x_i)=\frac{1}{h^{\alpha-1}\Gamma(2-\alpha)}\sum_{k=0}^{i-1}a_{i-k}u'(x_i)+R_i,\qquad i=1,\dots,N, \tag{35}$$

where according to Lemma 3

$$R_i\le\frac{1}{1-\alpha}\Big[\frac{1-\alpha}{12}+\frac{2^{2-\alpha}}{2-\alpha}-(1+2^{-\alpha})\Big]\max_{0\le s\le x_i}|u''(s)|h^{2-\alpha}. \tag{36}$$

For $x=x_i$, by replacing (35) into (2), we have

$$\sum_{k=0}^{i-1} a_{i-k}u'(x_i) = \rho(p(x_i) + q(x_i)u(x_i) + r(x_i)u^2(x_i)) + \tilde{R}_i, \qquad i = 1, \ldots, N, \tag{37}$$

where $\rho = h^{\alpha-1}\Gamma(2-\alpha)$ and $\tilde{R}_i = h^{\alpha-1}\Gamma(2-\alpha)R_i, i = 1, \ldots, N$.

In the matrix form, (37) is equivalent to

$$FU' = \rho(G + QU + RU^2) + \tilde{R}, \tag{38}$$

where F is the matrix defined in relation (33), $U' = [u'(x_1), \ldots, u'(x_N)]^T$, $U = [u(x_1), \ldots, u(x_N)]^T$, and $\tilde{R} = h^{\alpha-1}\Gamma(2-\alpha)[R_1, \ldots, R_N]^T$.

Theorem 2. Let $U = [u(x_1), \ldots, u(x_N)]^T$ be the vector of exact solution to (2) along with its initial condition at points $x_0, x_1, \ldots, x_N$, and let $u = [u_1, \ldots, u_N]^T$ be the numerical solution obtained by (31). Then

$$\|E\| \leq O(h^{2-\alpha}), \tag{39}$$

provided $h\|C^{-1}\|\|M + N\| \leq 1$, where $E = U - u$ and

$$J = \begin{pmatrix} 2u(x_1) & 0 & \ldots & 0 \\ 0 & 2u(x_2) & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 2u(x_N) \end{pmatrix}.$$

Proof. According to (38) and (32), for the exact and numerical solutions, we have

$$\begin{cases} FU' = \rho(G + QU + RU^2) + \tilde{R}, \\ Fu' = \rho(G + Qu + Ru^2). \end{cases} \tag{40}$$

By using (40), one concludes that

$$F(U' - u') = \rho(Q(U - u) + R(U^2 - u^2)) + \tilde{R}. \tag{41}$$

Therefore, by replacing $u' = \frac{1}{h}Cu$ from (8) and $U' = \frac{1}{h}CU + T_1$ into (41), we have

$$\frac{1}{h}C(U - u) - \rho F^{-1}Q(U - u) - \rho F^{-1}R(U^2 - u^2) = F^{-1}\tilde{R} + T_1, \tag{42}$$

where $T_1 \equiv O(h^4)$ is the local truncation error of system (23).

Moreover, $U^2 - u^2$ can be written as

$$U^2 - u^2 = \begin{pmatrix} u^2(x_1) - u_1^2 \\ u^2(x_2) - u_2^2 \\ \vdots \\ u^2(x_N) - u_N^2 \end{pmatrix} = JE + T_2, \tag{43}$$

where

$$T_2 = \begin{pmatrix} O(h^2) \\ O(h^2) \\ \vdots \\ O(h^2) \end{pmatrix}.$$

Therefore, by replacing (43) into (42), we have

$$\frac{1}{h}CE - \rho F^{-1}QE - \rho F^{-1}RJE = F^{-1}\tilde{R} + T_1 + \rho F^{-1}RT_2. \tag{44}$$

By inserting relations $M = \rho F^{-1}Q$ and $N = \rho F^{-1}R$ into (44), it can be written as

$$(C - hM - hN)E = h(F^{-1}\tilde{R} + T_1 + NT_2), \tag{45}$$

$$(I - hC^{-1}(M + N))E = hC^{-1}(F^{-1}\tilde{R} + T_1 + NT_2). \tag{46}$$

Now, if $h\|C^{-1}\|\|M + N\| \leq 1$, then $(I - hC^{-1}(M + N)$ is invertible and

$$E = h(I - hC^{-1}(M + N))^{-1}C^{-1}(F^{-1}\tilde{R} + T_1 + NT_2),$$

$$\|E\| \leq h\|(I - hC^{-1}(M + N))^{-1}\|\|C^{-1}\|(\|F^{-1}\|\|\tilde{R}\| + \|T_1\| + \|N\|\|T_2\|).$$

It follows that

$$\|E\| \leq \frac{h\|C^{-1}\|(\|F^{-1}\|\|\tilde{R}\| + \|T_1\| + \|N\|\|T_2\|)}{1 - h\|C^{-1}\|\|M + N\|}. \tag{47}$$

Therefore, using the relations $\tilde{R} = h^{\alpha-1}\Gamma(2 - \alpha)R$ and (36), we have $\|\tilde{R}\| \equiv O(h)$, so

$$\|E\| \leq \frac{O(h^2)}{O(h^\alpha)} + \frac{O(h^5)}{O(h^\alpha)} + \frac{O(h^3)}{O(h^\alpha)} = O(h^{2-\alpha}) + O(h^{5-\alpha}) + O(h^{3-\alpha}) \equiv O(h^{2-\alpha}). \tag{48}$$

□

5 Numerical results

This section applies our compact finite difference schemes to two examples to illustrate their effectiveness. Maple software is used for obtaining numerical results.

Example 1. Consider the following fractional RDE as the first example:

$$\begin{cases} D^{\alpha}u(x) = 1 - u^2(x), & 0 < \alpha \leq 1, \quad 0 < x < T, \\ u(0) = u_0 = 0. \end{cases} \tag{49}$$

The exact solution is $u(x) = \frac{\exp(2x)-1}{\exp(2x)+1}$ for $\alpha = 1$; see [2].

In Figure 1, a comparison between the exact solution for $\alpha = 1$ and the numerical solution for $\alpha = 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999, 1$, and $T = 1$ is shown. Table 1 presents numerical solutions at some points of $[0, 1]$ and for different values of α, at $T = 1$. Table 2 presents a comparison between the exact solution for $\alpha = 1$ and the numerical solution for $T = 10$. Also, Figure 2 shows a comparison between the exact solution for $\alpha = 1$ and the numerical solution for $T = 10$.

We have calculated the rate of convergence of our methods (denoted by ROC) with the following formula:

$$ROC = \log_2\left(\frac{Error^{2h}}{Error^{h}}\right). \tag{50}$$

Table 3 shows the obtained maximum errors and ROC for $\alpha = 1$, $T = 1$, and $N = 5, 10, 20, 40, 80, 160$. Also, Figure 3 shows the numerical and exact solutions for $\alpha = 1$, $T = 1$, and $N = 10$. The numerical rate of convergence is highly consistent with our theoretical analysis results.

In Table 4, we compare the approximate solution and exact solution of the present method with the trigonometric transform method (TTM) [2] at points $0.2, 0.4, 0.6, 0.8, 1$, for $\alpha = 1$. Also, in Table 5, we compare the error of solutions of the present method with TTM [2] for $\alpha = 1$.

Table 1: Exact solutions and numerical solutions of Example 1 for $N = 10$, $T = 1$, and $\alpha = 0.3, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999, 1$

α / x	0.3	0.6	0.7	0.8	0.9	0.95	0.99	0.999	1	Exact
0.1	5.38×10^{-1}	2.66×10^{-1}	2.06×10^{-1}	1.60×10^{-1}	1.25×10^{-1}	1.11×10^{-1}	1.01×10^{-1}	9.98×10^{-2}	9.96×10^{-2}	9.96×10^{-2}
0.2	7.43×10^{-1}	4.34×10^{-1}	3.53×10^{-1}	2.88×10^{-1}	2.37×10^{-1}	2.16×10^{-1}	2.00×10^{-1}	1.97×10^{-1}	1.97×10^{-1}	1.97×10^{-1}
0.3	8.34×10^{-1}	5.51×10^{-1}	4.66×10^{-1}	3.95×10^{-1}	3.37×10^{-1}	3.12×10^{-1}	2.95×10^{-1}	2.91×10^{-1}	2.91×10^{-1}	2.91×10^{-1}
0.4	8.83×10^{-1}	6.38×10^{-1}	5.57×10^{-1}	4.86×10^{-1}	4.27×10^{-1}	4.02×10^{-1}	3.84×10^{-1}	3.80×10^{-1}	3.79×10^{-1}	3.79×10^{-1}
0.5	9.14×10^{-1}	7.05×10^{-1}	6.31×10^{-1}	5.64×10^{-1}	5.07×10^{-1}	4.83×10^{-1}	4.66×10^{-1}	4.62×10^{-1}	4.62×10^{-1}	4.62×10^{-1}
0.6	9.34×10^{-1}	7.57×10^{-1}	6.91×10^{-1}	6.30×10^{-1}	5.79×10^{-1}	5.56×10^{-1}	5.40×10^{-1}	5.37×10^{-1}	5.37×10^{-1}	5.37×10^{-1}
0.7	9.48×10^{-1}	7.99×10^{-1}	7.41×10^{-1}	6.87×10^{-1}	6.41×10^{-1}	6.21×10^{-1}	6.07×10^{-1}	6.04×10^{-1}	6.04×10^{-1}	6.04×10^{-1}
0.8	9.58×10^{-1}	8.32×10^{-1}	7.82×10^{-1}	7.35×10^{-1}	6.95×10^{-1}	6.78×10^{-1}	6.66×10^{-1}	6.64×10^{-1}	6.64×10^{-1}	6.64×10^{-1}
0.9	9.65×10^{-1}	8.59×10^{-1}	8.16×10^{-1}	7.76×10^{-1}	7.42×10^{-1}	7.28×10^{-1}	7.18×10^{-1}	7.16×10^{-1}	7.16×10^{-1}	7.16×10^{-1}
1.0	9.71×10^{-1}	8.81×10^{-1}	8.44×10^{-1}	8.10×10^{-1}	7.82×10^{-1}	7.70×10^{-1}	7.63×10^{-1}	7.61×10^{-1}	7.61×10^{-1}	7.61×10^{-1}

Example 2. Let the following FRDE be the second example

$$D^{\alpha}u(x) = 1 + 2u(x) - u^2(x), \qquad 0 < \alpha \leq 1, \qquad 0 < x < T, \tag{51}$$

with the initial condition

$$u_0 = u(0) = 0. \tag{52}$$

The exact solution for $\alpha = 1$ is $u(x) = 1 + \sqrt{2}\tanh(\sqrt{2}x + \frac{1}{2}\log(\frac{\sqrt{2}-1}{\sqrt{2}+1}))$; see [2].

Table 2: Comparison between the exact solution and numerical solutions of Example 1 for $\alpha = 1$, $T = 10$, and $N = 100$

x	Numerical solution	Exact solution	Error
1	0.7615917576	0.7615941559	2.3983554×10^{-6}
2	0.9640223166	0.9640275800	5.2634336×10^{-6}
3	0.9950446865	0.9950547536	1.0067173×10^{-5}
4	0.9993096449	0.9993292997	1.9654751×10^{-5}
5	0.9998709681	0.9999092042	3.8236123×10^{-5}
6	0.9999133772	0.9999877116	7.4334413×10^{-5}
7	0.9998538332	0.9999983369	1.4450373×10^{-4}
8	0.9997188603	0.9999997749	2.8091453×10^{-4}
9	0.9994538522	0.9999999695	5.4611733×10^{-4}

Table 3: Maximum absolute errors and ROC of Example 1 for $\alpha = 1$, $T = 1$, and $N = 5, 10, 20, 40, 80, 160$

N	Maximum Absolute Error	ROC
5	7.95×10^{-4}	–
10	1.91×10^{-5}	5.38
20	9.47×10^{-7}	4.33
40	3.43×10^{-8}	4.79
80	1.16×10^{-9}	4.88
160	3.99×10^{-11}	4.87

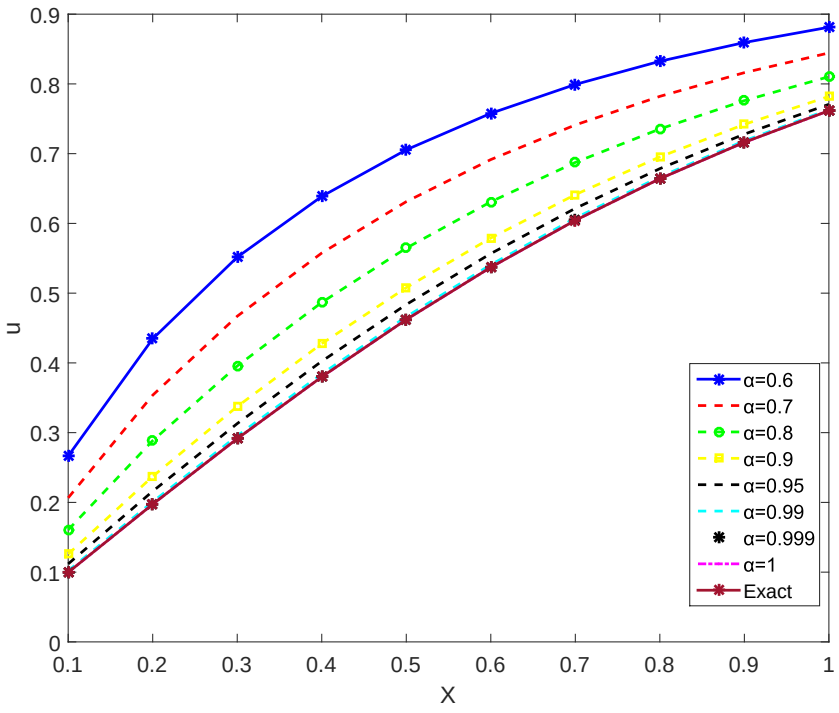


Figure 1: Comparison between the exact solution of Example 1 for $\alpha = 1$ and numerical solutions for $\alpha = 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999, 1$ and $T = 1$

Table 4: Comparison between the approximation solution and exact solution of the presented method with TTM [2] for $\alpha = 1$, $T = 1$, and $N = 10$ for Example 1

x	TTM [2]	proposed method	Exact
0.0	0.0	0.0	0.0
0.2	0.197773	0.197378	0.197374
0.4	0.380422	0.379951	0.379949
0.6	0.537449	0.537051	0.537050
0.8	0.664285	0.664036	0.664037
1.0	0.761671	0.761572	0.761594

In Figure 4, a comparison between the exact solution for $\alpha = 1$ and the numerical solution for $\alpha = 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999, 1$ and $T = 1$ is shown. Also, Table 6 presents numerical solutions at some points of $[0, 1]$ and for different values of α at $T = 1$.

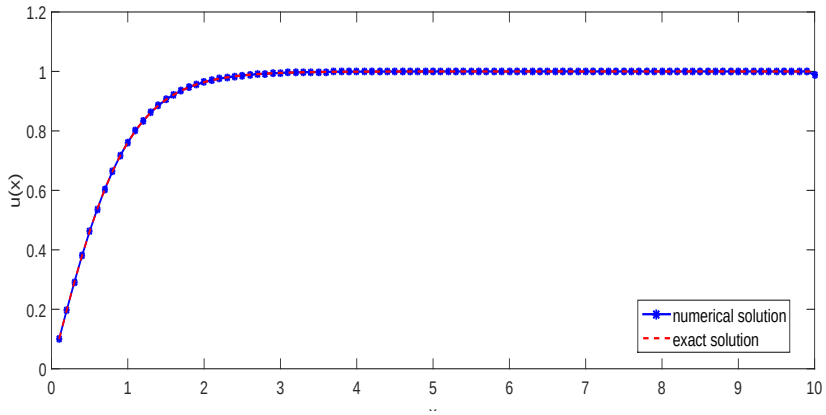


Figure 2: Comparison between the exact solution and numerical solutions of Example 1 for $\alpha = 1$, $T = 10$, and $N = 100$

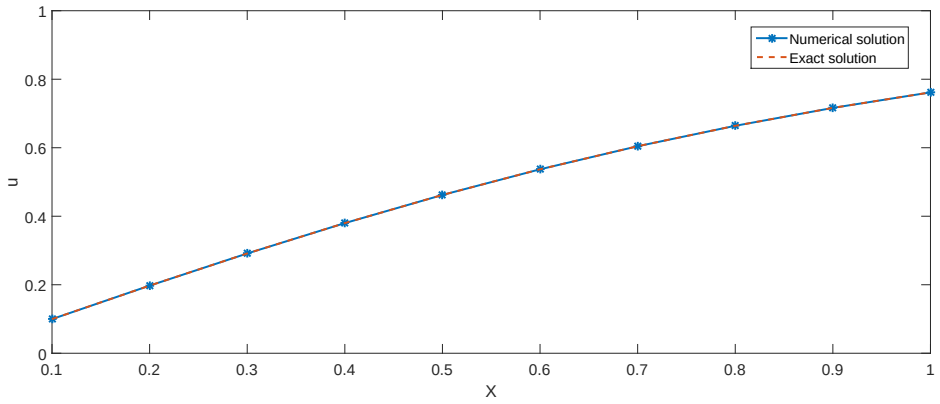


Figure 3: Comparison between the exact solution and numerical solutions of Example 1 for $\alpha = 1$, $T = 1$, and $N = 10$

Table 7 shows the obtained maximum errors and ROC for $\alpha = 1$, $T = 1$, and $N = 5, 10, 20, 40, 80, 160$ Also, Figure 5 shows the numerical and exact solutions for $\alpha = 1$, $T = 1$, and $N = 10$. The numerical rate of convergence is highly consistent with our theoretical analysis results.
Table 8 represents the present method and the achieved results of particle swarm optimization (PSO) [2], modified homotopy perturbation method (MHPM) [2], Chebyshev wavelets (CW) [2], fractional variational iteration method (FVI) [2], Legendre wavelets method (LWM) [2], and Pad´e-variational iteration method (PVI) [2].
Table 9 presents a comparison between the exact solution for $\alpha = 1$ and the

Table 5: Comparison between the absolute error of solution by our method with TTM [2] for $\alpha = 1$ and $T = 1$, for Example 1

x	Error of proposed method	Error of TTM [2]
0.0	0.0	0.0
0.2	1.4598×10^{-6}	7.2107×10^{-4}
0.4	1.5961×10^{-6}	1.7216×10^{-3}
0.6	4.6060×10^{-7}	2.7186×10^{-3}
0.8	1.1006×10^{-6}	3.3906×10^{-3}
1.0	1.9061×10^{-5}	3.6117×10^{-3}

Table 6: Exact solutions and Numerical solutions of Example 2 for $N = 10$, $T = 1$, and $\alpha = 0.3, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999, 1$

α / x	0.3	0.6	0.7	0.8	0.9	0.95	0.99	0.999	1	Exact
0.1	1.38	3.79×10^{-1}	2.65×10^{-1}	1.92×10^{-1}	1.44×10^{-1}	1.25×10^{-1}	1.13×10^{-1}	1.10×10^{-1}	1.10×10^{-1}	1.10×10^{-1}
0.2	1.92	7.21×10^{-1}	5.25×10^{-1}	3.94×10^{-1}	3.04×10^{-1}	2.70×10^{-1}	2.47×10^{-1}	2.42×10^{-1}	2.41×10^{-1}	2.41×10^{-1}
0.3	2.14	1.02	7.80×10^{-1}	6.05×10^{-1}	4.82×10^{-1}	4.35×10^{-1}	4.02×10^{-1}	3.95×10^{-1}	3.95×10^{-1}	3.95×10^{-1}
0.4	2.24	1.29	1.02	8.22×10^{-1}	6.74×10^{-1}	6.16×10^{-1}	5.76×10^{-1}	5.68×10^{-1}	5.67×10^{-1}	5.67×10^{-1}
0.5	2.30	1.51	1.25	1.03	8.75×10^{-1}	8.10×10^{-1}	7.66×10^{-1}	7.56×10^{-1}	7.55×10^{-1}	7.55×10^{-1}
0.6	2.34	1.70	1.45	1.24	1.07	1.01	9.64×10^{-1}	9.54×10^{-1}	9.53×10^{-1}	9.53×10^{-1}
0.7	2.36	1.84	1.62	1.43	1.27	1.20	1.16	1.15	1.15	1.15
0.8	2.37	1.96	1.77	1.59	1.45	1.39	1.35	1.34	1.34	1.34
0.9	2.38	2.05	1.89	1.74	1.61	1.56	1.53	1.52	1.52	1.52
1.0	2.39	2.12	1.99	1.87	1.76	1.72	1.69	1.69	1.68	1.68

numerical solution for $T = 8$. Also, Figure 6 shows a comparison between the exact solution for $\alpha = 1$ and the numerical solution for $T = 8$ and $N = 80$.

Table 7: Maximum absolute errors and ROC of Example 2 for $\alpha = 1$, $T = 1$, and $N = 5, 10, 20, 40, 80, 160$

N	Maximum Absolute Error	ROC
5	6.35×10^{-3}	–
10	3.63×10^{-5}	7.45
20	3.63×10^{-6}	3.32
40	1.73×10^{-7}	4.39
80	7.05×10^{-9}	4.62
160	2.98×10^{-10}	4.57

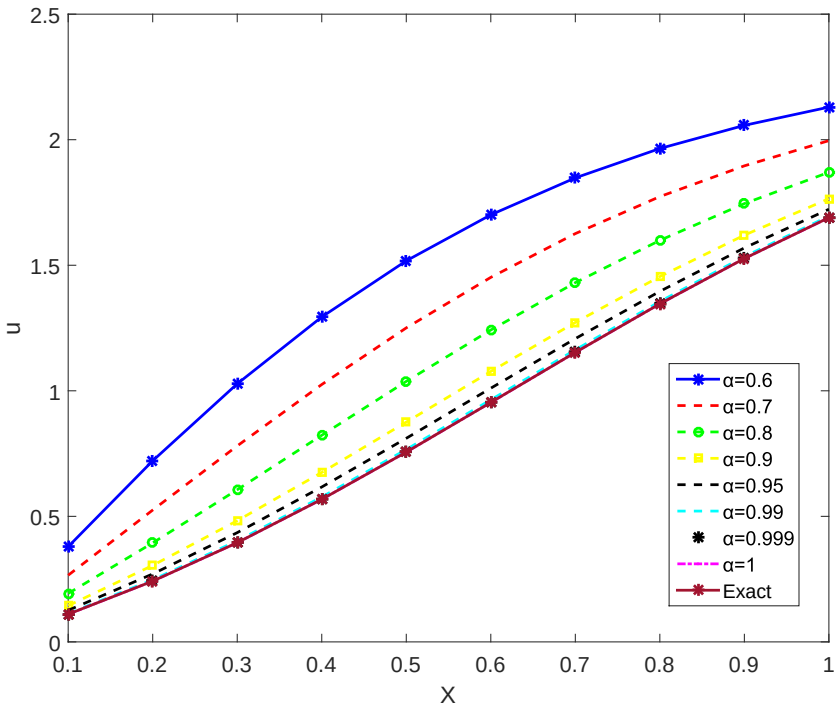


Figure 4: Comparison between exact solution of Example 2 for $\alpha = 1$ and numerical solutions for $\alpha = 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999, 1$ and $T = 1$

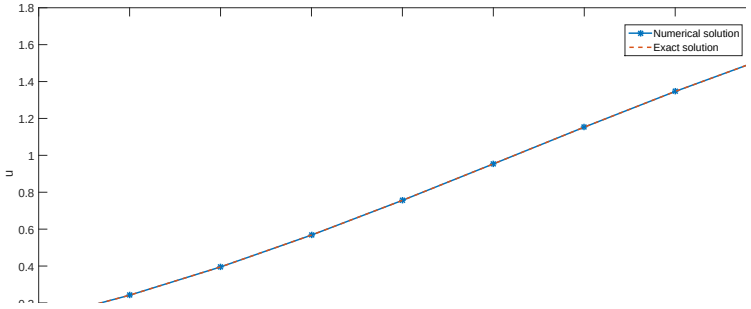


Figure 5: Comparison between the exact solution and numerical solution of Example 2 for $\alpha = 1$, $T = 1$, and $N = 10$

Table 8: Comparison of the numerical solutions of the equation in Example 2 with $\alpha = 1$ and $T = 1$

x	SJOM [2]	MHPM [2]	PSO [2]	CW [2]	FVI [2]	PVI [2]	LWM [2]	Our Method	Exact
0.6	1.007291	1.370240	1.296320	1.349150	1.331462	1.873658	1.296302	0.953552	0.953567
0.7	1.253674	1.367499	1.416139	1.481449	1.497600	2.112944	1.416311	1.152926	1.152950
0.8	1.467499	1.794879	1.506936	1.599235	1.630234	2.260134	1.506913	1.346363	1.346365
0.9	1.629901	1.962239	1.569252	1.705303	1.724439	2.339134	1.569221	1.526897	1.526913
1.0	1.787222	2.087384	1.605580	1.801763	1.776542	2.379356	1.605571	1.689487	1.689500

Table 9: Comparison between the exact solution and numerical solutions of Example 2 for $\alpha = 1$, $T = 8$, and $N = 80$

x	Numerical solution	Exact solution	Error
0.8	1.346362994	1.346363655	6.6128045×10^{-7}
1.6	2.246290755	2.246285959	4.7957279×10^{-6}
2.4	2.395782816	2.395756424	2.6391922×10^{-5}
3.2	2.412338083	2.412281528	5.6554231×10^{-5}
4.0	2.414131848	2.414012382	1.1946588×10^{-4}
4.8	2.414445422	2.414192625	2.527976×10^{-4}
5.6	2.414746423	2.414211383	5.3504015×10^{-4}
6.4	2.415345681	2.414213335	1.1323455×10^{-3}
7.2	2.416609669	2.414213538	2.3961302×10^{-3}
8.0	2.418416749	2.414213559	4.2031900×10^{-3}

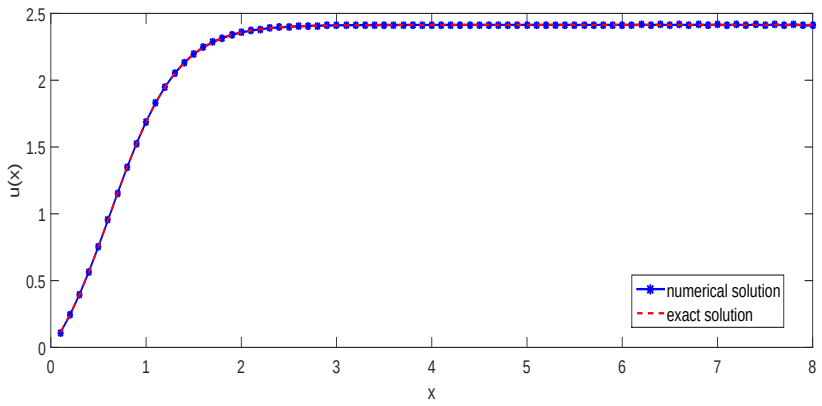


Figure 6: Comparison between the exact solution and numerical solutions of Example 2 for $\alpha = 1$, $T = 8$, and $N = 80$

6 Conclusions

This paper proposed a high-order compact finite difference method for the Riccati problem. The convergence analysis has been discussed. The numer-

ical results presented in Tables 1–9 showed that the method is effective and that the numerical experiment is very consistent with our theoretical analysis results.

References

1. Abbasbandy, S. *Homotopy perturbation method for quadratic Riccati differential equation and comparison with Adomian's decomposition method,* Appl. Math. Comput. 172(1), (2006) 91–102.
2. Agheli, B. *Approximate solution for solving fractional Riccati differential equations via trigonometric basic functions,* Trans. A. Razmadze Math. Inst. 172, (2018) 299–308.
3. Aminikhah, H., Sheikhani, A.H.R. and Rezazadeh, H. *Approximate analytical solutions of distributed order fractional Riccati differential equation,* Ain Shams Eng. J. 9 (4) (2018) 581–588.
4. Anderson, B.D.O. and Moore, J.B. *Optimal filtering,* Englewood, Cliffs. 1979.
5. Anderson, B.D. and Moore, J.B. *Optimal control: Linear quadratic methods,* Prentice-Hall, New Jersey, 2007.
6. Azin, H., Mohammadi, F. and Tenreiro Machado, J.A. *A piecewise spectral-collocation method for solving fractional Riccati differential equation in large domains,* Comp. Appl. Math. 38(3) (2019), Paper No. 96, 13 pp.
7. Biazar, J. and Eslami, M. *Differential transform method for quadratic Riccati differential equation,* Int. J. Nonlinear Sci. 9(4), (2010) 444–447.
8. Bota, C. and Căruntu, B. *Analytical approximate solutions for quadratic Riccati differential equation of fractional order using the Polynomial Least Squares Method,* Chaos Solitons Fractals, 102 (2017) 339–345.
9. Boyle, P.P., Tian, W. and Guan, F. *The Riccati equation in mathematical finance,* J. Symbolic Comput. 33(3), (2002) 343–355.
10. Chen, C.M., Liu, F., Turner, I. and Anh, V. *A Fourier method for the fractional diffusion equation describing sub-diffusion,* J. Comput. Phys. 227, (2007) 886–897.
11. Chen, C.M., Liu, F., Turner, I. and Anh, V. *Numerical methods with fourth-order spatial accuracy for variable-order nonlinear Stokes' first problem for a heated generalized second grade fluid,* Comput. Math. Appl. 62, (2011) 971–986.

12. Chen, S., Liu, F., Zhuang, P. and Anh, V. *Finite difference approximations for the fractional Fokker-Planck equation,* Appl. Math. Model. 33, (2009) 256–273.

13. Cui, M. *Compact finite difference method for the fractional diffusion equation,* J. Comput. Phys. 228, (2009) 7792–7804.

14. Du, R., Cao, W.R. and Sun, Z.Z. *A compact difference scheme for the fractional diffusion-wave equation,* Appl. Math. Model. 34 (2010) 2998–3007.

15. Einicke, G.A., White, L.B. and Bitmead, R.R. *The use of fake algebraic Riccati equations for co-channel demodulation,* IEEE Trans. Signal Process. 51(9), (2003) 2288–2293.

16. Esmaeili, S. and Shamsi, M. *A pseudo-spectral scheme for the approximate solution of a family of fractional differential equations,* Commun. Nonlinear Sci. Numer. Simul. 16, (2011) 3646–3654.

17. Gao, G. and Sun, Z.Z. *A compact finite difference scheme for the fractional sub-diffusion equations,* J. Comput. Phys. 230, (2011) 586–595.

18. Geng, F. *A modified variational iteration method for solving Riccati differential equations,* Comput. Math. Appl. 60(7), (2010) 1868–1872.

19. Gerber, M., Hasselblatt, B. and Keesing, D. *The Riccati equation: pinching of forcing and solutions,* Experiment. Math. 12(2), (2003) 129–134.

20. Langlands, T.A.M. and Henry, B.I. *The accuracy and stability of an implicit solution method for the fractional diffusion equation,* J. Comput. Phys. 205, (2005) 719–736.

21. Lasiecka, I. and Triggiani, R. *Differential and algebraic Riccati equations with application to boundary/point control problems: continuous theory and approximation theory,* Lecture Notes in Control and Information Sciences, 164. Springer-Verlag, Berlin, 1991.

22. Liu, F., Anh, V. and Turner, I. *Numerical solution of the space fractional Fokker-Planck equation,* J. Comput. Appl. Math. 166, (2004) 209–219.

23. Liu, F., Yang, C. and Burrage, K. *Numerical method and analytical technique of the modified anomalous subdiffusion equation with a nonlinear source term,* J. Comput. Appl. Math. 231, (2009) 160–176.

24. Liu, F., Zhuang, P., Anh, V., Turner, I. and Burrage, K.*Stability and convergence of the difference methods for the space-time fractional advection-diffusion equation,* Appl. Math. Comput. 191, (2007) 12–20.

25. Maleknejad, K. and Torkzadeh, L. *Hybrid functions approach for the fractional Riccati differential equation,* Filomat 30(9) (2016) 2453–2463.

26. Miller, K.S. and Ross, B. *An introduction to the fractional calculus and fractional differential equations.,* A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1993.

27. Neamaty, A., Agheli, B. and Darzi, R. *The shifted Jacobi polynomial integral operational matrix for solving Riccati differential equation of fractional order,* Appl. Appl. Math. 10(2) (2015) 878–892.

28. Ntogramatzidis, L. and Ferrante, A. *On the solution of the Riccati differential equation arising from the LQ optimal control problem,* Systems. Control. Lett. 59(2), (2010) 114–121.

29. Odibat, Z.M. *Computational algorithms for computing the fractional derivatives of functions,* Math. Comput. Simul. 79 , (2009) 2013–2020.

30. Odibat, Z. *A Riccati equation approach and travelling wave solutions for nonlinear evolution equations,* Int. J. Appl. Comput. Math. 3(1), (2017) 1–13.

31. Podlubny, I. *Fractional differential equations*, Academic Press, San Diego, 1999.

32. Reid, W.T. *Riccati differential equations,* Mathematics in Science and Engineering, Vol. 86. Academic Press, New York-London, 1972.

33. Saadatmandi, A. and Dehghan, M. *A new operational matrix for solving fractional-order differential equations,* Comput. Math. Appl. (2010) 59, 1326–1336.

34. Saadatmandi, A., Dehghan, M. and Azizi, M.R. *The sinc-Legendre collocation method for a class of fractional convection-diffusion equation with variable coefficients,* Commun. Nonlinear Sci. Numer. Simul. 17, (2012) 4125–4136.

35. Sun, Z.Z. and Wu, X.N.*A fully discrete difference scheme for a diffusion-wave system,* Appl. Numer. Math. 56, (2006) 193–209.

36. Tadjeran, C., Meerschaert, M.M. and Scheffler, H.P. *A second-order accurate numerical approximation for the fractional diffusion equation,* J. Comput. Phys. 213, (2006) 205–213.

37. Tan, Y. and Abbasbandy, S. *Homotopy analysis method for quadratic Riccati differential equation,* Commun. Nonlinear Sci. Numer. Simul. 13(3), (2008) 539–546.

38. Wess, W.*The fractional diffusion equation,* J. Math. Phys. 27, (1996) 2782–2785.

39. Yuste, S.B. *Weighted average finite difference methods for fractional diffusion equations,* J. Comput. Phys. 216, (2006) 264–274.

40. Zhang, P.G. and J. P. Wang, *A predictor–corrector compact finite difference scheme for Burgers' equation,* Appl. Math. Comput. 219(3), (2012) 892–898.

41. Zhuang, P., Liu, F., Anh, V. and Turner, I. *New solution and analytical techniques of the implicit numerical methods for the anomalous sub-diffusion equation,* SIAM J. Numer. Anal. 46, (2008) 1079–1095.

How to cite this article
H. Porki, M. Arabameri and R. Gharechahi . *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 585-606. doi: 10.22067/ijnao.2022.76489.1129.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 607–628
DOI:10.22067/IJNAO.2022.77142.1154
https://ijnao.um.ac.ir/

How to cite this article
Research Article

A numerical approximation for the solution of a time-fractional telegraph equation based on the Crank–Nicolson method

H. Hajinezhad* , A.R. Soheili

Abstract

In this paper, a two-dimensional time-fractional telegraph equation is considered with derivative in the sense of Caputo and $1 < \beta < 2$. The aim of this work is to extend the Crank–Nicolson method for this time-fractional telegraph equation. The stability and convergence of the numerical method are investigated. Also, the accuracy and efficiency of the proposed method are demonstrated by numerical experiments.

AMS subject classifications (2020): 65M06; 65M12; 35R11.

Keywords: Time-Fractional Telegraph Equation; Crank–Nicolson Method; Stability; Convergence.

1 Introduction

Fractional calculus can be used to model many complex problems. It has been used in many fields of science, engineering, and finance [1, 4, 18, 25, 26]; this fact is the main source of inspiration for most of the recent studies

* Corresponding author

Received 13 June 2022; revised 8 August 2022; accepted 3 September 2022

Haniye Hajinezhad
Department of Mathematics, Payame Noor University, Tehran, Iran. e-mail: H.Hajinezhad@pnu.ac.ir

Ali R. Soheili
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: soheili@um.ac.ir

conducted on fractional calculus. The classical telegraph equation is used in random walk theory [2]. The time-fractional telegraph equation (TFTE) models the neutron transport process in the core of a nuclear reactor [27, 28].

In recent decades, the fractional telegraph equation has been solved by many researchers. Here, we briefly describe some of the studies that have been conducted in this field of research. Chen, Liu, and Anh [5] proposed the analytic solution of the TFTE using the separating variables method. The homotopy analysis method was used for the TFTE by Das et al. [6]. Yildirim [31] applied the homotopy perturbation method to solve space- and time-fractional telegraph equations. Momani [17] used Adomian decomposition methods to obtain the analytic and approximate solutions of space- and time-fractional telegraph equations. Biazar, Ebrahimi, and Ayati [3] proposed the variational iteration method to solve the fractional telegraph equation. Jiang and Lin [11] presented the exact solution of the TFTE using the reproducing kernel theorem. Nikan, Avazzadeh, and Machado [19] used a mesh-free spectral approach based on LRBF-FD to solve the TFTE with the fractional derivative described in the sense of Caputo. A radial basis function collocation method was used for solving the nonlinear TFTE by Sepehrian and Shamohammadi [22]. Hosseini et al. [9, 10] considered the meshless local radial point interpolation method, and Mohebbi, Abbaszadeh, and Dehghan [16] used the radial basis function technique for the TFTE. Shivanian applied spectral meshless radial point interpolation methods in [23], and the meshless local Petrov–Galerkin scheme was used in [24] to approximate the TFTE.

Many researchers have studied the fractional telegraph equation by using the finite difference method. Liang, Yao, and Wang [15] considered the TFTE by using a fast, high-order difference scheme. The finite difference method was used to solve the linear TFTE by Li and Cao [14]. Wang and Mei [29] considered the TFTE using a Legendre spectral Galerkin method in space and the generalized finite difference scheme in time. For a time-space-fractional telegraph equation, Zhao and Li [32] used a finite difference method in time and a Galerkin finite element method in space. A numerical method for the TFTE was proposed by Wei, Liu, and Sun [30], in which they discretized this equation with a new finite difference scheme in time and a local discontinuous Galerkin (LDG) method in space.

In this work, we find an approximate solution to the following TFTE [13]:

$$\frac{\partial^{\beta} u(x,y,t)}{\partial t^{\beta}} + \frac{\partial^{\beta-1} u(x,y,t)}{\partial t^{\beta-1}} + u(x,y,t) = \Delta u(x,y,t) + f(x,y,t), \quad (x,y) \in \Omega \subset \mathbb{R}^2, 0 \le t \le T, \tag{1}$$

with initial and boundary conditions

$$u(x,y,0) = \varphi(x,y), \qquad (x,y) \in \overline{\Omega} = \Omega \cup \partial\Omega, \tag{2}$$

$$\frac{\partial u(x,y,0)}{\partial t} = \psi(x,y), \qquad (x,y) \in \overline{\Omega} = \Omega \cup \partial\Omega, \tag{3}$$

$$u(x,y,t) = h(x,y,t), \qquad (x,y) \in \partial\Omega, t > 0, \tag{4}$$

where $1 < \beta < 2, \Delta$ is the Laplace operator, $\partial\Omega$ is the boundary of Ω, $f(x,y), \varphi(x,y), \psi(x,y)$, and $h(x,y,t)$ are continuous functions, $u(x,y,t) \in C^2(\bar{\Omega} \times [0,T])$ is an unknown function, and the fractional derivatives are defined in the sense of Caputo, as follows:

$$\frac{\partial^{\beta-1}u(x,y,t)}{\partial t^{\beta-1}} = \frac{1}{\Gamma(2-\beta)} \int_0^t \frac{\partial u(x,y,s)}{\partial s}(t-s)^{1-\beta} ds, \qquad 1 < \beta < 2, \tag{5}$$

$$\frac{\partial^{\beta}u(x,y,t)}{\partial t^{\beta}} = \frac{1}{\Gamma(2-\beta)} \int_0^t \frac{\partial^2 u(x,y,s)}{\partial s^2}(t-s)^{1-\beta} ds, \qquad 1 < \beta < 2. \tag{6}$$

The Crank–Nicolson difference scheme can be used easily for space-fractional equations, but some manipulations are needed for time-fractional equations [12]. In [19, 13], a semi-discrete scheme based on the Crank–Nicolson method was used to discretize the time-fractional equation. In this work, the discretization of time-fractional derivatives is similar to [12]. The general idea for proving stability and convergence is taken from [19], but our approach differs from that in the details.

The remainder of this paper is organized as follows. In Section 2, the discretization of (1) is described. The stability and the convergence of the proposed method are proved in Sections 3 and 4, respectively. Section 5 is devoted to the numerical tests. Finally, the conclusion is given in Section 6.

2 Discretization of the problem

In this section, we explain the discretization of (1) by using the Crank–Nicolson difference scheme, such that the proposed difference schemes are uniquely solvable.

Consider Δx and Δy as the grid sizes in space for the finite difference scheme, where $\{(x_i, y_i), x_i = i\Delta x, \ \ y_j = j\Delta y, 0 \le i \le I, 0 \le j \le J; I, J \in \mathbb{R}\}$ covers $\overline{\Omega}$. Also, N is a positive integer, and the grid size in time for the finite difference scheme is $\Delta t = \frac{T}{N}$. Assume that $u_{i,j}^n$ is the value of $u(x_i, y_j, t_n)$.

The following lemma provides suitable tools for the discretization of (1).

Lemma 1. If $g(t) \in C^2[0,T]$ and $1 < \beta < 2$, then

(a)

$$\begin{aligned}\int_{t_{k-1}}^{t_k} & g'(s)(t_{n-\frac{1}{2}} - s)^{1-\beta} ds \\ &= \frac{(\Delta t)^{1-\beta}}{(2-\beta)} \left[(n-k+\frac{1}{2})^{2-\beta} - (n-k-\frac{1}{2})^{2-\beta} \right] [g(t_k) - g(t_{k-1})] \\ &\quad + O(\Delta t)^{3-\beta}, \qquad k = 1, 2, \ldots, N-1.\end{aligned}$$

(b)

$$\int_{t_{n-1}}^{t_{n-\frac{1}{2}}} g'(s)(t_{n-\frac{1}{2}}-s)^{1-\beta}ds$$
$$= \frac{(\Delta t)^{1-\beta}}{(2-\beta)2^{2-\beta}}\left[g\left(t_n\right)-g\left(t_{n-1}\right)\right]+O(\quad \Delta t)^{3-\beta}, \qquad n\in\mathbb{N}.$$

Proof. The Taylor expansion allows us to write

$$g'(s)=\frac{g\left(t_k\right)-g\left(t_{k-1}\right)}{\Delta t}-\frac{1}{2\Delta t}[(t_k-s)^2 g''\left(\eta_1\right)-\left(t_{k-1}-s\right)^2 g''\left(\eta_2\right)],$$
$$\eta_1\in\left(s,t_k\right),\eta_2\in\left(t_{k-1},s\right).$$

It is easy to show that

$$\int_{t_{k-1}}^{t_k}(t_k-s)^2(t_{n-\frac{1}{2}}-s)^{1-\beta}ds=\omega_1(\Delta t)^{4-\beta}, \qquad \omega_1\in\mathbb{R},$$
$$\int_{t_{k-1}}^{t_k}(t_{k-1}-s)^2(t_{n-\frac{1}{2}}-s)^{1-\beta}ds=\omega_2(\Delta t)^{4-\beta}, \qquad \omega_2\in\mathbb{R}.$$

Thus,

$$\begin{aligned}
&\int_{t_{k-1}}^{t_k} g'(s)(t_{n-\frac{1}{2}}-s)^{1-\beta}ds\\
=&\frac{g\left(t_k\right)-g(t_{k-1})}{\Delta t}\int_{t_{k-1}}^{t_k}(t_{n-\frac{1}{2}}-s)^{1-\beta}ds\\
&-\frac{1}{2\Delta t}g''(\eta_1)\int_{t_{k-1}}^{t_k}\left(t_k-s\right)^2\left(t_{n-\frac{1}{2}}-s\right)^{1-\beta}ds\\
&+\frac{1}{2\Delta t}g''(\eta_2)\int_{t_{k-1}}^{t_k}\left(t_{k-1}-s\right)^2\left(t_{n-\frac{1}{2}}-s\right)^{1-\beta}ds\\
=&\frac{g\left(t_k\right)-g(t_{k-1})}{\Delta t}\times\frac{(\Delta t)^{2-\beta}}{2-\beta}\left[(n-k+\frac{1}{2})^{2-\beta}-(n-k-\frac{1}{2})^{2-\beta}\right]\\
&+\omega(\Delta t)^{3-\beta}, \qquad \omega\in\mathbb{R}.
\end{aligned}$$

This completes the proof of part (a). Part (b) can be proved in the same way. □

By defining $b_s=\left(s+\frac{1}{2}\right)^{2-\beta}-\left(s-\frac{1}{2}\right)^{2-\beta}, \quad s=1,2,\ldots, \quad 1<\beta<2,$ it is easy to show that

$$\sum_{k=1}^{n-1}\left(u_{i,j}^{k}-u_{i,j}^{k-1}\right)\left((n-k+\frac{1}{2})^{2-\beta}-(n-k-\frac{1}{2})^{2-\beta}\right)+\frac{1}{2^{2-\beta}}\left(u_{i,j}^{n}-u_{i,j}^{n-1}\right)$$
$$=-\left[b_{n-1}u_{i,j}^{0}+\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)u_{i,j}^{k}+(\frac{1}{2^{2-\beta}}-b_1)u_{i,j}^{n-1}\right]+\frac{1}{2^{2-\beta}}u_{i,j}^{n}. \tag{7}$$

By using part (b) of Lemma 3, the discretization of (5) at the grid point (x_i, y_j) and the time step $\left(1-\frac{1}{2}\right)$ is as follows:

$$\begin{aligned}\frac{\partial^{\beta-1}u(x,y,t)}{\partial t^{\beta-1}}|_{i,j}^{1-\frac{1}{2}} &= \frac{1}{\Gamma(2-\beta)}\int_{t_0}^{t_{1-\frac{1}{2}}}\frac{\partial u(x_i,y_j,s)}{\partial s}(t_{1-\frac{1}{2}}-s)^{1-\beta}ds\\ &= \frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left[u_{i,j}^{1}-u_{i,j}^{0}\right]+O(\Delta t)^{3-\beta}.\end{aligned} \tag{8}$$

By using Lemma 3 and relation (7), the discretization of (5) at the grid point (x_i, y_j) and the time step $\left(n-\frac{1}{2}\right)$ is as follows:

$$\begin{aligned}\frac{\partial^{\beta-1}u(x,y,t)}{\partial t^{\beta-1}}\,|_{i,j}^{n-\frac{1}{2}} =&\frac{1}{\Gamma(2-\beta)}\sum_{k=1}^{n-1}\int_{t_{k-1}}^{t_k}\frac{\partial u(x_i,y_j,s)}{\partial s}(t_{n-\frac{1}{2}}-s)^{1-\beta}ds\\ &+\frac{1}{\Gamma(2-\beta)}\int_{t_{n-1}}^{t_{n-\frac{1}{2}}}\frac{\partial u\left(x_i,y_j,s\right)}{\partial s}(t_{n-\frac{1}{2}}-s)^{1-\beta}ds\\ =&\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}u_{i,j}^{0}-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})u_{i,j}^{k}\right.\\ &\left.-(\frac{1}{2^{2-\beta}}-b_1)u_{i,j}^{n-1}+\frac{1}{2^{2-\beta}}u_{i,j}^{n}\right\}\\ &+O(\Delta t)^{3-\beta},\qquad n\geq 2, 1\leq i\leq I-1, 1\leq j\leq J-1.\end{aligned} \tag{9}$$

In addition, similar to (8) and (9), and by using the relation

$$\left.\frac{\partial u}{\partial t}\right|_{i,j}^{k}=\frac{u_{i,j}^{k}-u_{i,j}^{k-1}}{\Delta t}+\frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}\left(x_i,y_j,\eta_1\right),\qquad k\geq 1,\eta_1\in\left(t_{k-1},t_k\right), \tag{10}$$

we obtain

$$\left.\frac{\partial^{\beta}u(x,y,t)}{\partial t^{\beta}}\right|_{i,j}^{1-\frac{1}{2}}=\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left[\frac{u_{i,j}^{1}-u_{i,j}^{0}}{\Delta t}-\left.\frac{\partial u}{\partial t}\right|_{i,j}^{0}\right]+O(\Delta t)^{2-\beta}, \tag{11}$$

$$\frac{\partial^\beta u(x,y,t)}{\partial t^\beta}|_{i,j}^{n-\frac{1}{2}}$$
$$= \frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}\frac{\partial u}{\partial t}\Big|_{i,j}^{0} - \sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\frac{u_{i,j}^{k}-u_{i,j}^{k-1}}{\Delta t}\right.$$
$$\left. -(\frac{1}{2^{2-\beta}}-b_1)\frac{u_{i,j}^{n-1}-u_{i,j}^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{u_{i,j}^{n}-u_{i,j}^{n-1}}{\Delta t}\right\}$$
$$+O(\Delta t)^{2-\beta}, \qquad n\geq 2, 1\leq i\leq I-1, 1\leq j\leq J-1. \tag{12}$$

Having the Taylor expansion in mind, we can write

$$u(x_i,y_j,t_{n-\frac{1}{2}}) = \frac{u_{i,j}^{n-1}+u_{i,j}^{n}}{2}+O(\Delta t)^2, \qquad n\geq 1, 1\leq i\leq I-1, 1\leq j\leq J-1, \tag{13}$$

$$\Delta u(x_i,y_j,t_{n-\frac{1}{2}}) = \frac{\Delta u_{i,j}^{n-1}+\Delta u_{i,j}^{n}}{2}+O(\Delta t)^2$$
$$= \frac{1}{2}\left\{\frac{u_{i+1,j}^{n-1}-2u_{i,j}^{n-1}+u_{i-1,j}^{n-1}}{(\Delta x)^2}+\frac{u_{i,j+1}^{n-1}-2u_{i,j}^{n-1}+u_{i,j-1}^{n-1}}{(\Delta y)^2}\right.$$
$$\left.+\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{(\Delta x)^2}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{(\Delta y)^2}\right\}$$
$$+O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^2, \qquad n\geq 1, 1\leq i\leq I-1, 1\leq j\leq J-1. \tag{14}$$

Using the finite difference schemes (11), (8), (13), and (14), the discretization of (1) at the grid point (x_i, y_j) and the time step $\left(1-\frac{1}{2}\right)$ is as follows:

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left[\frac{u_{i,j}^{1}-u_{i,j}^{0}}{\Delta t}-\frac{\partial u}{\partial t}\Big|_{i,j}^{0}\right]+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left(u_{i,j}^{1}-u_{i,j}^{0}\right)$$
$$+\frac{1}{2}\left(u_{i,j}^{1}+u_{i,j}^{0}\right) = \frac{1}{2}\left\{\frac{u_{i+1,j}^{0}-2u_{i,j}^{0}+u_{i-1,j}^{0}}{(\Delta x)^2}+\frac{u_{i,j+1}^{0}-2u_{i,j}^{0}+u_{i,j-1}^{0}}{(\Delta y)^2}\right.$$
$$\left.+\frac{u_{i+1,j}^{1}-2u_{i,j}^{1}+u_{i-1,j}^{1}}{(\Delta x)^2}+\frac{u_{i,j+1}^{1}-2u_{i,j}^{1}+u_{i,j-1}^{1}}{(\Delta y)^2}\right\}$$
$$+f_{i,j}^{1-\frac{1}{2}}+O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}, \qquad 1\leq i\leq I-1, 1\leq j\leq J-1. \tag{15}$$

Using the finite difference schemes (12), (9), (13), and (14), the discretization of (1) at the grid point (x_i, y_j) and the time step $\left(n-\frac{1}{2}\right)$ can be written as follows:

$$\begin{aligned}
&\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}\frac{\partial u}{\partial t}\Big|_{i,j}^{0}-\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\frac{u_{i,j}^{k}-u_{i,j}^{k-1}}{\Delta t}\right.\\
&\left.-\left(\frac{1}{2^{2-\beta}}-b_1\right)\frac{u_{i,j}^{n-1}-u_{i,j}^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{u_{i,j}^{n}-u_{i,j}^{n-1}}{\Delta t}\right\}\\
&+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}u_{i,j}^{0}-\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)u_{i,j}^{k}\right.\\
&\left.-\left(\frac{1}{2^{2-\beta}}-b_1\right)u_{i,j}^{n-1}+\frac{1}{2^{2-\beta}}u_{i,j}^{n}\right\}+\frac{u_{i,j}^{n-1}+u_{i,j}^{n}}{2} \qquad (16)\\
&\quad=\frac{1}{2}\left\{\frac{u_{i+1,j}^{n-1}-2u_{i,j}^{n-1}+u_{i-1,j}^{n-1}}{(\Delta x)^2}+\frac{u_{i,j+1}^{n-1}-2u_{i,j}^{n-1}+u_{i,j-1}^{n-1}}{(\Delta y)^2}\right.\\
&\quad\left.+\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{(\Delta x)^2}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{(\Delta y)^2}\right\}\\
&\quad+f_{i,j}^{n-\frac{1}{2}}+O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta},\\
&\qquad\qquad n\geq 2, 1\leq i\leq I-1, 1\leq j\leq J-1.
\end{aligned}$$

Finally, rearranging (15) and (16) and neglecting the truncation errors, it is obvious that the coefficient matrix of the unknowns is strictly diagonally dominant and so, it is nonsingular [8]. Therefore, by neglecting the truncation errors in (15) and (16), the unknowns $\left[u_{i,j}^{n}\right]$ $(1\leq i\leq I-1, 1\leq j\leq J-1)$ can be obtained for $n=1$ and $n\geq 2$, respectively. Hence, the proposed Crank–Nicolson scheme is uniquely solvable.

3 Stability

In this section, we study the stability of the proposed Crank–Nicolson scheme for (1) with initial and boundary conditions (2)–(4). To do so, we introduce the following spaces and recall some theorems and lemmas, which will be used hereafter.

$$\begin{aligned}
H^1(\Omega)&=\left\{v\in L^2(\Omega): Dv\in L^2(\Omega)\right\},\\
H_0^1(\Omega)&=\left\{v\in H^1(\Omega): Dv|_{\partial\Omega}=0\right\},\\
H^2(\Omega)&=\left\{v\in L^2(\Omega): D^{\alpha}v\in L^2(\Omega), |\alpha|\leq 2\right\}.
\end{aligned}$$

Theorem 1 (The Cauchy–Schwarz inequality)**.** [21]
If u and v are members of an inner product space Ω with inner product $\langle\cdot,\cdot\rangle$, then

$$|\langle u,v\rangle|=\left|\int_{\Omega}uvdx\right|\leq\|u\|_{L^2}\|v\|_{L^2}.$$

Theorem 2. (Green's theorem) [21]
If Ω is a boundary domain in $\mathbb{R}^n$, then

$$\int_{\Omega} \nabla u.\nabla v \mathrm{dx} = \int_{\partial\Omega} v\frac{\partial u}{\partial n} ds - \int_{\Omega} \Delta u\, v\, dx, \qquad for\ u \in H^2(\Omega), v \in H^1(\Omega).$$

Theorem 3 (The Poincare–Friedrich inequality)**.** [21]
Let Ω be a boundary domain in $\mathbb{R}^n$. Then, there exists a constant $c_p > 0$ such that

$$\|u\|_{L^2}^2 \leq c_p \|\nabla u\|_{L^2}^2, \qquad for\ all\ u \in H_0^1(\Omega).$$

Theorem 4 (The discrete Gronwall theorem)**.** [20]
Assume that k_n is a nonnegative sequence and that the sequence ϕ_n satisfies the following relations:

$$\begin{aligned} &\phi_0 \leq g_0, \\ &\phi_n \leq g_0 + \sum_{s=0}^{n-1} p_s + \sum_{s=0}^{n-1} k_s \phi_s, \qquad n \geqslant 1. \end{aligned}$$

If $g_0 \geq 0$ and $p_n \geq 0$ (for $n \geq 0$), then

$$\phi_n \leq \left(g_0 + \sum_{s=0}^{n-1} p_s\right) \exp\left(\sum_{s=0}^{n-1} k_s\right), \qquad n \geqslant 1.$$

We state some useful relations in Lemmas 2 and 3. These are easy to prove.

Lemma 2. It holds that $\|u\|\|v\| \leq \frac{\gamma^2}{2}\|u\|^2 + \frac{1}{2\gamma^2}\|v\|^2$, for all $u, v \in \Omega$, for all $\gamma \in \mathbb{R}$.

Lemma 3. If $b_s = \left(s+\frac{1}{2}\right)^{2-\beta} - \left(s-\frac{1}{2}\right)^{2-\beta}$ $(s = 1, 2, \ldots, 1 < \beta < 2)$, then $b_n < b_{n-1} < \cdots < b_2 < b_1 < 1$.

Neglecting the truncation errors, equations (15) and (16) can be written as

$$\begin{aligned} &\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)} \times \frac{1}{2^{2-\beta}} \left[\frac{u_{i,j}^1 - u_{i,j}^0}{\Delta t} - \left.\frac{\partial u}{\partial t}\right|_{i,j}^0\right] \\ &+ \frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)} \times \frac{1}{2^{2-\beta}}(u_{i,j}^1 - u_{i,j}^0) + \frac{1}{2}(u_{i,j}^1 + u_{i,j}^0) \\ &\qquad = \frac{1}{2}\left(\Delta u_{i,j}^1 + \Delta u_{i,j}^0\right) + f_{i,j}^{1-\frac{1}{2}}, \quad 1 \leqslant i \leqslant I-1, 1 \leqslant j \leqslant J-1, \end{aligned} \tag{17}$$

and

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}\frac{\partial u}{\partial t}\Big|_{i,j}^{0}-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\frac{u_{i,j}^{k}-u_{i,j}^{k-1}}{\Delta t}\right.$$
$$\left.-(\frac{1}{2^{2-\beta}}-b_1)\frac{u_{i,j}^{n-1}-u_{i,j}^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{u_{i,j}^{n}-u_{i,j}^{n-1}}{\Delta t}\right\}$$
$$+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}u_{i,j}^{0}-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,u_{i,j}^{k}\right.$$
$$\left.-(\frac{1}{2^{2-\beta}}-b_1)u_{i,j}^{n-1}+\frac{1}{2^{2-\beta}}u_{i,j}^{n}\right\}+\frac{u_{i,j}^{n-1}+u_{i,j}^{n}}{2}$$
$$=\frac{\Delta u_{i,j}^{n-1}+\Delta u_{i,j}^{n}}{2}+f_{i,j}^{n-\frac{1}{2}},\qquad n\geqslant 2, 1\leqslant i\leqslant I-1, 1\leqslant j\leqslant J-1, \tag{18}$$

respectively. Let $\widetilde{u}_{i,j}^{n}$ $(1\leq i\leq I-1, 1\leq j\leq J-1, n=1,2,\ldots)$ be the approximate solution of (17) and (18) with respect to the round-off error, and let $u_{i,j}^{n}$ $(1\leq i\leq I-1, 1\leq j\leq J-1, n=1,2,\ldots)$ be the exact solution of (17) and (18). Define

$$e_{i,j}^{n}=u_{i,j}^{n}-\widetilde{u}_{i,j}^{n}\quad(0\leq i\leq I,\quad 0\leq j\leq J,\quad \textit{and}\quad n=0,1,\ldots).$$

By considering e^n instead of $e_{i,j}^{n}$, we obtain the following round-off error equations:

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left\{\left[\frac{e^1-e^0}{\Delta t}-\delta e^0\right]+(e^1-e^0)\right\}+\frac{1}{2}(e^1+e^0)=\frac{1}{2}(\Delta e^1+\Delta e^0), \tag{19}$$

$$\begin{aligned}&\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}\delta e^0-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,\frac{e^k-e^{k-1}}{\Delta t}\right.\\&\left.-(\frac{1}{2^{2-\beta}}-b_1)\frac{e^{n-1}-e^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{e^n-e^{n-1}}{\Delta t}\right\}\\&+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}e^0-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,e^k-(\frac{1}{2^{2-\beta}}-b_1)e^{n-1}\right.\\&\left.+\frac{1}{2^{2-\beta}}e^n\right\}+\frac{e^{n-1}+e^n}{2}=\frac{\Delta e^{n-1}+\Delta e^n}{2},\qquad n\geqslant 2,\end{aligned} \tag{20}$$

where $\delta e^0=\frac{\partial u}{\partial t}\big|_{i,j}^{0}-\frac{\partial \widetilde{u}}{\partial t}\big|_{i,j}^{0}$. Now, we are ready to present the following theorem.

Theorem 5. If $e^k\in H_0^1(\Omega)$, then the solutions of the finite difference approaches (17) and (18) are unconditionally stable.

Proof. Let $\alpha=\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}$. If we multiply (19) by $\left(e^1-e^0\right)$, then we obtain

$$\frac{\alpha}{2^{2-\beta}(\Delta t)}\left\langle e^1-e^0,e^1-e^0\right\rangle+\frac{\alpha}{2^{2-\beta}}\left\langle e^1-e^0,e^1-e^0\right\rangle+\frac{1}{2}\left\langle e^1+e^0,e^1-e^0\right\rangle$$
$$-\frac{1}{2}\left\langle \Delta e^1+\Delta e^0,e^1-e^0\right\rangle=\frac{\alpha}{2^{2-\beta}}\left\langle \delta e^0,e^1-e^0\right\rangle. \tag{21}$$

Applying Theorem 2 (Green's theorem) to $\left\langle \Delta e^1+\Delta e^0,e^1-e^0\right\rangle$ in the left side of (51) and applying Theorem 1 (the Cauchy–Schwarz inequality) and Lemma 2 to the right side of (51), we can write

$$\frac{\alpha}{2^{2-\beta}(\Delta t)}\left\|e^1-e^0\right\|^2+\frac{\alpha}{2^{2-\beta}}\left\|e^1-e^0\right\|^2+\frac{1}{2}(\left\|e^1\right\|^2-\left\|e^0\right\|^2)$$
$$+\frac{1}{2}(\left\|\nabla e^1\right\|^2-\left\|\nabla e^0\right\|^2)\leq\frac{\alpha}{2^{2-\beta}}\left\{\frac{\left\|\delta e^0\right\|^2}{2}+\frac{\left\|e^1-e^0\right\|^2}{2}\right\}.$$

Therefore,

$$\left\|\nabla e^1\right\|^2\leq\left\|e^0\right\|^2+\left\|\nabla e^0\right\|^2+\frac{\alpha}{2^{2-\beta}}\left\|\delta e^0\right\|^2,$$

and by applying Theorem 3 (the Poincaré–Friedrich inequality), we find a constant $c_p>0$ such that

$$\left\|\nabla e^1\right\|^2\leq(c_p+1)\left\|\nabla e^0\right\|^2+\frac{\alpha c_p}{2^{2-\beta}}\left\|\nabla\delta e^0\right\|^2. \tag{22}$$

If we multiply (20) by $\left(e^n-e^{n-1}\right)$, then we find

$$\begin{aligned}
&\frac{\alpha}{2^{2-\beta}(\Delta t)}\left\langle e^n-e^{n-1},e^n-e^{n-1}\right\rangle+\frac{\alpha}{2^{2-\beta}}\left\langle e^n,e^n-e^{n-1}\right\rangle\\
&+\frac{1}{2}\left\langle e^n+e^{n-1},e^n-e^{n-1}\right\rangle-\frac{1}{2}\left\langle \Delta e^n+\Delta e^{n-1},e^n-e^{n-1}\right\rangle\\
&\quad=\alpha b_{n-1}\left\langle \delta e^0,e^n-e^{n-1}\right\rangle\\
&\qquad+\alpha\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\langle\frac{e^k-e^{k-1}}{\Delta t},e^n-e^{n-1}\rangle\\
&\qquad+\alpha(\frac{1}{2^{2-\beta}}-b_1)\langle\frac{e^{n-1}-e^{n-2}}{\Delta t},e^n-e^{n-1}\rangle+\alpha b_{n-1}\left\langle e^0,e^n-e^{n-1}\right\rangle\\
&\qquad+\alpha\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\left\langle e^k,e^n-e^{n-1}\right\rangle\\
&\qquad+\alpha(\frac{1}{2^{2-\beta}}-b_1)\left\langle e^{n-1},e^n-e^{n-1}\right\rangle.
\end{aligned} \tag{23}$$

Applying Theorem 2 (Green's theorem) to $\left\langle \Delta e^n+\Delta e^{n-1},e^n-e^{n-1}\right\rangle$ in the left side of (23) and applying Theorem 1 (the Cauchy–Schwarz inequality) and Lemma 2 to the right side of (23), we obtain

$$\begin{aligned}
&\frac{\alpha}{2^{2-\beta}(\Delta t)}\|e^n-e^{n-1}\|^2+(\frac{\alpha}{2^{2-\beta}}\|e^n\|^2-\frac{\alpha}{2^{2-\beta}}\left\langle e^n,e^{n-1}\right\rangle)\\
&+\frac{1}{2}(\|e^n\|^2-\left\|e^{n-1}\right\|^2)+\frac{1}{2}(\|\nabla e^n\|^2-\left\|\nabla e^{n-1}\right\|^2)\\
&\quad\le \alpha b_{n-1}(\frac{\gamma^2}{2}\left\|\delta e^0\right\|^2+\frac{\left\|e^n-e^{n-1}\right\|^2}{2\gamma^2})\\
&\qquad+\alpha\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,(\frac{\gamma^2}{2}\|\frac{e^k-e^{k-1}}{\Delta t}\|^2+\frac{1}{2\gamma^2}\left\|e^n-e^{n-1}\right\|^2)\\
&\qquad+\alpha(\frac{\gamma^2}{2}\|\frac{e^{n-1}-e^{n-2}}{\Delta t}\|^2+\frac{1}{2\gamma^2}\|e^n-e^{n-1}\|^2)\\
&\qquad+\alpha b_{n-1}(\frac{\gamma^2}{2}\left\|e^0\right\|^2+\frac{\left\|e^n-e^{n-1}\right\|^2}{2\gamma^2})\\
&\qquad+\alpha\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,(\frac{\gamma^2}{2}\left\|e^k\right\|^2+\frac{1}{2\gamma^2}\left\|e^n-e^{n-1}\right\|^2)\\
&\qquad+\alpha(\frac{\gamma^2}{2}\|e^{n-1}\|^2+\frac{1}{2\gamma^2}\|e^n-e^{n-1}\|^2),\qquad\qquad \gamma\in\mathbb{R}.
\end{aligned}\tag{24}$$

Furthermore, from Lemma 3, we deduce that

$$\frac{\alpha b_{n-1}}{\gamma^2}+\frac{\alpha}{\gamma^2}\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})+\frac{\alpha}{\gamma^2}\le\frac{2\alpha}{\gamma^2},\quad \gamma\in\mathbb{R}.\tag{25}$$

Having (25) in mind, equation (23) gives

$$\begin{aligned}
&\frac{\alpha}{2^{2-\beta}(\Delta t)}\|e^n-e^{n-1}\|^2+\frac{\alpha}{2^{2-\beta}}\|e^n\|^2+\frac{1}{2}\|e^n\|^2+\frac{1}{2}\|\nabla e^n\|^2\\
&\quad\le\alpha b_{n-1}\frac{\gamma^2}{2}\left\{\left\|\delta e^0\right\|^2+\left\|e^0\right\|^2\right\}+\frac{\alpha\gamma^2}{2}\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\left\|e^k\right\|^2\\
&\qquad+\frac{\alpha\gamma^2}{2(\Delta t)^2}\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\left\|e^k-e^{k-1}\right\|^2+\frac{\alpha\gamma^2}{2(\Delta t)^2}\left\|e^{n-1}-e^{n-2}\right\|^2\\
&\qquad+(\frac{\alpha\gamma^2}{2}+\frac{1}{2})\left\|e^{n-1}\right\|^2+\frac{2\alpha}{\gamma^2}\left\|e^n-e^{n-1}\right\|^2+\frac{\alpha}{2^{2-\beta}}\left\langle e^n,e^{n-1}\right\rangle\\
&\qquad+\frac{1}{2}\left\|\nabla e^{n-1}\right\|^2,\quad \gamma\in\mathbb{R}.
\end{aligned}\tag{26}$$

By using Theorem 1 (the Cauchy–Schwarz inequality) and Lemma 2, we obtain

$$\frac{\alpha}{2^{2-\beta}}\left\langle e^n,e^{n-1}\right\rangle\le\frac{\alpha}{2^{2-\beta}}\left\{\frac{\|e^n\|^2}{2}+\frac{\left\|e^{n-1}\right\|^2}{2}\right\}.\tag{27}$$

Consider the following relations:

$$\begin{aligned}\frac{\alpha\gamma^2}{2(\Delta t)^2}\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\left\|e^k-e^{k-1}\right\|^2+\frac{\alpha\gamma^2}{2(\Delta t)^2}\left\|e^{n-1}-e^{n-2}\right\|^2\\ \leq\frac{\alpha\gamma^2}{(\Delta t)^2}\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\left(\left\|e^k\right\|^2+\left\|e^{k-1}\right\|^2\right)\\ +\frac{\alpha\gamma^2}{(\Delta t)^2}\left(\left\|e^{n-1}\right\|^2+\left\|e^{n-2}\right\|^2\right).\end{aligned}\tag{28}$$

If we use (27)–(28) and assume that $\gamma^2=2^{3-\beta}(\Delta t)$, then relation (26) allows us to write

$$\begin{aligned}\frac{1}{2}\times\frac{\alpha}{2^{2-\beta}}\left\|e^n\right\|^2+\frac{1}{2}\left\|e^n\right\|^2+\frac{1}{2}\left\|\nabla e^n\right\|^2\\ \leq\alpha b_{n-1}\frac{\gamma^2}{2}\left\{\left\|\delta e^0\right\|^2+\left\|e^0\right\|^2\right\}+\frac{\alpha\gamma^2}{2}\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\left\|e^k\right\|^2\\ +\frac{\alpha\gamma^2}{(\Delta t)^2}\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\left(\left\|e^k\right\|^2+\left\|e^{k-1}\right\|^2\right)+\frac{\alpha\gamma^2}{(\Delta t)^2}\left\|e^{n-2}\right\|^2\\ +(\frac{\alpha\gamma^2}{2}+\frac{\alpha\gamma^2}{(\Delta t)^2}+\frac{\alpha}{2\times2^{2-\beta}}+\frac{1}{2})\left\|e^{n-1}\right\|^2+\frac{1}{2}\left\|\nabla e^{n-1}\right\|^2,\quad n\geq2.\end{aligned}\tag{29}$$

By using Theorem 3 (the Poincaré–Friedrich inequality), we find a constant $c_p>0$ such that relation (29) implies

$$\begin{aligned}\frac{1}{2}\left\|\nabla e^n\right\|^2\leq&\frac{\alpha b_{n-1}\gamma^2}{2}c_p\left\|\nabla e^0\right\|^2+\frac{\alpha b_{n-1}\gamma^2}{2}c_p\left\|\nabla\delta e^0\right\|^2\\ &+\frac{\alpha\gamma^2}{2}c_p\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\left\|\nabla e^k\right\|^2\\ &+\frac{\alpha\gamma^2}{(\Delta t)^2}c_p\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})(\left\|\nabla e^k\right\|^2+\left\|\nabla e^{k-1}\right\|^2)\\ &+\frac{\alpha\gamma^2}{(\Delta t)^2}c_p\left\|\nabla e^{n-2}\right\|^2+(\frac{\alpha\gamma^2}{2}+\frac{\alpha\gamma^2}{(\Delta t)^2}\\ &+\frac{\alpha}{2\times2^{2-\beta}}+\frac{1}{2})c_p\left\|\nabla e^{n-1}\right\|^2+\frac{1}{2}\left\|\nabla e^{n-1}\right\|^2,\quad n\geq2.\end{aligned}\tag{30}$$

We may assume without loss of generality that there exist constants $\theta_1,\theta_2\geq0$ such that relations (22) and (30) can be written as

$$\begin{aligned}
&\left\|\nabla e^1\right\|^2 \leq \theta_1 \left\|\nabla e^0\right\|^2 + \theta_2 \left\|\nabla \delta e^0\right\|^2, \\
&\|\nabla e^n\|^2 \leq \left(\theta_1 \left\|\nabla e^0\right\|^2 + \theta_2 \left\|\nabla \delta e^0\right\|^2\right) + \sum_{k=1}^{n-1} \left(c_k \left\|\nabla e^k\right\|^2\right), \qquad (31) \\
&\qquad\qquad n \geq 2, \quad \theta_1, \theta_2 \geq 0, \quad c_k > 0 \ for \quad k = 1, \ldots, n-1.
\end{aligned}$$

By Theorem 4 (the discrete Gronwall theorem), equation (31) yields

$$\|\nabla e^n\|^2 \leq \left(\theta_1 \left\|\nabla e^0\right\|^2 + \theta_2 \left\|\nabla \delta e^0\right\|^2\right) \exp\left(\sum_{k=1}^{n-1} c_k\right), \quad n \geq 1, \theta_1, \theta_2 \geq 0,$$

and according to Theorem 3 (the Poincare–Friedrich inequality), there exists a constant $\widehat{c}_p > 0$ such that

$$\|e^n\|^2 \leq \widehat{c}_p \left(\theta_1 \left\|\nabla e^0\right\|^2 + \theta_2 \left\|\nabla \delta e^0\right\|^2\right) \exp\left(\sum_{k=1}^{n-1} c_k\right), \quad n \geq 1, \theta_1, \theta_2 \geq 0. \qquad (32)$$

By using Lemma 3, it is easy to show that

$$\sum_{k=1}^{n-1} c_k \leq \left(2\alpha\gamma^2 + \frac{8\alpha\gamma^2}{(\Delta t)^2} + \frac{\alpha}{2^{2-\beta}} + 1\right) c_p + 1. \qquad (33)$$

Set $\theta = \left(2\alpha\gamma^2 + \frac{8\alpha\gamma^2}{(\Delta t)^2} + \frac{\alpha}{2^{2-\beta}} + 1\right) c_p + 1$. Then, it follows from relations (32) and (33) that

$$\|e^n\| \leq \sqrt{\widehat{c}_p \left(\theta_1 \left\|\nabla e^0\right\|^2 + \theta_2 \left\|\nabla \delta e^0\right\|^2\right) \exp(\theta)}, \quad n \geq 1, \theta_1, \theta_2, \theta \geq 0, \widehat{c}_p > 0,$$

where θ_1, θ_2, θ, $\widehat{c}_p$ are independent of n. □

4 Convergence

In this section, we study the convergence of the proposed Crank–Nicolson scheme for (1) with initial and boundary conditions (2)–(4).

Let $u_{i,j}^n$ $(1 \leq i \leq I-1, 1 \leq j \leq J-1, n = 1, 2, \ldots)$ be the exact solution of (17) and (18), and let $U_{i,j}^n$ $(1 \leq i \leq I-1, 1 \leq j \leq J-1, n = 1, 2, \ldots)$ be the exact solution of (15) and (16). Define $\xi_{i,j}^n = U_{i,j}^n - u_{i,j}^n$ $(1 \leq i \leq I-1, 1 \leq j \leq J-1, n = 1, 2, \ldots)$. By considering ξ^n instead of $\xi_{i,j}^n$ we obtain

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)} \times \frac{1}{2^{2-\beta}} \frac{\xi^1}{\Delta t} + \frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)} \times \frac{1}{2^{2-\beta}} \xi^1 + \frac{1}{2}\xi^1 = \frac{\Delta \xi^1}{2} + O(\Delta x)^2 + (O(\Delta y)^2 + O(\Delta t)^{2-\beta}), \tag{34}$$

and

$$\begin{aligned} &\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)} \left\{ -\sum_{k=1}^{n-2} (b_{n-k-1} - b_{n-k}) \frac{\xi^k - \xi^{k-1}}{\Delta t} - (\frac{1}{2^{2-\beta}} - b_1) \frac{\xi^{n-1} - \xi^{n-2}}{\Delta t} \right. \\ &\quad \left. + \frac{1}{2^{2-\beta}} \frac{\xi^n - \xi^{n-1}}{\Delta t} \right\} \\ &\quad + \frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)} \left\{ -\sum_{k=1}^{n-2} (b_{n-k-1} - b_{n-k})\, \xi^k - (\frac{1}{2^{2-\beta}} - b_1)\xi^{n-1} + \frac{1}{2^{2-\beta}} \xi^n \right\} \\ &\quad + \frac{\xi^{n-1} + \xi^n}{2} \\ &\quad = \frac{\Delta \xi^{n-1} + \Delta \xi^n}{2} + \left(O(\Delta x)^2 + O(\Delta y)^2 + O(\Delta t)^{2-\beta}\right), \quad n \geq 2. \end{aligned} \tag{35}$$

Now, we are ready to present our next theorem.

Theorem 6. If $\xi^k \in H_0^1(\Omega)$, then the solutions of the finite difference approaches (17) and (18) are unconditionally convergent.

Proof. Let $\alpha = \frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}$. If we multiply (34) by $\left(\xi^1\right)$, then we obtain

$$\begin{aligned} &\frac{\alpha}{2^{2-\beta}(\Delta t)} \left\langle \xi^1, \xi^1 \right\rangle + \frac{\alpha}{2^{2-\beta}} \left\langle \xi^1, \xi^1 \right\rangle + \frac{1}{2} \left\langle \xi^1, \xi^1 \right\rangle - \frac{1}{2} \left\langle \Delta \xi^1, \xi^1 \right\rangle \\ &\quad =< \left(O(\Delta x)^2 + O(\Delta y)^2 + O(\Delta t)^{2-\beta}\right), \xi^1 > 0. \end{aligned} \tag{36}$$

Applying Theorem 2 (Green's theorem) to $\left\langle \Delta \xi^1, \xi^1 \right\rangle$ in the left side of (36) and applying Theorem 1 (the Cauchy–Schwarz inequality) and Lemma 2 to the right side of (36), we find that

$$\frac{\alpha}{2^{2-\beta}(\Delta t)} \left\|\xi^1\right\|^2 + \frac{\alpha}{2^{2-\beta}} \left\|\xi^1\right\|^2 + \frac{\left\|\xi^1\right\|^2}{2} + \frac{1}{2} \left\|\nabla \xi^1\right\|^2 \leq \frac{\left\|O(\Delta x)^2 + O(\Delta y)^2 + O(\Delta t)^{2-\beta}\right\|^2}{2} + \frac{\left\|\xi^1\right\|^2}{2}.$$

Therefore,

$$\|\nabla \xi^1\|^2 \leq \|O(\Delta x)^2 + O(\Delta y)^2 + O(\Delta t)^{2-\beta}\|^2. \tag{37}$$

If we multiply (35) by $\left(\xi^n - \xi^{n-1}\right)$, then we obtain

$$\begin{aligned}
&\frac{\alpha}{2^{2-\beta}(\Delta t)}\left\langle \xi^n-\xi^{n-1},\xi^n-\xi^{n-1}\right\rangle+\frac{\alpha}{2^{2-\beta}}\left\langle \xi^n,\xi^n-\xi^{n-1}\right\rangle\\
&+\frac{1}{2}\left\langle \xi^n+\xi^{n-1},\xi^n-\xi^{n-1}\right\rangle-\frac{1}{2}\left\langle \Delta\xi^n+\Delta\xi^{n-1},\xi^n-\xi^{n-1}\right\rangle\\
&\quad=\alpha\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\langle\frac{\xi^k-\xi^{k-1}}{\Delta t},\xi^n-\xi^{n-1}\rangle\\
&\quad+\alpha(\frac{1}{2^{2-\beta}}-b_1)\langle\frac{\xi^{n-1}-\xi^{n-2}}{\Delta t},\xi^n-\xi^{n-1}\rangle\\
&\quad+\alpha\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\left\langle \xi^k,\xi^n-\xi^{n-1}\right\rangle\\
&\quad+\alpha(\frac{1}{2^{2-\beta}}-b_1)\left\langle \xi^{n-1},\xi^n-\xi^{n-1}\right\rangle\\
&\quad+<\left(O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}\right),\xi^n-\xi^{n-1}\rangle.
\end{aligned}\tag{38}$$

Again, using Theorem 1 (the Cauchy–Schwarz inequality) and Lemma 2, we can write

$$\begin{aligned}
&<(O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}),\xi^n-\xi^{n-1}>\\
&\qquad\leq\|O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}\|^2+\frac{\|\xi^n\|^2}{2}+\frac{\left\|\xi^{n-1}\right\|^2}{2}.
\end{aligned}$$

Simplifying relation (38) (similar to Theorem 5, in which the simplification of (23) resulted in (29)) and using the recent relation, we obtain

$$\begin{aligned}
&\frac{1}{2}\times\frac{\alpha}{2^{2-\beta}}\|\xi^n\|^2+\frac{1}{2}\|\nabla\xi^n\|^2\\
&\quad\leq\left\|O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}\right\|^2+\frac{\alpha\gamma^2}{2}\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\left\|\xi^k\right\|^2\\
&\quad+\frac{\alpha\gamma^2}{(\Delta t)^2}\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\left(\left\|\xi^k\right\|^2+\left\|\xi^{k-1}\right\|^2\right)+\frac{\alpha\gamma^2}{(\Delta t)^2}\left\|\xi^{n-2}\right\|^2\\
&\quad+(\frac{\alpha\gamma^2}{2}+\frac{\alpha\gamma^2}{(\Delta t)^2}+\frac{\alpha}{2\times2^{2-\beta}}+1)\left\|\xi^{n-1}\right\|^2+\frac{1}{2}\left\|\nabla\xi^{n-1}\right\|^2,\quad n\geq2.
\end{aligned}$$

By Theorem 3 (the Poincaré–Friedrich inequality), there exists a constant $c_p>0$ such that

$$\begin{aligned}\frac{1}{2}\left\|\nabla\xi^n\right\|^2 \leq{}& \left\|O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}\right\|^2\\ &+\frac{\alpha\gamma^2}{2}c_p\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\left\|\nabla\xi^k\right\|^2\\ &+\frac{\alpha\gamma^2}{(\Delta t)^2}c_p\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\left(\left\|\nabla\xi^k\right\|^2+\left\|\nabla\xi^{k-1}\right\|^2\right)\\ &+\frac{\alpha\gamma^2}{(\Delta t)^2}c_p\left\|\nabla\xi^{n-2}\right\|^2\\ &+\left(\frac{\alpha\gamma^2}{2}+\frac{\alpha\gamma^2}{(\Delta t)^2}+\frac{\alpha}{2\times 2^{2-\beta}}+1\right)c_p\left\|\nabla\xi^{n-1}\right\|^2\\ &+\frac{1}{2}\left\|\nabla\xi^{n-1}\right\|^2,\quad n\geq 2.\end{aligned}\tag{39}$$

As we know, $\xi^0=0$. Without loss of generality, relations (37) and (39) can be written as

$$\begin{aligned}&\left\|\nabla\xi^1\right\|^2\leq\left\|O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}\right\|^2,\\ &\left\|\nabla\xi^n\right\|^2\leq 2\left\|O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}\right\|^2+\sum_{k=1}^{n-1}C_k\left\|\nabla\xi^k\right\|^2,\\ &\qquad n\geq 2,\quad C_k>0\quad for\quad k=1,\ldots,n-1.\end{aligned}\tag{40}$$

Thus, by using Theorem 4 (the discrete Gronwall theorem), the set of equations (40) yields

$$\left\|\nabla\xi^n\right\|^2\leq 2\left\|O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}\right\|^2\exp\left(\sum_{k=1}^{n-1}C_k\right),\quad n\geq 1,$$

and according to Theorem 3 (the Poincaré–Friedrich inequality), there exists a constant $\overline{c}_p>0$ such that

$$\left\|\xi^n\right\|^2\leq 2\overline{c}_p\left\|O(\Delta x)^2+O(\Delta y)^2+O(\Delta t)^{2-\beta}\right\|^2\exp\left(\sum_{k=1}^{n-1}C_k\right),\quad n\geq 1.\tag{41}$$

By using Lemma 3, it is easy to show that

$$\sum_{k=1}^{n-1}C_k\leq\left(2\alpha\gamma^2+\frac{8\alpha\gamma^2}{(\Delta t)^2}+\frac{\alpha}{2^{2-\beta}}+2\right)c_p+1.\tag{42}$$

Set $\zeta=\left(2\alpha\gamma^2+\frac{8\alpha\gamma^2}{(\Delta t)^2}+\frac{\alpha}{2^{2-\beta}}+2\right)c_p+1$. Then, relations (41) and (42) allow us to write

$$\|\xi^n\| \leq \sqrt{2\bar{c}_p \exp(\zeta)}\left\|O(\Delta x)^2 + O(\Delta y)^2 + O(\Delta t)^{2-\beta}\right\|, \quad n \geq 1, \zeta \geq 0, \bar{c}_p > 0. \tag{43}$$

□

5 Numerical experiments

In this section, we present some numerical tests that confirm the validity of the proposed numerical method. To measure the accuracy of the proposed method, we use the maximum absolute error given by

$$L_\infty = \max_{1\leq i\leq I, 1\leq j\leq J}\left|\tilde{U}_{i,j}(T) - U_{i,j}(T)\right|,$$

where $\tilde{U}_{i,j}(T)$ and $U_{i,j}(T)$ denote the numerical solution and the exact solution of (1) with initial and boundary conditions (2)–(4) at (x_i, y_j) and time T, respectively.

Example 1. Consider a two-dimensional test problem of the form (1), with $\Omega = [0,1]\times[0,1]$, $f(x,t) = \left(\frac{24t^{4-\beta}}{\Gamma(5-\beta)} + \frac{24t^{5-\beta}}{\Gamma(6-\beta)} + 2t^4\pi^2\right)\sin(\pi x + \pi y) + t^4\sin(\pi x + \pi y)$, and suppose that the initial and boundary conditions are assumed using the exact solution $u(x,y,t) = t^4\sin(\pi x + \pi y)$; see [13]. Now, we provide some tests.

Test 1 Kumar, Bhardwaj, and Dubey [13] considered this example using a local meshless method with 2025 points on Ω. They reported the maximum absolute errors and CPU time with $\beta = 1.7, 1.9$ and different values for Δt at the time $T = 1.0$. Using the proposed method, we repeated this test. We considered this example by assuming $I = J = 45$ (2025 points on $\overline{\Omega}$). To solve the linear system of equations, we used the GMRES-m method with $m = 20$.

Table 1 presents the maximum absolute errors and CPU time obtained by Kumar, Bhardwaj, and Dubey [13] and the results of the proposed method with $\beta = 1.7$, different values for Δt, and 2025 points on $[0,1]\times[0,1]$ at $T = 1.0$. Table 2 presents the maximum absolute errors and CPU time obtained in [13] and the results of the proposed method with $\beta = 1.9$, different values for Δt, and 2025 points on $[0,1]\times[0,1]$ at $T = 1.0$.

As Tables 1 and 2 show, the maximum absolute errors and the CPU time of [13] and those of the proposed method are close, but the CPU time of the proposed method is smaller than that of [13].

The following tests show that the proposed method provides acceptable accuracy with a smaller number of points on Ω.

Test 2 We considered this example by the proposed method with $\Delta x = \Delta y = 0.1$, $\beta = 1.5, 1.9$, and different values for Δt. According to Table 3, with different values for Δt, the maximum absolute errors were small enough

at $T = 1.0$. Also, decreasing the size of the time step increased the CPU time very slowly and improved the accuracy. The value $\Delta t = \frac{1}{80}$ was selected for the next test.

Test 3 We considered this example by the proposed method with $\Delta t = \frac{1}{80}$, $\beta = 1.5, 1.9$, and different values for $\Delta x, \Delta y$. According to Table 4, the accuracy was acceptable. Also, the CPU time was reasonable with $\Delta x = \Delta y = \frac{1}{10}, \frac{1}{20}$. Moreover, by decreasing Δx and Δy to $\frac{1}{40}, \frac{1}{80}$, the CPU time increased rapidly, and the accuracy did not improve significantly. According to relation (43), the convergence rate of our method depends on Δx, Δy, and Δt. In this case, the space steps decrease, but the time step is constant. Therefore the accuracy does not improve.

As shown in Tests 2 and 3, a very small size the of space step is not recommended, but small size of a time step is recommended. According to Diethelm, Garrappa, and Stynes [7], a high-order space discretization for a time-fractional partial differential equation is not advisable. They believe that to reach a high convergence, we must choose very small size of the time step in comparison with the size of the space step. Our experiments confirmed this idea.

Table 1: Comparison of the maximum absolute errors and CPU time with $\beta = 1.7$, different values for Δt, and 2025 points on $[0, 1] \times [0, 1]$ at $T = 1.0$

Δt	L_∞ [13]	L_∞	CPU (s) [13]	CPU (s)
$\frac{1}{10}$	$1.2917e-02$	$1.2575e-02$	1.751	1.414
$\frac{1}{20}$	$5.4532e-03$	$8.7100e-03$	2.210	1.996
$\frac{1}{40}$	$2.3351e-03$	$5.0648e-03$	3.062	2.746

Table 2: Comparison of the maximum absolute errors and CPU time with $\beta = 1.9$, different values for Δt, and 2025 points on $[0, 1] \times [0, 1]$ at $T = 1.0$

Δt	L_∞ [13]	L_∞	CPU (s) [13]	CPU (s)
$\frac{1}{10}$	$2.7619e-02$	$1.6456e-02$	1.751	1.298
$\frac{1}{20}$	$1.3079e-02$	$1.0294e-02$	2.210	1.613
$\frac{1}{40}$	$6.1953e-03$	$5.7166e-03$	3.062	2.119

6 Conclusion

The Crank–Nicolson difference scheme can be used easily for space-fractional equations, but some manipulations are needed for time-fractional equations. In this paper, the Crank–Nicolson method was extended for the discretization of a TFTE. The solvability, stability, and convergence of this proposed

Table 3: Maximum absolute errors and CPU time for different values of Δt and β, with $\Delta x = \Delta y = 0.1$ at $T = 1.0$

	$\beta = 1.5$		$\beta = 1.9$	
Δt	L_∞	$CPU(s)$	L_∞	CPU (s)
$\frac{1}{10}$	$1.3159e-02$	0.1069	$1.9190e-03$	0.1041
$\frac{1}{20}$	$1.0567e-02$	0.1249	$1.2971e-03$	0.1269
$\frac{1}{40}$	$7.5001e-03$	0.1673	$8.3675e-03$	0.1713
$\frac{1}{80}$	$5.5027e-03$	0.2944	$5.7025e-03$	0.2908
$\frac{1}{160}$	$4.3813e-03$	0.6513	$1.9956e-03$	0.6423
$\frac{1}{320}$	$3.7898e-03$	1.7490	$3.3609e-03$	1.7547

Table 4: Maximum absolute errors and CPU time for different values of Δx, Δy, and β, with $\Delta t = \frac{1}{80}$ at $T = 1.0$

	$\beta = 1.5$		$\beta = 1.9$	
$\Delta x = \Delta y$	L_∞	$CPU(s)$	L_∞	CPU (s)
$\frac{1}{10}$	$5.5027e-03$	0.2944	$5.7025e-03$	0.2908
$\frac{1}{20}$	$3.1200e-03$	0.4722	$3.6015e-03$	0.4273
$\frac{1}{40}$	$2.5315e-03$	3.6787	$3.0738e-03$	2.2135
$\frac{1}{80}$	$2.3744e-03$	74.7221	$2.9432e-03$	35.9883

method were proved. The numerical results were accurate enough. According to the numerical tests, to reach a high convergence, a very small size of the space step is not recommended, but a small size of the time step is recommended.

References

1. Bagley, R.L. and Torvik, P. *A theoretical basis for the application of fractional calculus to viscoelasticity.* J. Rheol. 27(3) (1983) 201–210.

2. Banasiak, J. and Mika, J.R. *Singularly perturbed telegraph equations with applications in the random walk theory.* J. Appl. Math. Stochastic Anal. 11(1) (1998) 9–28.

3. Biazar, J., Ebrahimi, H. and Ayati, Z. *An approximation to the solution of telegraph equation by variational iteration method.* Numer. Methods Partial Differ. Equ. 25(4) (2009) 797–801.

4. Cen, Z. Huang, J. Xu, A. and Le, A. *Numerical approximation of a time-fractional Black–Scholes equation.* Comput. Math. Appl. 75(8) (2018) 2874–2887.

5. Chen, J., Liu, F. and Anh, V. *Analytical solution for the time-fractional telegraph equation by the method of separating variables.* J. Math. Anal. Appl. 338(2) (2008) 1364–1377.

6. Das, S. Vishal, K. Gupta, P.k. and Yildirim, A. *An approximate analytical solution of time-fractional telegraph equation.* Appl. Math. Comput. 217(18) (2011) 7405–7411.

7. Diethelm, k. Garrappa, R. and Stynes, M. *Good (and not so good) practices in computational methods for fractional calculus.* Mathematics, 8(3) (2020) 324.

8. Horn, R. A. and Johnson, C. R. *Matrix analysis.* Cambridge university press, 2012.

9. Hosseini, V.R. Chen, W. and Avazzadeh, Z. *Numerical solution of fractional telegraph equation by using radial basis functions.* Eng. Anal. Bound. Elem. 38 (2014) 31–39.

10. Hosseini, V.R. Shivanian, E. and Chen, W. *Local integration of 2-D fractional telegraph equation via local radial point interpolant approximation.* Eur. Phys. J. Plus, 130(2) (2015) p. 1–21.

11. Jiang, W. and Lin, Y. *Representation of exact solution for the time-fractional telegraph equation in the reproducing kernel space.* Commun. Nonlinear Sci. Numer. Simul. 16(9) (2011) 3639–3645.

12. Karatay, I. Kale, N. and Bayramoglu, S. *A new difference scheme for time fractional heat equations based on the Crank-Nicholson method.* Fract. Calc. Appl. Anal. 16(4) (2013) 892–910.

13. Kumar, A. Bhardwaj, A. and Dubey, S. *A local meshless method to approximate the time-fractional telegraph equation.* Eng. Comput. 37(4) (2021) 3473–3488.

14. Li, C. and Cao, J. *A finite difference method for time-fractional telegraph equation.* in Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications. 2012. IEEE 314–318.

15. Liang, Y. Yao, Z. and Wang, Z. *Fast high order difference schemes for the time fractional telegraph equation.* Numer. Methods Partial Differ. Equ. 36(1) (2020) 154–172.

16. Mohebbi, A. Abbaszadeh, M. and Dehghan, M. *The meshless method of radial basis functions for the numerical solution of time fractional telegraph equation.* Internat. J. Numer. Methods Heat Fluid Flow 24 (2014), no. 8, 1636–1659.

17. Momani, S. *Analytic and approximate solutions of the space- and time-fractional telegraph equations.* Appl. Math. Comput. 170(2) (2005) 1126–1134.

18. Nemati, S. Lima, P.M. and Torres, D.F. *A numerical approach for solving fractional optimal control problems using modified hat functions.* Commun. Nonlinear Sci. Numer. Simul. 78 (2019) 104849.

19. Nikan, O. Avazzadeh, Z. and Machado, J.T. *Numerical approximation of the nonlinear time-fractional telegraph equation arising in neutron transport.* Commun. Nonlinear Sci. Numer. Simul. 99 (2021) 105755.

20. Quarteroni, A. and Valli, A. *Numerical approximation of partial differential equations.* Vol. 23. Springer Science and Business Media, 2008.

21. Reddy, B.D. *Introductory functional analysis: with applications to boundary value problems and finite elements.* Springer Science and Business Media, 2013.

22. Sepehrian, B. and Shamohammadi, Z. *Numerical solution of nonlinear time-fractional telegraph equation by radial basis function collocation method.* Iran. J. Sci. Technol. Trans. A: Sci. 42(4) (2018) 2091–2104.

23. Shivanian, E. *Spectral meshless radial point interpolation (SMRPI) method to two-dimensional fractional telegraph equation.* Math. Methods Appl. Sci. 39(7) (2016) 1820–1835.

24. Shivanian, E. Abbasbandy, S. Alhuthali, M.S. and Alsulami, H.H. *Local integration of 2-D fractional telegraph equation via moving least squares approximation.* Eng. Anal. Bound. Elem. 56 (2015) 98–105.

25. Sun, H. Zhang, Y. Baleanu, D. Chen, W. and Chen, Y. *A new collection of real world applications of fractional calculus in science and engineering.* Commun. Nonlinear Sci. Numer. Simul. 64 (2018) 213–231.

26. Uchaikin, V.V. *Fractional derivatives for physicists and engineers.* Vol. 2. 2013: Springer.

27. Vyawahare, V.A. and Nataraj, P. *Fractional-order modeling of neutron transport in a nuclear reactor.* Appl. Math. Model. 37(23) (2013) 9747–9767.

28. Vyawahare, V.A. and Nataraj, P. *Analysis of fractional-order telegraph model for neutron transport in nuclear reactor with slab geometry.* in 2013 European control conference (ECC). 2013. IEEE.

29. Wang, Y. and Mei, L. *Generalized finite difference/spectral Galerkin approximations for the time-fractional telegraph equation.* Adv. Difference Equ. 2017(1) (2017) 1–16.

30. Wei, L., Liu, L. and Sun, H. *Numerical methods for solving the time-fractional telegraph equation.* Taiwanese J. Math. 22(6) (2018) 1509–1528.

31. Yildirim, A. *He's homotopy perturbation method for solving the space- and time-fractional telegraph equations.* Int. J. Comput. Math. 87(13) (2010) 2998–3006.

32. Zhao, Z. and Li, C. *Fractional difference/finite element approximations for the time-space fractional telegraph equation.* Appl. Math. Comput. 219(6) (2012) 2975–2988.

How to cite this article
H. Hajinezhad, A.R. Soheili A numerical approximation for the solution of a time-fractional telegraph equation based on the Crank–Nicolson method. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 607-628. doi: 10.22067/IJNAO.2022.77142.1154.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 629–657
DOI:10.22067/ijnao.2022.77130.1153
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Differential transform method: A comprehensive review and analysis

H.H. Mehne

Abstract

The complexity of solving differential equations in real-world applications motivates researchers to extend numerical methods. Among different numerical and semi-analytical methods for solving initial and boundary value problems, the differential transform method (DTM) has received notable attention. It has developed and experienced generalizations for implementing other types of mathematical problems such as optimal control, calculus of variations, and integral equations. This review aims to provide insight into DTM. History, theoretical base, applications, computational aspects, and its revisions are reviewed. The present study helps to understand the theory, capabilities, and features of the DTM, as well as its drawbacks and limitations.

AMS subject classifications (2020): Primary 34A25; Secondary 65L10, 65N99.

Keywords: Boundary value problems; Initial value problems; Differential Transform Method; Semi-analytical methods.

1 Introduction

There are many practical problems, which are formulated as boundary value problems (BVPs). They have appeared, for example, in studying the boundary layer flow [61], the squeezing nanofluids [62], the formation of rogue waves in the ocean [20], electrical heating of conductors [26], and modeling the behavior of induction motors [6]. In addition, other types of important problems, such as calculus of variations or optimal control problems, are reduced to a set of BVPs or initial value problems (IVPs). The multitude of such

* Corresponding author

Received 11 June 2022; revised 9 October 2022; accepted 15 October 2022

Hamed Hashemi Mehne
Aerospace Research Institute, Tehran, Iran. e-mail: hmehne@ari.ac.ir

applications and the complexity of solving BVPs motivated the researcher to develop solving methods. Solving strategy has three categories as follows:

- Analytical methods: Methods that find the exact or analytical solution of the BVPs as a function or closed-form are known as analytical methods. Direct integration [44], method of images [44], separation of variables, and Green's function method [37] are some examples of analytical methods. Despite exact results, the analytical methods are usually restricted to simple or special forms of BVPs. Moreover, they require manual calculations that make their implementation on computers difficult.

- Numerical algorithms: Numerical methods are suitable for computerization while their results contain errors and convergences issues should be checked. These types of methods are based on the numerical approximation of derivatives like finite difference [51] and shooting method [52]. Some of the numerical methods have also motivated from the physics of the problem, such as lattice Boltzmann [83].

- Semi-analytical methods: When the result of a method is a function or a sequence of functions converging to the exact solution, the method is semi-analytical. Collocation finite element [52], Galerkin finite element [52], Adomian decomposition [22], and iterative approximations [55] are some examples of semi-analytical methods. They have the advantage of finding function answers and computer implementation while they have errors in results.

Among semi-analytical methods, the differential transform method (DTM) is one of the most popular and practical algorithms. This method was introduced by Zhou [95] in 1986 for solving IVPs in the field of electrical circuits. The method is based on the calculation of the coefficients of the Taylor series of the problem's solution. The method has been developed for solving BVPs in one and more dimensions, integral equations, calculus of variations, and optimal control. Especially in the last decade, it was used for analyzing several physical phenomena with stochastic and fractional behavior.

These applications and implementations of the DTM are motivations for reviewing the method in the present work. The aim is to give a comprehensive review of the method, including the theory, improvements, and applications. Some examples related to the method are explained to show its accuracy and benefits. Finally, the restrictions and drawbacks of the method are noted.

The present review helps researchers who are attending to use the method for solving a practical problem to be familiar with this method and its limitations.

The review is organized as follows: after introductions in Section 1, the literature review and history are given in Section 2. Section 3 assigns the method description, benefits, and drawbacks. Finally, some concluding remarks are expressed in Section 4.

2 Literature review

This section is divided into two subsections: the historical and application reviews. A graphical review based on the research' subjects is also included.

2.1 Historical review

In this subsection, the papers with independent research on extending or improving the DTM have been reviewed in historical order.

- 1986: The concept of differential transform was established by Zhou, a Chinese researcher in the field of electrical engineering. The method was originally explained in [95] for IVPs.
- 1996: DTM has been extended to cover the hybrid boundary conditions for eigenvalue problems in [23].
- 1998: The method was improved in the case of BVPs in an infinite horizon. As a practical implementation, it has been implemented to solve the Blasius problem efficiently in [94].
- 1999: The two-dimensional differential transform has been proposed for solving IVPs with partial differential equations in [24].
- 2003: Following the extensions for two-dimensional DTM, new theorems were given in [14] with applications of the method for diffusion equation.
- 2004: The three-dimensional DTM was introduced for solving systems of partial differential equations (PDEs) accompanied by the initial conditions in [15]. The DTM has also been applied to find accurate solutions for algebraic differential equations of ordinary type in [16].
- 2005: The integro-differential equations with boundary conditions were the next type of problems examined for their solution by DTM in [9]. General theorems were derived, and the method was successfully applied to solve examples of linear and nonlinear integro-differential problems.
- 2006: The DTM was extended to solve difference equations with different types and orders in [10]. Solving differential-difference equations with boundary conditions with DTM have been also reported in [11].
- 2007: The concept of fractional derivatives and the growing topic of fractional differential equations cause to define the fractional differential transform. In [65], the theory of fractional DTM was established for

solving ordinary fractional with initial conditions. The method was later proposed using generalized Taylor series and Caputo fractional derivative.

- 2008: The fractional DTM for ordinary equations has been more generalized for equations with multi-order in [31]. The method of fractional DTM has been extended to linear PDEs of fractional order in [69]. The extension of the method to systems of fractional PDEs with initial conditions was also given in [32]. A modified DTM based on Laplace transform and Padé approximation was introduced to find oscillatory solutions.

- 2009: The two-dimensional DTM was implemented to solve a class of linear and nonlinear Volterra integral equations in [87]. To resolve the complexity of computation in a multidimensional DTM, a reduced method was introduced in [53]. The method is based on the separation of variables. The efficiency of the method has been demonstrated by its application on several IVPs. In the follow-up to the fractional derivatives, the DTM was examined for fractional integro-differential equations in [12]. Another notable work is [76], where the DTM is combined with Padé approximation to solve BVPs with infinite horizon. The fuzzy DTM was introduced in [7] to solve fuzzy differential equations. The method is based on the generalized H-differentiability. The DTM was also applied to solving nonlinear optimal control problems in [43]. Two approaches for finite and infinite horizon problems were proposed based on the minimum principle and the dynamic programming on Hamilton–Jacobi–Bellman equations, respectively, in combination with DTM.

- 2010: To accelerate the convergence of the DTM solution, a multi-step DTM was proposed in [70]. In this version of DTM, the solution is a piecewise function consisting of a finite number of DTM solutions for consecutive time intervals. Another derivation of DTM called projected DTM, was also proposed in [45]. In this method, the solution of two-dimensional PDEs is obtained with DTM for one variable while the coefficients are functions of the other variable.

- 2011: The piecewise DTM has been further extended for solving fractional chaotic dynamical systems in [8]. It is indeed the extension of [70] in fractional cases.

- 2012: Random DTM is another version of DTM for solving random differential equations based on the mean fourth calculus proposed in [90]. The results of the implementation of the method for Riccati differential problems show the efficiency of this approach. A combination of the Adomian decomposition method and DTM was proposed in [29] for solving fractional differential equations.

- 2013: The fuzzy DTM method [7] has been extended to cover solving Volterra integral equations in [81]. A combination of DTM with Adomian polynomial was proposed in [34] to overcome the problem of nonlinear terms in ordinary differential equations (ODEs) when DTM is used. For the calculus of variation problem with a differentiable solution, there exists a two-point boundary problem obtained by the Euler–Lagrange equation. Using the method proposed in [67], this problem was solved by the DTM to derive a numerical method for finding semi-analytical stationary functions. A DTM for solving linear optimal control problems with a quadratic performance index was introduced in [80]. The method uses the Pontryagin maximum principle to obtain a BVP, which is finally solved by DTM. Reduced DTM is examined for solving two-dimensional Volterra integral equations in [2]. Based on the simulations, the results of reduced MTD are more accurate in comparison with traditional DTM.

- 2014: In [3], the nonlinear integro-differential equations with proportional delay are under investigation with DTM. Some theorems related to the delayed functions and their transforms were also proved in addition to the numerical simulation. To enlarge the domain of convergence of DTM, a method was proposed in [17]. The Laplace–Padé resummation was examined to solve partial differential algebraic equations.

- 2015: The generalized DTM method for IVPs on fractional PDEs has been extended to BVPs in [27]. DTM was applied in [35] as a new tool to compute the Laplace transform of real-valued single variable functions. The Cauchy-type singular integral equations are solved by a proposed method based on DTM in [4]. The forms of differential transform of kernel functions were obtained with high-accuracy solutions on several examples with two kernel types. DTM was also used to solve optimal control problems in [66]. The method is based on applying the DTM to the BVPs resulting from sufficient conditions for solving linear and nonlinear optimal control problems.

- 2016: Two-dimensional extended DTM was proposed for solving PDEs with local fractional derivatives in [93]. The concept of this version of DTM for nondifferential functions was analyzed, and basic theorems were proved. The efficiency and accuracy of the method were shown via numerical simulations. A class of BVPs defined for nonlinear singular second-order ODE was examined with DTM in [91]. The method benefits from the Adomian polynomials to overcome the nonlinear terms. In addition, to demonstrate the applicability of the method by some examples, an upper bound for the error was also obtained. Multi-point BVPs also found their DTM solution. The problem of unknown initial conditions in these types of problems was resolved in [92]. The first

two DTM coefficients were taken unknown and were determined from a system of algebraic equations.

- 2017: A version of DTM was introduced in [88] for solving comfortable fractional differential equations. The generalized DTM for solving fractional problems was further studied from a theoretical perspective in [71]. The sufficient conditions for convergence of the method and estimation of truncation error were obtained. An efficient version of multi-step DTM was addressed in [70]. The method reduces the number of subintervals and consequently improves the computational complexity of multi-step DTM.

- 2018: The projected DTM was combined with integral transform to provide an efficient method for solving fractional PDEs in [82]. The results showed that the method is accurate and fast convergent. Fuzzy DTM was extended for solving fuzzy Volterra integro-differential equations in [19]. The method is based on a generalization of Seikkala differentiability for fuzzy functions.

- 2019: The switching DTM was introduced in [61] to cover infinite horizons, that is, boundary conditions at infinity. In the proposed approach, the solution has two parts: a DTM solution and an analytical solution that matches the condition at infinity.

- 2020: In [30], the method of [29] was applied for computing two-dimensional DTM solutions of PDE problems. The method reduced the computational complexity of traditional two-dimensional DTM. A combination of differential transform and smoothed particle hydrodynamics was proposed in [57] for solving transient heat conduction problems. Numerical simulations showed that the method is robust and accurate. Tarig transform was combined with projected DTM to develop an effective method for solving fractional nonlinear PDEs in [60].

- 2021: As recent applications of DTM to practical problems, we can address [46], where the problem of thermal distribution through a longitudinal trapezoidal moving fin has been investigated using one-dimensional Padé-DTM. Similar work for a moving rod was reported in [84], where two-dimensional Padé-DTM is implemented.

- 2022: A comparison between sinc approximation and DTM on nonlinear Hammerstein integral equations has been made in [50]. In the case of separable kernels, the DTM performs more accurately and faster than the sinc approximation. Integro-differential equations with a retarded argument have notable engineering applications. In [42], these types of problems have been solved by DTM with satisfactory and applicable results.

To demonstrate the review of fulfilled research on the concept of DTM, a graphical tree of a subject is given in Figure 1. There are four main blocks determining the top subjects, along with subblocks indicating the details. The related references to each subject are written close to the related box.

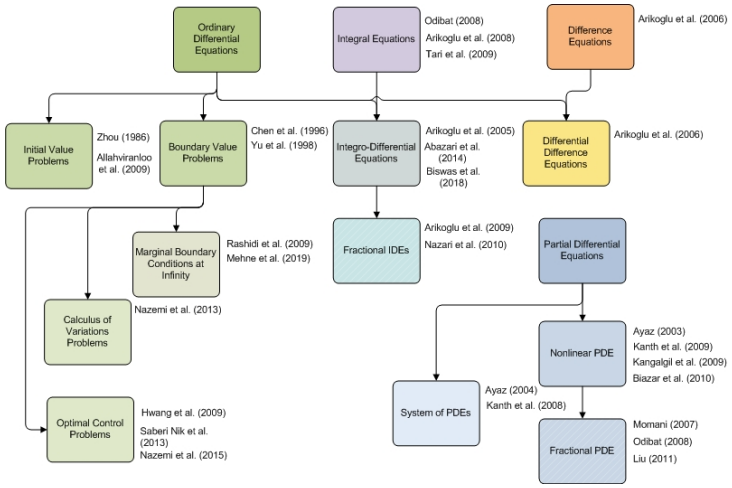


Figure 1: Subjective review of researches on DTM

2.2 Application review

In this subsection, some notable applications of DTM in the real world and practical problems are listed.

- Fluid mechanics: Fractional coupled Burgers' equations [58], Blssius equation of boundary layer flow [94], nanoparticle migration [56], nano boundary-layers over-stretching surfaces [77], magnetohydrodynamics (MHD) boundary-layer equations [76], MHD in a laminar liquid film [78], parametric investigation of the thermal analysis for solar collectors [28], the study of time-dependent MHD heat transfer flow of Jeffrey fluid [54], and analysis an unsaturated single-phase fluid flow in porous media [25].

- Electrical engineering: Solving telegraph equation by DTM [18] and reduced DTM [86, 85], solving Thomas-Fermi equation by the improved DTM [38], and dynamic simulation of power systems [59].

- Acoustics: KdV and modified KdV equations [48], two-dimensional fractional Helmholtz equation [5], and Kadomtsev–Petviashvili equations [63].
- Physics: Solving a model of fractional telegraph point reactor kinetics [39] and solving Fokker–Planck equation [41].
- Quantum Mechanics: Klein–Gordon equation [79] and Burgers–Huxley equations [1].
- Structures and vibration: Vibration analysis of a rotating tapered cantilever Bernoulli–Euler beam [72], nonlinear oscillators [64], analysis and prediction of vibration of a nanobeam [40], investigation of flapwise bending free vibration of isotropic rotating Timoshenko microbeams [13], analyzing the thermal buckling of a functionally graded circular plate [33], solving nonlinear Duffing oscillator [68], and buckling analysis of nanobeams [47].
- Miscellaneous applications: Population growth estimation [73], solving a typhoid fever model [74], solving tumor-immune system [49], analysis of fish-farm model [89], solving the model of pollution for a system of lakes [17], and modeling of jamming transition problem in traffic flow [36].

3 How does DTM work?

In this section, the basic definitions and fundamental properties of the differential transform method are presented. Let us consider the following ordinary differential equation:

$$T(x, u(x), u'(x), u''(x), \ldots, u^{(n)}(x)) = 0, \tag{1}$$

where T is a transformation on a class of sufficiently differentiable functions $u(x)$. Assume that under specific conditions, the above-mentioned differential equation has a unique solution $u(x)$ satisfying in

$$u(0) = u_0, u'(0) = u'_0, \ldots, u^{(n)}(0) = u_0^{(n)}, \tag{2}$$

where $u_0, u'_0, \ldots, u_0^{(n)}$ are given. Now, the aim of DTM in the simplest case is to solve the IVP (1)–(2). Let us consider the Taylor series of the solution in a neighborhood of $x = 0$:

$$u(x) = \sum_{k=0}^{\infty} \frac{u^{(k)}(0)}{k!} x^k. \tag{3}$$

It may be also rewritten as

$$u(x) = \sum_{k=0}^{\infty} U(k)x^k, \tag{4}$$

where

$$U(k) = \frac{u^{(k)}(0)}{k!} \tag{5}$$

Therefore, if the values of $U(k)$ are available, then the solution may be constructed from (4). This is the key of the DTM method that defines a transformation from $u(x)$ to the set of coefficients $\{U(1), U(2), \ldots\}$ and vice versa. This transformation is called the differential transformation. Now, in DTM, $U(k)$'s are substituted in (1) converting it to a system of algebraic equations. This will be performed using the basic properties of the differential transform. Some of these properties are listed below:

Let us assume, for simplicity, that $\xrightarrow{\text{DT}}$ denotes the differential transform. If λ is a constant scalar, $u(x) \xrightarrow{\text{DT}} U(k)$, and $v(x) \xrightarrow{\text{DT}} V(k)$, where $U(k)$ and $V(k)$ are differential transformations of $u(x)$ and $v(x)$, respectively, then

- $u(x) + v(x) \xrightarrow{\text{DT}} U(k) + V(k)$;
- $\lambda u(x) \xrightarrow{\text{DT}} \lambda U(k)$;
- $u(x)v(x) \xrightarrow{\text{DT}} \sum_{i=0}^{k} U(i)V(k-i)$;
- $u'(x) \xrightarrow{\text{DT}} (k+1)U(k+1)$;
- $u''(x) \xrightarrow{\text{DT}} (k+1)(k+2)U(k+2)$;
- $u^{(n)}(x) \xrightarrow{\text{DT}} (k+1)(k+2)\cdots(k+n)U(k+n)$;
- $u(x) = \int_0^x v(s)ds \xrightarrow{\text{DT}} U(k) = \begin{cases} \frac{V(k-1)}{k}, & k \geq 1, \\ 0, & k = 0; \end{cases}$
- $u(x) = x^n \xrightarrow{\text{DT}} U(k) = \delta(k-n) = \begin{cases} 1, & k = n, \\ 0, & k \neq n; \end{cases}$
- $u(x) = e^{\lambda x} \xrightarrow{\text{DT}} U(k) = \frac{\lambda^k}{k!}$.

These properties will be obtained directly for the definition of the differential transform given by (5).

In what follows, the implementation of the method for solving a simple IVP is given.

Example 1. Let us consider the following IVP:

$$(x^2+9)u'' + 2xu' = 0, \tag{6}$$

$$u(0) = \pi, \qquad u'(0) = \frac{4}{3}. \tag{7}$$

The problem has the following unique solution:

$$u(x) = 4\tan^{-1}\left(\frac{x}{3}\right) + \pi. \tag{8}$$

To implement the DTM on this problem, let us assume that $u(x) \xrightarrow{\text{DT}} U(k)$. Then, by the above-mentioned properties of differential transform and substituting the corresponding transforms of individuals terms of (6), we have

$$\sum_{i=0}^{k} \left(\delta(i-2) + 9\delta(i)\right)(k-i+1)(k-i+2)U(k-i+2)$$
$$+2\sum_{i=0}^{k} \delta(i-1)(k-i+1)U(k-i+1) = 0. \tag{9}$$

Regarding the initial conditions in (7) and the definition of Dirac delta function, the transformed problem is defined by the following recursive converted equation:

$$U(0) = \pi, \tag{10}$$
$$U(1) = \frac{4}{3}, \tag{11}$$
$$U(k+2) = \frac{-k}{9(k+2)}U(k), \quad k \geq 0. \tag{12}$$

The unknown coefficients will be calculated from the above relations, and then the solution of the problem in the form of an infinite series is determined by (4). One can truncate this series with n terms as

$$u_n(x) = \sum_{k=0}^{n} U(k)x^k \tag{13}$$

to approximate the solution. For example, the solution for $n = 8$ with 4-digit accuracy is calculated as follows:

$$u_8(x) = 3.1416 + 1.3333x - 0.04934x^3 + 0.0033x^5 - 0.0003x^7. \tag{14}$$

The approximate solution (14) has decreasing coefficients and indicates that $\{u_n(x)\}$ converges pointwise to the solution of the problem when $0 \leq x < 1$. As also depicted in Figure 2, the obtained DTM solution (14) is very close to the exact one. However, when $x > 1$, the convergence of the sequence of approximations does not guarantee. For instance, in Figure 3, the exact and the DTM solutions have been drawn for $0 \leq x \leq 4$. As it can be seen, despite the coincidence of the DTM and exact solutions in $[0, 1]$, the DTM solution diverges for $x > 1$. Therefore, when using the DTM, we have to check the

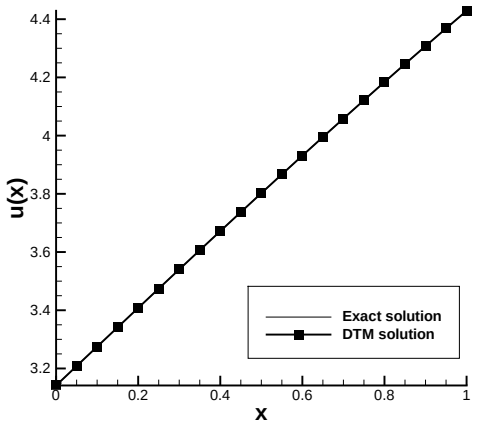


Figure 2: Exact and numerical solutions of Example 1 for $0 \leq x \leq 1$.

range of validity of the solution. In the next example, the implementation of the method on a BVP is discussed.

Example 2. Let us consider the following BVP:

$$(1+x^2)u'' + xu' - u = x^2, \tag{15}$$

$$u(0) = 1, \qquad u(1) = -\frac{\sqrt{5}}{6} + \frac{\sqrt{2}}{3} + 1. \tag{16}$$

The problem has the following unique solution:

$$u(x) = -\frac{\sqrt{5}}{6}x + \frac{1}{3}\sqrt{1+x^2} + \frac{1}{3}(2+x^2). \tag{17}$$

The corresponding equation in the transform space has the following form:

$$\sum_{i=0}^{k} \left(\delta(i-2) + \delta(i)\right)(k-i+1)(k-i+2)U(k-i+2) + \sum_{i=0}^{k} \delta(i-1)(k-i+1)U(k-i+1) - U(k) = \delta(k-2). \tag{18}$$

The first boundary condition leads to $U(0) = 1$, however the value of $U(1)$ is unknown and will be found by using the second boundary condition. Let us assume temporary that $U(1) = \alpha$. Then implementing the conditions and properties of Dirac delta function results in

$$U(0) = 1, \tag{19}$$

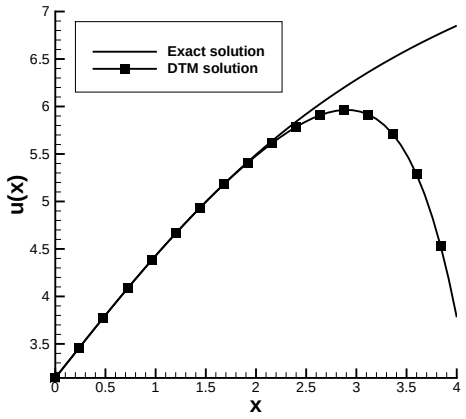


Figure 3: Exact and numerical solutions of Example 1 for $0 \leq x \leq 4$.

$$U(1) = \alpha, \tag{20}$$

$$U(2) = \frac{1}{2}, \tag{21}$$

$$U(3) = 0, \tag{22}$$

$$U(4) = -\frac{1}{24}, \tag{23}$$

$$U(k+2) = -\frac{k-1}{k+2}U(k), \qquad k \geq 3. \tag{24}$$

Therefore, the solution with $n = 8$ terms has the following form:

$$u_8(x) = 1.0000 + \alpha x + 0.5000x^2 - 0.0417x^4 + 0.0208x^6 - 0.0130x^8. \tag{25}$$

Now, we implement the second boundary condition at $x = 1$ to the above solution and find $\alpha = -0.3674$. The exact value of α, which is obtained from the exact solution, is -0.3727. The resulting DTM solution in this case is close to the exact one as depicted in Figure 4.

If the interval of the solution is extended to $[0, 2]$, with boundary condition $u(2) = 2$, then the exact solution remains unchanged and $\alpha = 0.8320$ differs from -0.3727, for the exact value of $u(2)$. Therefore, a large deviation from the exact solution is anticipated. When the two curves are compared (Figure 4), we can see that the behavior of the DTM solution differs from the exact one due to the power of x above 1. This is similar to the case of IVPs, except that here the constraint on the second point forces the solution to prevent large oscillations.

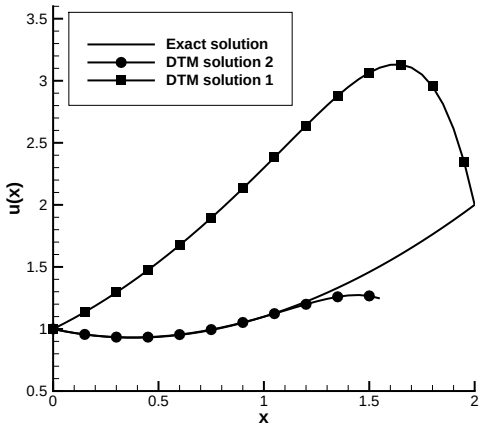


Figure 4: Exact and numerical solutions of Example 2 for $0 \leq x \leq 1$ and $0 \leq x \leq 2$.

3.1 Extensions and improvements

After the early implementation of DTM to initial and boundary value problems for ODEs, the researchers extended the method for other types of mathematical problems as encountered in Section 2. In the present section, some of these modifications are explained.

3.2 Multi-step DTM

As indicated in Example 2, the domain where the DTM solution is valid is usually narrow. In order to extend the solution for large intervals of independent variables, the multi-step DTM is proposed. The method is applied in sub-intervals instead of the entire domain. The solution is, indeed, a piecewise function of particular DTM solutions of the following form:

$$u(x) = \begin{cases} \sum_{k=0}^{N} U_0(k)x^k, & x \in [0, x_1], \\ \sum_{k=0}^{N} U_1(k)(x-x_1)^k, & x \in [x_1, x_2], \\ \vdots & \vdots \\ \sum_{k=0}^{N} U_p(k)(x-x_{p-1})^k, & x \in [x_{p-1}, x_p]. \end{cases} \tag{26}$$

The initial condition for each piece is obtained from the previous stage. Therefore, the method has errors but leads to better results compared to the traditional one. As an example, the multi-step DTM solution has been

obtained for Example 1 as follows:

$$u(x) = \begin{cases} 3.1416 + 1.3333x - 0.0494x^3 + 0.0033x^5 - 0.0003x^7, & x \in [0,1], \\ 3.2705 + 1.0797x + 0.1086x^2 - 0.0230x^3 - 0.00066x^4 \\ \quad - 0.0020x^5 + 0.0016x^6 - 0.0002x^7, & x \in [1,2], \\ 3.6890 + 0.7183x + 0.1688x^2 - 0.0518x^3 + 0.0326x^4 \\ \quad - 0.0151x^5 + 0.0030x^6 - 0.0002x^7, & x \in [2,3], \\ 4.5191 + 0.6477x^2 - 0.3598x^3 + 0.1439x^4 - 0.0336x^5 \\ \quad + 0.0040x^6 - 0.0002x^7, & x \in [3,4]. \end{cases} \tag{27}$$

Figure 5 Demonstrates the resulting multi-step DTM solution, the exact

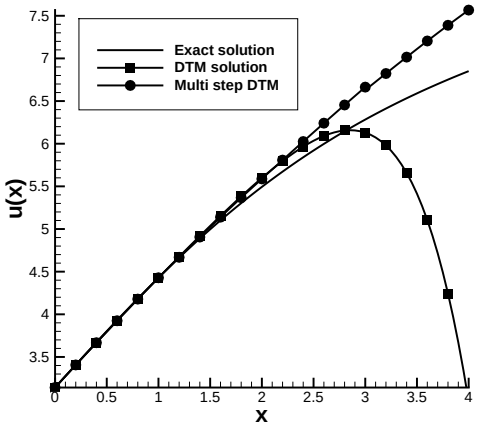


Figure 5: Exact, DTM, and multi-step DTM solutions of Example 1 for $0 \leq x \leq 4$.

and the one-step DTM solution. Comparing these curves elucidate that the multi-step DTM is more close to the exact solution and does not diverge like the traditional DTM solution.

3.3 Infinite horizons

There are BVPs with some conditions at infinity; that is, the domain of the independent variable is not bounded. In this subsection, two approaches of DTM when facing this situation are reviewed.

3.3.1 Padé Approximation

One of the well-known methods to approximate a real-valued function as a rational function is Padé approximation, which is usually used when simulating the behavior of a function at infinity is desired. This method has been combined with DTM to solve the infinite horizon BVP for the first time in [76]. Despite the application of this method in solving problems with conditions at infinity, such as [78, 75], it seems that this approach is not applicable. To illustrate this issue, let us consider the following rational function:

$$R_{L,M}(x) = \frac{p_0 + p_1 x + p_2 x^2 + \cdots + p_L x^L}{1 + q_1 x + q_2 x^2 + \cdots + q_L x^M}. \tag{28}$$

Moreover, $R_{L,M}$ is the Padé approximation of $u(x)$ if its value and derivatives coincide with those of $u(x)$ at $x = 0$, that is

$$R_{L,M}(0) = u(0), \tag{29}$$
$$R'_{L,M}(0) = u'(0), \tag{30}$$
$$R''_{L,M}(0) = u''(0), \tag{31}$$
$$\vdots$$
$$R^{(L+M)}_{L,M}(0) = u^{(L+M)}(0). \tag{32}$$

Therefore, in approximating $u(x) \approx R_{L,M}(x)$, the rational function has the initial value and derivatives as the main function. Also, far field behavior may be controlled with degrees of $R_{L,M}$.
Let us examine the method on a famous problem in fluid dynamics (see [76]):

$$u''' + uu'' - \beta u'^2 - Mu' = 0, \tag{33}$$
$$u(0) = 0, \quad u'(0) = 1, \tag{34}$$
$$u'(+\infty) = 0. \tag{35}$$

Clearly, by the application of DTM, a polynomial approximating the solution in the vicinity of $x = 0$ will result. However, the polynomial does not have marginal behavior at infinity as required by (35). To cope with the problem, after finding the DTM solution $u^n(x)$, a Padé approximation with $L+M = n$ is obtained. Therefore, the following relation should be occurred:

$$U(0)+U(1)x+U(2)x^2+\cdots+U(n)x^n = \frac{p_0 + p_1 x + p_2 x^2 + \cdots + p_L x^L}{1 + q_1 x + q_2 x^2 + \cdots + q_L x^M}. \tag{36}$$

Two initial conditions will translate to $U(0) = 0$ and $U(1) = 1$, however the degree of equation requires another initial condition. Therefore, $U(2)$ is taken as an unknown α, which will be determined from $u'(+\infty) = 0$ after finding the rational approximation. Then $L + M$ unknown coefficients a_0, a_1, ...,

a_L, b_1, b_2, …, b_M will be found by equating two sides up to x^{L+M}.

Studying the results of this method in [78] in detail shows that the method does not lead to valid solutions easily. For example, the following Padé-DTM solution is claimed in [76] for $\beta = 1.5$ and $M = 50$:

$$\begin{aligned} R_{10,10}(x) = (&x - 22.6935x^2 - 20.8798x^3 - 31.0628x^4 - 19.2098x^5 \\ &-0.841719x^6 + 16.6574x^7 + 1.82323x^8 + 5.78142x^9 \\ &-0.00715648x^{10})/(1 - 19.1112x - 97.9261x^2 - 202.307x^3 \\ &-222.828x^4 - 119.697x^5 + 11.8529x^6 + 70.5985x^7 + 55.4051x^8 \\ &+22.1859x^9 + 4.17935x^{10}) \end{aligned}$$

If we draw the above $u(x)$ near the origin with step size larger than 10^{-5}, then the solution agrees with physics as depicted in Figure 7 of [76]. However, when we take distance from $x = 0$, the solution shows different behavior. In Figure 6, the claimed solution is drawn for $0 \leq x \leq 2$ with step size $\Delta x = 0.01$. It has a clear jump near $1 = 1.4$, which is unexpected. Therefore, it cannot be the correct solution. If the step size of the graph is finer, then the amplitude of the jump increases, indicating a singularity in the rational function. When we examine the roots of the denominator, it reveals that it has two real roots, approximately at $x = 0.0423118$ and $x = 1.40647$. The first one is visible when the step size of plotting is smaller than 10^{-5}. Therefore, the resulting solution is not acceptable near the $x = 0$ nor beyond $x = 1$. This is just an example, and there are other examples showing the inefficiency of the Padé-DTM. Because of the following problems, using Padé

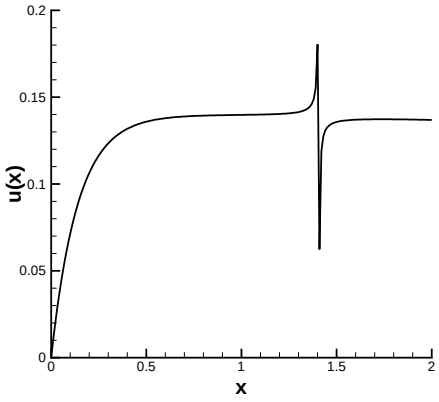


Figure 6: Jump of a Padé-DTM solution.

with DTM is not recommended in general:

- Rational functions may have singularities as indicated in the case study.

- In taking the derivative, the degree of the numerator and denominator will change. This may lead to $R_{L,M}(x) \to 0$ at infinity without obtaining a condition on α. As an example, as indicated in [21], for a Blasius problem, the Padé approximation does not match the required asymptotic behavior.
- Even the marginal condition of u at infinity satisfies, there is no guarantee that the resulting Padé has the same rate of convergence as the exact solution.

3.3.2 Switching DTM

Another DTM-based method for problems with a boundary condition at infinity was proposed in [61]. The method finds a solution consisting of two parts; the first part is a DTM solution, and the second part is a solution of the differential equation satisfying the marginal condition. The method has a successful implementation, but it is case-dependent in finding the second part of the solution.

3.4 Multidimensional DTM

One of the most important and practical extensions of DTM is multidimensional DTM. Let us consider, for example, a two-dimensional BVP or IVP having $u(t,x)$ as the solution. The two-dimensional extension of DTM transform $u(t,x) \xrightarrow{\text{DT}} U(k,h)$ is defined as

$$U(k,h) = \frac{1}{k!h!}\left[\frac{\partial^{k+h}u(t,x)}{\partial t^k x^h}\right]_{(0,0)}, \tag{37}$$

and the inverse transform is

$$u(t,x) = \sum_{k=0}^{\infty}\sum_{h=0}^{\infty} U(k,h)t^k x^h. \tag{38}$$

Substituting (37) into the equations and applying the boundary-initial conditions will result in a set of algebraic equations. Then the equations are solved for $U(k,h)$, and the truncated inverse transform (38) gives an approximate solution. Some of the properties of the two-dimensional DTM transform used to build the algebraic equations are listed below. Assume that $u(t,x) \xrightarrow{\text{DT}} U(k,h)$, that $w(t,x) \xrightarrow{\text{DT}} W(k,h)$, and that λ is a constant scalar.

- $u(t,x) + w(t,x) \xrightarrow{\text{DT}} U(k,h) + W(k,h)$.

- $\lambda u(t,x) \xrightarrow{\text{DT}} \lambda U(k,h)$.
- $u(x)w(x) \xrightarrow{\text{DT}} \sum_{i=0}^{k}\sum_{j=0}^{k} U(i,h-j)W(k-i,j)$.
- $\frac{\partial u(t,x)}{\partial t} \xrightarrow{\text{DT}} (k+1)U(k+1,h)$.
- $\frac{\partial u(t,x)}{\partial x} \xrightarrow{\text{DT}} (h+1)U(k,h+1)$.
- $\frac{\partial^{i+j} u(t,x)}{\partial t^i \partial x^j} \xrightarrow{\text{DT}} (k+1)(k+2)\cdots(k+i)(h+1)(h+2)\cdots(k+j)U(k+i,h+j)$.
- $t^i x^j \xrightarrow{\text{DT}} \delta(k-i,h-j)$.

Now, the two-dimensional DTM in its traditional form is applied to a problem.

Example 3. Let us consider the following IVP defined on the telegraph equation (see [18]):

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2} + 2\frac{\partial u}{\partial t} + u, \tag{39}$$

$$u(0,x) = e^x, \qquad \frac{\partial u}{\partial t}(0,x) = -2e^x. \tag{40}$$

The exact solution is $u(t,x) = e^{x-2t}$.
The corresponding differential transform of (39) is as follows:

$$\begin{aligned}(h+1)(h+2)U(k,h+2) = (k+1)(k+2)U(k+2,h) + 2(k+1)U(k+1,h)\\ +U(k,h).\end{aligned} \tag{41}$$

Taking differential transform from two sides of initial conditions implies that

$$U(0,h) = \frac{1}{h!}, \tag{42}$$

$$U(1,h) = \frac{-2}{h!}. \tag{43}$$

Now, the following recursive relation is obtained to find the coefficients:

$$U(k+2,h) = \frac{(h+1)(h+2)U(k,h+2) - 2(k+1)U(k+1,h) - U(k,h)}{(k+1)(k+2)}. \tag{44}$$

Setting $k = 0,1,2$ in (44) and then $h = 0,1,\ldots,4$ in the results, we obtain the coefficients $U(k,h)$ and construct an approximate solution as

$$\begin{aligned}u_{4,4}(t,x) = 1 + x + 0.5x^2 + 0.1667x^3 + 0.0417x^4\\ -2t - 2tx - tx^2 - 0.3333tx^3 - 0.0833tx^4\\ +2t^2 + 2t^2x + t^2x^2 + 0.3333t^2x^3 + 0.0833t^2x^4\end{aligned}$$

$$-1.3333t^3 - 1.3333t^3x - 0.6667t^3x^2 - 0.2222t^3x^3$$
$$-0.0556t^3x^4 + 0.6667t^4 + 0.6667t^4x + 0.3333t^4x^2$$
$$+0.1111t^4x^3 + 0.0278t^4x^4. \quad (45)$$

The absolute errors of the resulting DTM solution on an 8×8 grid are given in Table 1. The error is small near the initial condition, and the DTM solution approximates the exact solution. However, it grows slowly with t and x. The error may be reduced by increasing the number of terms in (45) since the DTM solution is convergent to the exact one in this case (see [18]).

Table 1: The absolute error of the DTM solution of Example 3

t, x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
0.0	0.0000	0.0000	0.0000	0.0000	0.0002	0.0005	0.0012	0.0027
0.1	0.0000	0.0000	0.0000	0.0000	0.0001	0.0004	0.0010	0.0021
0.2	0.0001	0.0002	0.0002	0.0002	0.0001	0.0001	0.0005	0.0014
0.3	0.0010	0.0011	0.0012	0.0014	0.0015	0.0015	0.0013	0.0008
0.4	0.0040	0.0045	0.0050	0.0056	0.0062	0.0068	0.0073	0.0077
0.5	0.0119	0.0133	0.0148	0.0166	0.0185	0.0205	0.0227	0.0249
0.6	0.0286	0.0320	0.0357	0.0399	0.0446	0.0497	0.0554	0.0615
0.7	0.0599	0.0669	0.0748	0.0836	0.0934	0.1043	0.1163	0.1296

3.4.1 Projected DTM

The projected DTM was introduced in [45] to reduce the computational complexity and simplify the solution in the case of multidimensional DTM. In this approach, the differential transform is applied on only one variable. Therefore, the coefficients are not constant and are functions of the remaining variables. For example, in Example 3, if we take the differential transform with respect to t, then the unknown coefficients are in the form of $U(h, x)$. Consequently, instead of (38), the solution has simpler form as

$$u(t, x) = \sum_{k=0}^{\infty} U(k, x)t^k, \quad (46)$$

which requires lower computational task. If we apply the method to Example 3, then the corresponding equation is changed to

$$\frac{\partial^2 U}{\partial x^2}(k, x) = (k + 1)(k + 2)U(k + 2, x) + 2(k + 1)U(k + 1, x) + U(k, x) \quad (47)$$

with initial conditions:

$$U(0, x) = e^x, \quad (48)$$
$$U(1, x) = -2e^x. \quad (49)$$

The coefficients are also calculated from

$$U(k+2,x)=\frac{\frac{\partial^2 U}{\partial x^2}(k,x)-2(k+1)U(k+1,x)-U(k,x)}{(k+1)(k+2)}. \tag{50}$$

Setting $k=0,1,\ldots$ and using the initial conditions, will result in $U(2,x)=2e^x$, $U(3,x)=-\frac{4}{3}e^x$, $U(4,x)=\frac{1}{6}e^x$, ... Then, the truncated solution up to 4 terms is

$$u_4^p(t,x)=e^x-2e^xt+2e^xt^2-\frac{4}{3}e^xt^3+\frac{1}{6}e^x. \tag{51}$$

From a computational viewpoint, calculating each coefficient in (44) requires 13 elementary operations, while in (44), nine operations are required. On the other hand, the number of terms in $u_{4,4}$ is 4 times than in u_4^p. Therefore, in this case, the projected DTM has lower complexity of order 5.78 with respect to the traditional DTM. However, it should be noted that when estimating the solution at a mesh on (t,x), the computation of e^x terms has more complexity than the power of x but is more accurate.

3.4.2 Reduced DTM

Another approach for simplifying and reducing the computational cost of multidimensional DTM is the reduced DTM proposed in [53]. This modification benefits from a separation of variables. The solution $u(t,x)$ in two-dimensional, for example, is written as

$$u(t,x)=f(t)g(x). \tag{52}$$

Then, one-dimensional DTM is applied, and the corresponding differential transform is obtained similar to the projected DTM.

4 Advantages and disadvantages of DTM

In the previous section, the implementation of DTM on a set of different problems was expressed. Based on the results of these examples and other references, the DTM has advantages and disadvantages as a semi-analytical method for solving initial and boundary value problems. In this section, we mention some of these advantages and disadvantages.

4.1 Advantages of DTM

The advantages of DTM may be encountered as follows:

- The solution has a closed form as a series. This enables us to use it quickly for more analysis, such as calculating derivatives, for example.
- DTM usually results in high-accuracy solutions in the domain of convergence.
- Low computational complexity in solving the transformed equations for linear systems.
- The method does not require discretization; therefore, the results are not affected by this type of error.
- Based on the literature review, the method is flexible to be adopted with various kinds of dynamical systems and boundary conditions.

4.2 Disadvantages of DTM

When using DTM, we have to care about the restrictions of the method. Some of the disadvantages of this method that restricts its application are listed below:

- The implementation of the method for nonlinear systems may lead to complex forms of the algebraic system of equations that restrict the implementation of the method to linear systems. There are, however, some approaches, such as the polynomial expansion of nonlinear terms or using Adomian decomposition. Such tricks may reduce the degree of nonlinearity, however, they add additional errors and increase computational complexity.
- The domain of convergence is usually small, and the results are valid close to $x = 0$. The multi-step DTM resolves this problem relatively. However, as inferred from Figure 5, the multi-step solution itself leads to accumulated errors that show the limited use of the method in short ranges.
- Documented efforts to extend DTM to infinite horizon problems, such as Padé approximation and switching DTM, do not guarantee valid and general solutions.

5 Concluding remarks

The method of differential transform was described and reviewed in this paper. Progress in the implementation, application, and improvements of DTM was expressed. The method gives an analytical solution that has advantages

in comparison with the numerical methods for boundary/initial value problems. However, detailed investigations showed that the method has convergence restrictions. Indeed, when using DTM, it is important to note that the solution is accurate in an interval close to the initial conditions.

References

1. Abazari, R. and Abazari, M. *Numerical study of Burgers–Huxley equations via reduced differential transform method.* Comput. Appl. Math. 32 (2013) 1–17.

2. Abazari, R., Kılıçman, A. *Numerical study of two-dimensional Volterra integral equations by RDTM and comparison with DTM.* Abstr. Appl. Anal. 2013 (929478) (2013) 1–10.

3. Abazari, R. and Kılıçman, A. *Application of differential transform method on nonlinear integro-differential equations with proportional delay.* Neural Comput. Appl. 24 (2014) 391–397.

4. Abdulkawi, M. *Solution of Cauchy type singular integral equations of the first kind by using differential transform method.* Appl. Math. Model. 39 (2015) 2107–2118.

5. Abuasad, S., Moaddy, K. and Hashim, I. *Analytical treatment of two-dimensional fractional Helmholtz equations.* J. King Saud Univ. Sci. 31 (2019) 659–666.

6. Ahmad I. an Ahmad, F., Raja, M.A.Z., Ilyas, H., Anwar, N and Azad, Z. I*ntelligent computing to solve fifth-order boundary value problem arising in induction motor models.* Neural Comput. Appl. 29 (7) (2018) 449–466.

7. Allahviranloo, T., Kiani N.A. and Motamedi, N. *Solving fuzzy differential equations by differential transformation method.* Inf. Sci. 179 (2009) 956–966.

8. Alomari, A.K. *A new analytic solution for fractional chaotic dynamical systems using the differential transform method.* Comput. Math. with Appl. 61 (2011) 2528–2534.

9. Arikoglu, A. and Ozko, I. *Solution of boundary value problems for integro-differential equations by using differential transform method.* Appl. Math. Comput. 168 (2005) 1145–1158.

10. Arikoglu, A. and Ozko, I. *Solution of difference equations by using differential transform method.* Appl. Math. Comput. 174 (2006) 1216–1228.

11. Arikoglu, A. and Ozko, I. *Solution of differential–difference equations by using differential transform method.* Appl. Math. Comput. 181 (2006) 153–162.

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

13. Arvin, H. *The flapwise bending free vibration analysis of micro-rotating timoshenko beams using the differential transform method.* J. Vib. Control 24 (20) (2018) 4868–4884.

14. Ayaz, F. *On the two-dimensional differential transform method.* Appl. Math. Comput. 143 (2003) 361–374.

15. Ayaz, F. *Applications of differential transform method to differential-algebraic equations.* Appl. Math. Comput. 152 (2004) 649–657.

16. Ayaz, F. *Solutions of the system of differential equations by differential transform method.* Appl. Math. Comput. 147 (2004) 547–567.

17. Benhammouda, B., Vazquez-Leal, H. and Sarmiento-Reyes, A. *Modified reduced differential transform method for partial differential-algebraic equations.* J. Appl. Math. (2014) 2014.

18. Biazar, J. and Eslami, M. *Analytic solution for telegraph equation by differential transform method.* Phys. lett. A 374, (2010) 2904–2906.

19. Biswas, S. and Roy, T.K. *Generalization of Seikkala derivative and differential transform method for fuzzy Volterra integro-differential equations.* J. Intell. Fuzzy Syst. 34 (2018) 2795–2806.

20. Bona, J.L., Ponce, G., Saut, J.C. and Sparber, C. *Dispersive blow-up for nonlinear Schrödinger equations revisited.* J. Math. Pures Appl. 102, (2014) 782–811.

21. Boyd, J. *Pade approximant algorithm for solving nonlinear ordinary differential equation boundary value problems on an unbounded domain.* Comput. phys 11 (3) (1997) 299–303.

22. Chakraverty, S., Mahato, N., Karunakar, P. and Rao, T.D. *Adomian decomposition method,* pp. 119–130. Wiley, 2019.

23. Chen, C.K. and Ho, S.H. *Application of differential transformation to eigenvalue problems.* Appl. Math. Comput. 79 (1996) 173–188.

24. Chen, C.K. and Ho, S.H. *Solving partial differential equations by two-dimensional differential transform method.* Appl. Math. Comput. 106 (1999) 171–179.

25. Chen, X. and Dai, Y. *Differential transform method for solving Richards equation.* Appl. Math. Mech. 37 (2016) 169–180.

26. Cimatti, G. *A nonlinear elliptic boundary value problem relevant in general relativity and in the theory of electrical heating of conductors.* Bolletino dell Unione Mat. Ital. 11 (2) (2018) 191–204.

27. Di Matteo, A. and Pirrotta, A. *Generalized differential transform method for nonlinear boundary value problem of fractional order.* Commun. Nonlinear Sci. Numer. Simul. 29 (2015) 88–101.

28. Dutta, J. and Kundu, B. *Thermal analysis on variable thickness absorber plate fin in flatplate solar collectors using differential transform method.* J. Therm. Eng. 6 (2020) 157–169.

29. Elsaid, A. *Fractional differential transform method combined with the Adomian polynomials.* Appl. Math. Comput. 218 (2012) 6899–6911.

30. Elsaid, A. and Helal, S.M. *A new algorithm for computing the differential transform in nonlinear two-dimensional partial differential equations.* J. King Saud Univ. Sci. 32 (2020) 858–861.

31. Erturk, V.S., Momani, S. and Odibat, Z. *Application of generalized differential transform method to multi-order fractional differential equations.* Commun. Nonlinear Sci. Numer. Simul. 13 (2008) 1642–1654.

32. Erturk, V.S. and Momani, S. *Solving systems of fractional differential equations using differential transform method.* Journal of Comput. Appl. Math. 215, (2008) 142–151.

33. Farhatnia, F., Ghanbari-Mobarakeh, M., Rasouli-Jazi, S. and Oveissi, S. *Thermal buckling analysis of functionally graded circular plate resting on the Pasternak elastic foundation via the differential transform method.* Facta Univ. Ser.: Mech. Eng. 15 (2017) 545–563.

34. Fatoorehchi, H. and Abolghasemi, H. *Improving the differential transform method: A novel technique to obtain the differential transforms of nonlinearities by the Adomian polynomials.* Appl. Math. Model. 37 (2013) 6008–6017.

35. Fatoorehchi, H., Abolghasemi, H. and Magesh, N. *The differential transform method as a new computational tool for Laplace transforms.* Natl. Acad. Sci. Lett. 38 (2015) 157–160.

36. Ganji, S.S., Barari, A., Ibsen, L.B. and Domairry, G. *Differential transform method for mathematical modeling of jamming transition problem in traffic congestion flow.* Cent. Eur. J. Oper. 20 (2012) 87–100.

37. Greenberg, M.D. *Applications of Green's functions in science and engineering.* Dover Publications, 2015.

38. Fatoorehchi, H. and Abolghasemi, H. *An explicit analytic solution to the Thomas-Fermi equation by the improved differential transform method.* Acta Phys. Pol. A 125 (2014) 1083–1087.

39. Hamada, Y.M. *Solution of a new model of fractional telegraph point reactor kinetics using differential transformation method.* Appl. Math. Model. 78 (2019) 297–321.

40. Hamza-Cherif, R., Meradjah, M., Zidour, M., Tounsi, A., Belmahi, S. and Bensattalah, T. *Vibration analysis of nano beam using differential transform method including thermal effect.* J. Nano Res. 54 (2018) 1–14.

41. Hesam, S., Nazemi, A.R., and Haghbin, A. *Analytical solution for the fokker–planck equation by differential transform method.* Sci. Iran. 19 (2012) 1140–1145.

42. Hetmaniok, E., Pleszczynski, M. and Khan, Y. *Solving the integral differential equations with delayed argument by using the DTM method.* Sensors 22 (11), (2022) 1–21.

43. Hwang, I., Li, J. and Du, D. *Differential Transformation and Its Application to Nonlinear Optimal Control.* J. Dyn. Syst. Meas. Control, 131(5) (2009) 051010.

44. Ida, N. *Boundary value problems: Analytic methods of solution,* pp. 231–288. Springer International Publishing, Cham, 2015.

45. Jang, B. *Solving linear and nonlinear initial value problems by the projected differential transform method.* Comput. Phys. Commun. 181 (2010) 848–854.

46. Jayaprakash, M., Alzahrani, H., Sowmya, G., Varun Kumar, R., Malik, M., Alsaiari, A. and Prasannakumara, B. *Thermal distribution through a moving longitudinal trapezoidal fin with variable temperature-dependent thermal properties using DTM-Pade approximant.* Case Stud. Therm. Eng. 28 (2021) 101697.

47. Jena, S.K. and Chakraverty, S. *Differential quadrature and differential transformation methods in buckling analysis of nanobeams.* Curved Layer. Struct. 6 (2019) 1629–1641,

48. Kangalgil, F. and Ayaz, F. *Solitary wave solutions for the kdv and mkdv equations by differential transform method.* Chaos Solit. Fractals 41 (2009) 464–472.

49. Kassem, M.A., Hemeda, A.A. and Abdeen, M.A. *Solution of the tumor-immune system by differential transform method.* J. Nonlinear Sci. Appl. 13 (1) (2020) 9–21.

50. Kazemi Gelian, G., Ghoochani Shirvan, R. and Fariborzi Araghi, M.A. *Comparison between sinc approximation and differential transform methods for nonlinear Hammerstein integral equations.* Abstr. Appl. Anal. 13(1) (2022) 1291–1301.

51. Keller, H.B. *Finite-difference methods,* pp. 103–144. Dover publications, New York, 2018.

52. Keskin, A.U. *The shooting method for the solution of one-dimensional BVPs,* pp. 167–258. Springer International Publishing, Cham, 2019.

53. Keskin, Y. and Oturanc, G. *Reduced differential transform method for partial differential equations.* Int. J. Nonlinear Sci. Numer. Simul. 10 (2009) 741–750.

54. Kuma, M. *Study of differential transform technique for transient hydromagnetic Jeffrey fluid flow from a stretching sheet.* Nonlinear Engineering, Modeling and Application 9 (1) (2020) 145–155.

55. Li, T. and Lan, H. *New approximation methods for solving elliptic boundary value problems via picard-mann iterative processes with mixed errors.* Bound. Value Probl. 184 (2017) 449–466.

56. Li, Z., Saleem, S., Shafee, A., Chamkha, A.j. and Du, S. *Analytical investigation of nanoparticle migration in a duct considering thermal radiation.* J. Therm. Anal. Calorim. 135 (2019) 1629–1641.

57. Lin, Y., Chang, K.H. and Chen, C.K. *Hybrid differential transform method/smoothed particle hydrodynamics and DT/finite difference method for transient heat conduction problems.* Int. Commun. Heat Mass Transf. 113 (2020) 297–321.

58. Liu, J. and Hou, G. *Numerical solutions of the space- and time-fractional coupled burgers equations by generalized differential transform method.* Appl. Math. Comput. 217 (2011) 7001–7008.

59. Liu, Y., Sun, K., Yao, R. and Wang, B. *Power system time domain simulation using a differential transformation method.* IEEE Trans. Power Syst. 34(5) (2019) 3739– 3748.

60. Bagyalakshmi M. and SaiSundarakrishnan, G. *Tarig projected differential transform method to solve fractional nonlinear partial differential equations.* Boletim da Sociedade Paranaense de Matematica 38 (2020) 23–46.

61. Mehne, H.H. and Esmaeili, M. *Analytical solution to the boundary layer slip flow and heat transfer over a flat plate using the switching differential transform method.* J. Appl. Fluid Mech. 12, (2019) 433–444.

62. Mittal, R.C. and Pandit, S. *Numerical Simulation of Unsteady Squeezing Nanofluid and Heat Flow between two Parallel Plates using Wavelets.* Int. J. Therm. Sci. 118 (2017) 410–422.

63. Mohamed, M.S. and Gepreel, K.L. *Reduced differential transform method for nonlinear integral member of Kadomtsev–Petviashvili hierarchy differential equations.* J. Egypt. Math. Soc. 25(1) (2017) 1–7.

64. Momani, S. and Erturk, V.S. *Solutions of non-linear oscillators by the modified differential transform method.* Comput. Math. Appl.55 (2007) 833–842.

65. Momani, S., Odibat, Z. and Erturk, V.S. *Generalized differential transform method for solving a space- and time-fractional diffusion-wave equation.* Phys. lett. A 370 (2007) 379–387.

66. Nazemi, A., Hesam, S. and Haghbin, A. *An application of differential transform method for solving nonlinear optimal control problems.* Comput. Methods Differ. Equ. 3 (2015) 200–217.

67. Nazemi, A.R., Hesam and S., Haghbin, A. *A fast numerical method for solving calculus of variation problems.* Adv. Model. Optim. 15 (2013) 133–149.

68. Nourazar, S. and Mirzabeigy, A. *Approximate solution for nonlinear duffing oscillator with damping effect using the modified differential transform method.* Sci. Iran. 20 (2013) 364–368.

69. Odibat, Z. and Momani, S. *A generalized differential transform method for linear partial differential equations of fractional order.* Appl. Math. Lett. 21 (2008) 194–199.

70. Odibat, Z.M., Bertelle, C., Aziz-Alaoui, M.A. and Duchamp, G.H.E. *A multi-step differential transform method and application to non-chaotic or chaotic systems.* Comput. Math. Appl. 59 (2010) 1462–1472.

71. Odibat, Z.M., Kumar, S., Shawagfeh, N., Alsaedi, A. and Hayat, T. *A study on the convergence conditions of generalized differential transform method.* Math. Methods Appl. Sci. 40 (2017) 40–48.

72. Ozdemir, O. and Kaya, M.O. *Flapwise bending vibration analysis of a rotating tapered cantilever Bernoulli–Euler beam by differential transform method.* J. Sound Vib. 289 (2006) 413–420.

73. Paripour, M., Karimi, L. and Abbasbandy, S. *Differential transform method for Volterra's population growth model.* Nonlinear Stud. 24(1) (2017) 227–234.

74. Peter, O.J. and Ibrahim, M.O. *Application of differential transform method in solving a typhoid fever model.* International Journal of Mathematical Analysis and Optimization: Theory and Applications 2017 (2017) 250–260.

75. Rashidi, M., Laraqi, N.R. and Sadri, M. *A novel analytical solution of mixed convection about an inclined flat plate embedded in a porous medium using the DTM-Padé.* Int. J. Therm. Sci. 49 (2010) 2405–2412.

76. Rashidi, M.M. *The modified differential transform method for solving mhd boundarylayer equations.* Comput. Phys. Commun. 180 (2009) 2210–2217.

77. Rashidi, M.M. and Erfani, S. *The modified differential transform method for investigating Nano boundary-layers over stretching surfaces.* Int. J. Numer. Methods Heat Fluid Flow, 21 (2011) 864–883.

78. Rashidi, M.M. and Keimanesh, M. *Using differential transform method and Pade approximant for solving MHD flow in a laminar liquid film from a horizontal stretching surface.* Math. Probl. Eng. 2010 (2010) 1–14.

79. Ravi Kanth, A.S.V. and Aruna, K. *Differential transform method for solving the linear and nonlinear Klein–Gordon equation.* Comput. Phys. Commun. 180 (2009) 708– 711.

80. Saberi Nik, H., Effati, S. and Yildirim, A. *Solution of linear optimal control systems by differential transform method.* Neural Comput. Appl. 23 (2013) 1311–1317.

81. Salahshour, S. and Allahviranloo, T. *Application of fuzzy differential transform method for solving fuzzy Volterra integral equations.* Appl. Math. Model. 37 (2013) 1016– 1027.

82. Shah, K., Singh, T. and Kılıçman, A. *Combination of integral and projected differential transform methods for time-fractional gas dynamics equations.* Ain Shams Eng. J. 9 (2018) 1683–1688.

83. Sheikholeslam Noori, S.M., Taeibi Rahni, M. and Shams Taleghani, S.A. *Multiplerelaxation time color-gradient lattice boltzmann model for simulating contact angle in two-phase flows with high density ratio.* Eur. Phys. J. Plus, 134 (2019) 449–466.

84. Sowmya, G., Sarris, I., Vishalakshi, C., Kumar, R. and Prasannakumara, B. *Analysis of transient thermal distribution in a convective–radiative moving rod using two-dimensional differential transform method with multivariate Pade approximant.* Symmetry, 13 (10) (2021) p.1793.

85. Srivastava, V.K., Awasthi, M.K. and Chaurasia, R.K. *Reduced differential transform method to solve two and three dimensional second order*

hyperbolic telegraph equations. J. King Saud Univ. Eng. Sci. 29 (2017) 166–171.

86. Srivastava, V.K., Awasthi, M.K., Chaurasia, R.K. and Tamsir, M. *The telegraph equation and its solution by reduced differential transform method.* Model. Simul. Eng. 15 (2013) 545–563.

87. Tari, A., Rahimi, M.Y., Shahmorad, S. and Talati, F. *Solving a class of two-dimensional linear and nonlinear Volterra integral equations by the differential transform method.* J. Comput. Appl. Math. 228 (2009) 70–76.

88. Unal, E. and Gökdoğan, A. *Solution of conformable fractional ordinary differential equations via differential transform method.* Optik 128 (2017) 264–273.

89. Varsoliwala, A. and Singh, T. *Analysis of fish farm model by differential transform method.* In Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), pp. 371–379. Amity University Rajasthan, Jaipur, India, 2019.

90. Villafuerte L. and Chen-Charpentier, B.M. *A random differential transform method: Theory and applications.* Appl. Math. Lett. 25 (2012) 1490–1494.

91. Xie, L., Zhou, C. and Xu, S. *An effective numerical method to solve a class of nonlinear singular boundary value problems using improved differential transform method.* SpringerPlus 5(1) (2016) 1–19.

92. Xie, L.J., Zhou, C.L. and Xu, S. *A new algorithm based on differential transform method for solving multi-point boundary value problems.* Int. J. Comput. Math. 93(6) (2016) 981–994.

93. Yang, X.J., Tenreiro Machado, J.A. and Srivastava, H.M. *A new numerical technique for solving the local fractional diffusion equation: Two-dimensional extended differential transform approach.* Appl. Math. Comput. 274 (2016) 143–151.

94. Yu, L.T. and Chen, C.K. *The solution of the Blasius equation by the differential transformation method.* Math. Comput. Modelling, 28 (1998) 101–111.

95. Zhou, J.K. *Differential transformation and its applications for electrical circuits (in Chinese).* Huazhong Univ. Press, 1986.

How to cite this article
H.H. Mehne Differential transform method: A comprehensive review and analysis. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 629-657. doi: 10.22067/ijnao.2022.77130.1153.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 658–679
DOI:10.22067/ijnao.2022.78966.1186
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Global and extended global Hessenberg processes for solving Sylvester tensor equation with low-rank right-hand side

T. Cheraghzadeh, F. Toutounian* , and R. Khoshsiar Ghaziani

Abstract

In this paper, we introduce two new schemes based on the global Hessenberg processes for computing approximate solutions to low-rank Sylvester tensor equations. We first construct bases for the matrix and extended matrix Krylov subspaces by applying the global and extended global Hessenberg processes. Then the initial problem is projected into the matrix or extended matrix Krylov subspaces with small dimensions. The reduced Sylvester tensor equation obtained by the projection methods can be solved by using a recursive blocked algorithm. Furthermore, we present the upper bounds for the residual tensors without requiring the computation of the approximate solutions in any iteration. Finally, we illustrate the performance of the proposed methods with some numerical examples.

AMS subject classifications (2020): 65F10.

Keywords: Low-rank Sylvester tensor equation; Global Hessenberg process; Extended Global Hessenberg process; CP decomposition.

---

* Corresponding author

Received 27 September 2022; revised 29 October 2022; accepted 31 October 2022

T. Cheraghzadeh
Department of Applied Mathematics, Faculty of Mathematical Science, Shahrekord University, Shahrekord, Iran. e-mail: heraghzadeh@stu.sku.ac.ir

F. Toutounian
Department of Applied Mathematics, Faculty of Mathematical Science, The Center of Excellence on Modeling and Control Systems, Ferdowsi University of Mashhad, Iran. e-mail: toutouni@math.um.ac.ir

R. Khoshsiar
Department of Applied Mathematics, Faculty of Mathematical Science, Shahrekord University, Shahrekord, Iran. e-mail: Khoshsiar@sku.ac.ir

1 Introduction

Let $I_1, I_2, \ldots, I_N \in \mathbb{N}$. The multidimensional array $\mathcal{X} = (\mathcal{X}_{i_1 i_2 \cdots i_N})(1 \leq i_j \leq I_j, j = 1, \ldots, N)$ is called an N-mode tensors with $I_1 I_2 \cdots I_N$ entries. There has been increasing research on tensors in recent years. For instance, Chang, Pearson, and Zhang [8] generalized the Perron–Frobenius theorem for nonnegative matrices to the nonnegative tensors. Eigenvalues, eigenvectors, symmetric hyperdeterminants were defined by Qi [31] for the real supersymmetric tensors, and their properties were described. In [30], the restart techniques are described for the tensor infinite Arnoldi method.

In this work, we introduce two new projection methods for solving the low-rank Sylvester tensor equation

$$\mathcal{X} \times_1 A^{(1)} + \mathcal{X} \times_2 A^{(2)} + \cdots + \mathcal{X} \times_N A^{(N)} = \mathcal{B}, \tag{1}$$

where the matrices $A^{(n)} \in \mathbb{R}^{I_n \times I_n}$, $n = 1, 2, \ldots, N$, and right-hand side tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ are given, and $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an unknown tensor. The Sylvester tensor equation (1) has a unique solution if and only if $\lambda_1 + \lambda_2 + \cdots + \lambda_N \neq 0$, for all $\lambda_i \in \sigma(A^{(i)})$, $i = 1, 2, \ldots, N$, where $\sigma(A^{(i)})$ is the spectral of matrix $A^{(i)}$ [9]. In this study, it is assumed that the Sylvester tensor equation has a unique solution. The Sylvester tensor equations are one of the famous problems arising from the discretization of a linear partial differential equation in high dimensions by the use of finite elements, finite differences, and spectral methods [27, 28, 37]. The Sylvester matrix equation

$$A^{(1)} X + X A^{(2)T} = B,$$

is a special case of the Sylvester tensor equation (1), where X is a 2-mode tensor. Many iteration methods for computing approximate solutions for the Sylvester tensor equations (1) have been introduced in recent years. For example, Chen and Lu [9] proposed the GMRES method based on tensor form (GMRES-BTF) to solve the Sylvester tensor equation. Also, to speed up the convergence of the GMRES-BTF method, they proposed preconditioned GMRES-BTF. Beik, Saberi Movahed, and Ahmadi-Asl [4] presented some iterative methods based on the tensor format to solve the Sylvester tensor equations (1). In [33, 34], Saberi–Movahed et al. introduced the tensor format of restarted Simpler GMRES, (SGMRES-BTF(m)), to solve the Sylvester tensor equation and described an accelerating method in accordance with a modification of the generalized conjugate residual with inner orthogonalization (GCRO) method based on the tensor format. Bi-conjugate gradient (BiCG) and bi-conjugate residual (BiCR) methods as well as their preconditioned versions based on the tensor format, have been presented in [39]. The tensor form of the global least squares method is proposed in [24]. Huang, Xie, and Ma [22] proposed the tensor form of the GMRES method for solving a class of tensor equations via the Einstein product. Furthermore,

for the case in which the coefficient tensor is symmetric, they proposed the MINRES and SYMMLQ methods based on the tensor format. Dehdezi and Karimi [15] extended the conjugate gradient squared and the conjugate residual squared methods to solve the generalized coupled Sylvester tensor equations. In [16], the authors proposed a gradient based iterative method version for solving the tensor equations and presented a new preconditioner to accelerate the convergence rate of the proposed iterative methods. A projection method has been introduced in [3] to find approximations of linear systems in low-rank tensor format. Kressner and Tobler [25] proposed the Krylov subspace for the case in which the right-hand side tensor has a low-rank. Recently, Bentbib, El-Halouy, and Sadek [5] introduced a new projection method to compute approximate solutions for the low-rank Sylvester tensor equations. The extended Krylov-like methods were proposed in [6] to find the solutions for the low-rank Sylvester and Stein tensor equations. The block and extended block Hessenberg algorithms for solving the Sylvester tensor equation with low-rank right-hand side (1) were presented in [12]. Hessenberg based methods are among the popular methods in terms of the Krylov subspace methods, with less need for arithmetic operations and less storage space compared to the Arnoldi-based methods. The Hessenberg process constructs nonorthogonal bases for the associated Krylov subspace. The schemes based on the Hessenberg process have recently received great attention; see, for instance, [32, 35, 19, 17, 21, 12]. This motivated us to introduce two new projection schemes, employing the global Hessenberg process on the matrix Krylov subspaces. The main idea of this scheme is to project the problem onto a matrix or an extended matrix Krylov subspace. Then the reduced problem can be solved by using the recursive blocked algorithm [11]. Complexity consideration is given to show that the global and extended global Hessenberg processes are less expensive than the global and extended global Arnoldi ones.

We use the following notations. For the matrices X and Y in $\mathbb{R}^{n\times n}$, we consider the following inner product $\langle X, Y\rangle_F = \mathrm{tr}(X^TY)$, where $\mathrm{tr}(\cdot)$ denotes the trace. The associated norm is the Frobenius norm denoted by $\|E\|_F$. The notation $X \perp_F Y$ means that $\langle X, Y\rangle_F = 0$. The $n \times n$ identity matrix is denoted by $I^{(n)}$. Moreover, $e_j^{(k)}$ denotes the jth canonical vector of $\mathbb{R}^k$, and $0_{m\times n}$ denotes the $m \times n$ zero matrix.

The remainder of this paper is organized as follows. In section 2, we review some basic notations and definitions. In section 3, the global Hessenberg process with maximum strategy and an approach for solving (1) with a right-hand side tensor of a specific rank is described. The extended global Hessenberg approach is presented in section 4. The complexity of the new methods is considered in section 5. Some numerical examples for evaluating the performance of our approaches are given in section 6. Finally, section 7 gives a brief conclusion.

2 Preliminaries

In this part, the notations and basic definitions of tensors are presented. Throughout this paper, we denote tensors by Euler script letters. Matrices and vectors are denoted by capital and lowercase letters, respectively. Also, the Kronecker product of matrices A and B is denoted by $A \otimes B$ and the Kronecker product of tensors $\mathcal{A}$ and $\mathcal{B}$, is denoted by $\mathcal{A} \otimes \mathcal{B}$. Norm of an Nth order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\|\mathcal{X}\|_F$ and is defined as follows:

$$\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{X}_{i_1 i_2 \cdots i_N}^2}.$$

Definition 1 ([13]). Denote the N-mode (matrix) product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $U \in \mathbb{R}^{J \times I_n}$ by $\mathcal{X} \times_n U$. It is of dimension $I_1 \times I_2 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ and defined as

$$(\mathcal{X} \times_n U)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1 i_2 \cdots i_N} u_{j i_n}.$$

Proposition 1 ([13]). Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an Nth order tensor, let $B \in \mathbb{R}^{J \times I_m}, C \in \mathbb{R}^{K \times I_n}$, and let $W \in \mathbb{R}^{I_n \times I_n}$. Then

$$\mathcal{A} \times_m B \times_n C = \mathcal{A} \times_n C \times_m B,$$
$$\mathcal{A} \times_n W \times_n C = \mathcal{A} \times_n CW.$$

Definition 2 ([14]). Assume that $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an Nth order tensor and that $\{U\}$ is a set of matrices $U_n \in \mathbb{R}^{I_n \times I_n} (n = 1, \ldots, N)$. Then their product in all possible modes $(n = 1, 2, \ldots, N)$ is of size $I_1 \times I_2 \times \cdots \times I_N$ and defined as follows:

$$\mathcal{X} \times \{U\} = \mathcal{X} \times_1 U_1 \times_2 U_2 \cdots \times_N U_N,$$

and

$$\mathcal{X} \times \{U\}^T = \mathcal{X} \times_1 U_1^T \times_2 U_2^T \cdots \times_N U_N^T.$$

Definition 3 ([13]). . The outer product of two tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ and $\mathcal{B} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is denoted by $\mathcal{A} \circ \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M \times J_1 \times J_2 \times \cdots \times J_N}$, with entries

$$\mathcal{C}_{i_1 \cdots i_M j_1 \cdots j_N} = \mathcal{A}_{i_1 \cdots i_M} \mathcal{B}_{j_1 \cdots j_N}.$$

If $v_1, v_2, \ldots, v_N$ are N vectors of sizes $I_i, i = 1, \ldots, N$, then their outer product is an Nth order tensor of size $I_1 \times I_2 \times \cdots \times I_N$ and is given by

$$v_1 \circ \cdots \circ v_{Ni_1,\ldots,i_N} = v_1(i_1) \cdots v_N(i_N).$$

Definition 4 ([13]). An Nth order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is called a rank one tensor if it can be written as the outer product of N vectors $a_i \in \mathbb{R}^{I_i}$ $(i = 1, \ldots, N)$ as follows:

$$\mathcal{X} = a_1 \circ a_2 \circ \cdots \circ a_N.$$

If a tensor can be written as a sum of R rank one tensors, then it is called a rank R tensor.

Definition 5 ([26]). The Kronecker product of two tensor $\mathcal{A} = a_1 \circ a_2 \circ \cdots \circ a_N$ and $\mathcal{B} = b_1 \circ b_2 \circ \cdots \circ b_N$ is defined as

$$\mathcal{A} \otimes \mathcal{B} = (a_1 \otimes b_1) \circ \cdots \circ (a_N \otimes b_N).$$

Proposition 2 ([5]). Assume that $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ are Nth order tensors, that $A \in \mathbb{R}^{k_n \times I_n}$, and that $B \in \mathbb{R}^{I_n \times J_n}$. Then

$$(\mathcal{A} \otimes \mathcal{B}) \times_n (A \otimes B) = (\mathcal{A} \times_n A) \otimes (\mathcal{B} \times_n B).$$

Proposition 3 ([5]). The product of a rank one tensor $\mathcal{A} = a_1 \circ a_2 \circ \cdots \circ a_N \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a set of matrices $U_n \in \mathbb{R}^{I_n \times I_n}, (n = 1, \ldots, N)$ is defined as follows:

$$\mathcal{A} \times \{U\} = U_1 a_1 \circ \cdots \circ U_N a_N. \tag{2}$$

Definition 6 ([13]). The CP decomposition of an Nth order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is written as follows:

$$\mathcal{A} = \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(N)},$$

where $R \in \mathbb{N}$ and $a_r^{(i)} \in \mathbb{R}^{I_i}$, $(i = 1, \ldots, N)$. Assume that $a_r^{(i)}$, $(i = 1, \ldots, N)$, are normalized. Then the CP decomposition is given by

$$\mathcal{A} = \sum_{r=1}^{R} \lambda_r a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(N)},$$

where $\lambda_r \in \mathbb{R}$.

Definition 7 (Left inverse[35]). Consider $Z_k \in \mathbb{R}^{n \times k}$ as a matrix partitioned as follows:

$$Z_k = \begin{bmatrix} Z_{1k} \\ Z_{2k} \end{bmatrix},$$

where Z_{1k} is a $k \times k$ matrix. If the matrix Z_{1k} is nonsingular, then a left inverse of Z_k is defined as follow

$$Z_k^L = \left[Z_{1k}^{-1}, 0_{k\times(n-k)}\right].$$

Definition 8 ([7]). Let $A = [A_1, A_2, \ldots, A_p]$ and $B = [B_1, B_2, \ldots, B_l]$ be matrices of dimension $n \times ps$ and $n \times ls$, respectively, where A_i and B_j $(i = 1, \ldots, p; j = 1, \ldots, l)$ are $n \times s$ matrices. Then the $\diamond$-product of matrices A and B denoted by $A^T \diamond B$ is the $p \times l$ matrix defined by:

$$(A^T \diamond B)_{i,j} = \langle A_i, B_j \rangle_F.$$

Some properties that are verified by the $\otimes$- and $\diamond$-products are as follows:

1. $(DA)^T \diamond B = A^T \diamond (D^T B)$.
2. $A^T \diamond (B(L \otimes I^{(p)}) = (A^T \diamond B)L$.

In what follows, we assume that the right-hand side $\mathcal{B}$ in (1) is of rank R. As known [13], by using the CP decomposition, $\mathcal{B}$ can be written as

$$\mathcal{B} = \sum_{r=1}^{R} b_1^{(r)} \circ \cdots \circ b_N^{(r)}, \tag{3}$$

where $B^{(i)} = \left[b_i^{(1)}, b_i^{(2)}, \ldots, b_i^{(R)}\right] \in \mathbb{R}^{I_i \times R}, i = 1, \ldots, N$, are the factor matrices. By simple calculations, we can rewrite the relation (3) as

$$\mathcal{B} = \mathcal{I}_R \times_1 B^{(1)} \cdots \times_N B^{(N)}, \tag{4}$$

in which $\mathcal{I}_R$ denotes the identity tensor of Nth order of size $R \times R \times \cdots \times R$ with ones along the super-diagonal.

3 Global Hessenberg process with maximum strategy

The global Hessenberg process with maximum strategy was first presented in [17] by Heyouni to build a basis of the matrix Krylov subspace

$$\mathcal{K}_m(A, V) = \left\{ \sum_{i=0}^{m-1} \gamma_i A^i V, \quad \text{where } \gamma_i \in \mathbb{R} \text{ for } i = 0, 1, \ldots, m-1 \right\},$$

where $A \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times s}$. The global Hessenberg process with maximum strategy can be summarized in Algorithm 1 [17].

By employing Algorithm 1 with $m = m_i$ and $s = R$ for the pair $(A^{(i)}, B^{(i)})$, we obtain $\mathbb{V}_{m_i+1} = [V_1^{(i)}, \ldots, V_{m_i+1}^{(i)}] \in \mathbb{R}^{n\times(m_i+1)R}$ with $V_k^{(i)} \in \mathbb{R}^{n\times R}$, for $k = 1, \ldots, m_i+1$, and the upper Hessenberg matrix $\bar{H}_{m_i} = (h_{i,j}^{(i)}) \in \mathbb{R}^{(m_i+1)\times m_i}$, which satisfy

$$A^{(i)}\mathbb{V}_{m_i} = \mathbb{V}_{m_i+1}(\bar{H}_{m_i} \otimes I^{(R)}), \tag{5}$$

Algorithm 1 The Global Hessenberg process with Maximum Strategy

1. **Input:** Nonsingular matrix A, initial block V, and an integer m.
2. Determine i_0 and j_0 such that $|V_{i_0,j_0}| = \max\{|V_{i,j}|\}_{1\le i\le n}^{1\le j\le s}$; $\beta = V_{i_0,j_0}$; $V_1 = V/\beta$; $l_1 = i_0$; $c_1 = j_0$.
3. For $k = 1, 2, \ldots, m$
4. $U = AV_k$.
5. For $j = 1, 2, \ldots, k$
6. $h_{j,k} = U_{l_j,c_j}$; $U = U - h_{j,k}V_j$.
7. End For.
8. Determine i_0 and j_0 such that $|U_{i_0,j_0}| = \max\{|U_{i,j}|\}_{1\le i\le n}^{1\le j\le s}$ $h_{k+1,k} = U_{i_0,j_0}$; $V_{k+1} = U/h_{k+1,k}$; $l_{k+1} = i_0$; $c_{k+1} = j_0$.
9. End For.

$$= \mathbb{V}_{m_i}(H_{m_i} \otimes I^{(R)}) + h^{(i)}_{m_i+1,m_i} V^{(i)}_{m_i+1}(e^{(m_i)^T}_{m_i} \otimes I^{(R)}), \tag{6}$$

where H_{m_i} denotes the matrix obtained from $\bar{H}_{m_i}$ by deleting its last row. As [5], we consider an approximate solution of (1) as

$$\mathcal{X}_m = (\mathcal{Y}_m \otimes \mathcal{I}_R) \times \{\mathbb{V}_m\}, \tag{7}$$

where $\{\mathbb{V}_m\}$ denotes a set of matrices $\{\mathbb{V}_{m_1}, \mathbb{V}_{m_2}, \ldots, \mathbb{V}_{m_N}\}$ and $\mathcal{Y}_m$ is an $m_1 \times \cdots \times m_N$ tensor satisfying the low-dimensional Sylvester tensor equation

$$\sum_{i=1}^{N} \mathcal{Y}_m \times_i H_{m_i} = \beta \mathcal{E}_m, \tag{8}$$

where $\beta = \prod_{i=1}^{N} \beta_i$ and $\mathcal{E}_m = (e_1^{(m_1)} \circ \cdots \circ e_1^{(m_N)})$.

Proposition 4. Let $\mathcal{R}_m$ be the residual tensor corresponding to the approximate solution $\mathcal{X}_m$ of (1). Then

$$\mathcal{R}_m = -\sum_{i=1}^{N} h_{m_i+1,m_i}(\mathcal{Y}_m \times_i e^{(m_i)^T}_{m_i}) \otimes \mathcal{I}_R \times_1 \mathbb{V}_{m_1} \cdots \times_i V^{(i)}_{m_i+1} \cdots \times_N \mathbb{V}_{m_N}, \tag{9}$$

where $\mathcal{Y}_m$ is the solution to (8).

Proof. The proof is similar to that of Proposition 6 in [12]. □

Theorem 1. Let $\mathcal{X}_m$ be an approximate solution of (1). Then the corresponding residual $\mathcal{R}_m$ satisfies

$$\|\mathcal{R}_m\| \le \sqrt{((2nR - (m-1))\frac{m}{2})^N} \sqrt{\sum_{i=1}^{N} |\, h_{m_i+1,m_i}\,|^2 \, \|\mathcal{Y}_m \times_i e^T_{m_i}\|^2}, \tag{10}$$

where $m = \max_{1\le i\le N} m_i$.

Proof. The proof is similar to that of Theorem 7 in [12]. □

Furthermore, from the fact that

$$\|\mathbb{V}_{m_j}\|^2 \leq nm_jR, \quad i = 1, \ldots, N,$$

we have

$$\|\mathcal{R}_m\| \leq \sqrt{(nmR)^N}\sqrt{\sum_{i=1}^{N} \mid h_{m_i+1,m_i} \mid^2 \|\mathcal{Y}_m \times_i e_{m_i}^T\|^2}. \tag{11}$$

The upper bounds (10) and (11) are pessimistic. We propose the following approximation, which is derived heuristically,

$$\|\mathcal{R}_m\| \approx E_m := \sqrt[N]{(nmR)}\sqrt{\sum_{i=1}^{N} \mid h_{m_i+1,m_i} \mid^2 \|\mathcal{Y}_m \times_i e_{m_i}^T\|^2}. \tag{12}$$

Similar to Algorithm 2 in [5], the global Hessenberg process with the maximum strategy for the Sylvester tensor equation (1) can be summarized in Algorithm 2.

Algorithm 2

1. **Input**: Coefficient matrices $A^{(i)}, i = 1, \ldots N$, and the right-hand side in low-rank representation,
   $\mathcal{B} = \left[B^{(1)}, B^{(2)}, \ldots, B^{(N)}\right]$.
2. **Output**: An approximate solution $\mathcal{X}_m$ for equation (1).
3. Choose a tolerance $\epsilon > 0$, integer parameters $k_i^{'}$, $i = 1, \ldots, N$. Set $k_i = 0, m_i = k_i^{'}$.
4. For $i = 1 : N$
5. For $j = k_i + 1 : k_i + k_i^{'}$
6. Construct the basis $\left[V_{k_i+1}, \ldots, V_{k_i+k_i^{'}}\right]$ and the matrix $\mathbb{H}_{m_i}$ by Algorithm 1.
7. End For
8. End For
9. Solve the low-dimensional equation $\sum_{i=1}^{N} \mathcal{Y}_m \times_i \mathbb{H}_{m_i} = \beta\mathcal{E}_m$ by the recursive blocked algorithms presented in [11].
10. Compute the estimated residual norm of $\mathcal{R}_m$,
    i.e., $E_m = \sqrt[N]{(nmR)}\sqrt{\sum_{i=1}^{N} \mid h_{m_i+1,m_i} \mid^2 \|\mathcal{Y}_m \times_i e_{m_i}^T\|^2}$.
11. If $E_m > \epsilon$, set $k_i = k_i + k_i^{'}$, $m_i = k_i + k_i^{'}$ for $i = 1, \ldots, N$, and go to step 4.
12. Compute the approximate solution by $\mathcal{X}_m = (\mathcal{Y}_m \otimes \mathcal{I}^{(R)}) \times_1 \mathbb{V}_{m_1} \cdots \times_N \mathbb{V}_{m_N}$.

4 The extended global Hessenberg process

We first recall the extended matrix Krylov subspace. Let $A \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times s}$. The extended global Hesssenberg process corresponding to the pair (A, V) is defined as follows [17]:

$$\begin{aligned}\mathcal{K}_m^e(A,V) &= \text{span}(V, A^{-1}V, AV, \ldots, A^{m-1}V, A^{-m}V),\\ &= \mathcal{K}_m(A,V) + \mathcal{K}_m(A^{-1}, A^{-1}V).\end{aligned}$$

The algorithm proceeds by running one step of the Global Hessenberg process with A and one step with A^{-1}, while maintaining orthogonalization among all generated vectors and the $n \times s$ matrices $Y_j = e_{l_j}^{(n)} e_{c_j}^{(s)^T}$ whose entries are zero except $(Y_j)_{l_j,c_j} = 1$. The first two block vectors $V_1^{(1)}$ and $V_1^{(2)}$ are obtained as follows:

$$V_1^{(1)} = V/r_{11}, \tag{13}$$

where $r_{11} = V_{l_1,c_1}$ and $|V_{l_1,c_1}| = \max\{|V_{i,j}|\}_{1\le i\le n}^{1\le j\le s}$, and

$$V_2^{(2)} = W/r_{2,2}, \tag{14}$$

where $W = A^{-1}V - r_{1,2}V_1^{(1)}, r_{1,2} = (A^{-1}V)_{l_1,c_1}, r_{2,2} = W_{l_2,c_2}$, and $|W_{l_2,c_2}| = \max\{|W_{i,j}|\}_{1\le i\le n}^{1\le j\le s}$.

Let $V_i = [V_i^{(1)}, V_i^{(2)}]$ be the ith $n \times 2s$ block vector of $\mathbb{V}_m = [V_1, \ldots, V_m]$ and let

$$H_{i,j} = \begin{bmatrix} h_{2i-1,2j-1} & h_{2i-1,2j} \\ h_{2i,2j-1} & h_{2i,2j} \end{bmatrix},$$

be the 2×2 block matrix (i, j) of the upper block Hessenberg matrix $\overline{\mathbb{H}}_m \in \mathbb{R}^{2(m+1)\times 2m}$. Then we compute the two block vectors $V_{k+1}^{(1)}$ and $V_{k+1}^{(2)}$ by the relation

$$\left[V_{k+1}^{(1)}\ V_{k+1}^{(2)}\right](H_{k+1,k} \otimes I^{(s)}) = [AV_k^{(1)}, A^{-1}V_k^{(2)}] - \sum_{j=1}^{k} [V_j^{(1)}, V_j^{(2)}](H_{j,k} \otimes I^{(s)}), \tag{15}$$

where the entries of coefficients matrices $H_{k+1,k}$ and $H_{i,k}$, for $i = 1, \ldots, k$, will be determined such that the relations

$$V_{k+1}^{(1)} \perp_F Y_1, \ldots, Y_{2k} \qquad \text{and} \qquad (V_{k+1}^{(1)})_{l_{2k+1},c_{2k+1}} = 1,$$

and

$$V_{k+1}^{(2)} \perp_F Y_1, \ldots, Y_{2k+1} \qquad \text{and} \qquad (V_{k+1}^{(2)})_{l_{2k+2},c_{2k+2}} = 1$$

hold for $k = 1, \ldots, m$. The determination of indices l_{2k+1}, c_{2k+1} and l_{2k+2}, c_{2k+2} is similar to that of indices l_1, c_1 and l_2, c_2, respectively. The main steps of the extended global Hessenberg process algorithm to generate $\mathbb{V}_m$ and $\overline{\mathbb{H}}_m$ may be summarized as follows.

Algorithm 3 The Extended Global Hessenberg process with Maximum Strategy

1. **Input:** Nonsingular matrix A, initial block V, and an integer m.
2. Determine i_0 and j_0 such that $|V_{i_0,j_0}| = \max\{|V_{i,j}|\}_{1\le i\le n}^{1\le j\le s}$; $r_{1,1} = V_{i_0,j_0}$; $V_1^{(1)} = V/r_{1,1}$; $l_1 = i_0$; $c_1 = j_0$; .
3. $W = A^{-1}V$; $r_{1,2} = W_{l_1,c_1}$.
4. $W = W - r_{1,2}V_1^{(1)}$, $|W_{i_0,j_0}| = \max\{|W_{i,j}|\}_{1\le i\le n}^{1\le j\le s}$; $r_{2,2} = W_{i_0,j_0}$; $V_1^{(2)} = W/r_{2,2}$; $l_2 = i_0$, $c_2 = j_0$.
5. For $k = 1, 2, \ldots, m$
6. $W = AV_k^{(1)}$.
7. For $i = 1, \ldots, k$
8. $h_{2i-1,2k-1} = W_{l_{2i-1},c_{2i-1}}$, $W = W - h_{2i-1,2k-1}V_i^{(1)}$; $h_{2i,2k-1} = W_{l_{2i},c_{2i}}$, $W = W - h_{2i,2k-1}V_i^{(2)}$.
9. End For.
10. Determine i_0 and j_0 such that $|W_{i_0,j_0}| = \max\{|W_{i,j}|\}_{1\le i\le n}^{1\le j\le s}$; $h_{2k+1,2k-1} = W_{i_0,j_0}$; $V_{k+1}^{(1)} = W/h_{2k+1,2k-1}$; $l_{2k+1} = i_0$; $c_{2k+1} = j_0$.
11. $W = A^{-1}V_k^{(2)}$.
12. For $i = 1, \ldots, k$.
13. $h_{2i-1,2k} = W_{l_{2i-1},c_{2i-1}}$, $W = W - h_{2i-1,2k}V_i^{(1)}$; $h_{2i,2k} = W_{l_{2i},c_{2i}}$; $W = W - h_{2i,2k}V_i^{(2)}$.
14 End For.
15 $h_{2k+1,2k} = W_{l_{2k+1},c_{2k+1}}$, $W = W - h_{2k+1,2k}V_{k+1}^{(1)}$.
16. Determine i_0 and j_0 such that $|W_{i_0,j_0}| = \max\{|W_{i,j}|\}_{1\le i\le n}^{1\le j\le s}$; $h_{2k+2,2k} = W_{i_0,j_0}$; $V_{k+1}^{(2)} = W/h_{2k+2,2k}$; $l_{2k+2} = i_0$; $c_{2k+2} = j_0$.
17. End For.

Suppose that the matrix $\mathbb{P}_m$ is defined by $[Y_1, Y_2, \ldots, Y_{2m}]$. Then

$$\mathbb{P}_m^T \diamond \mathbb{V}_m = \mathbb{L}_m,$$

where $\mathbb{L}_m \in \mathbb{R}^{2m\times 2m}$ is a unit lower triangular matrix. So, we have $\mathbb{L}_{m_i}^{-1}(\mathbb{P}_{m_i}^T \diamond \mathbb{V}_{m_i}) = I^{(2m_i)}$. As in [1], we consider $\mathbb{V}_m^L = (\mathbb{P}_m(\mathbb{L}_m^{-T} \otimes I^{(s)}))^T = (\mathbb{L}_m^{-1} \otimes I^{(s)})\mathbb{P}_m^T$, as a left inverse for the $\diamond$-product, which verifies the relation $\mathbb{V}_m^L \diamond \mathbb{V}_m = I^{(2ms)}$. Using this matrix, we can state the following proposition.

Proposition 5. Let $\overline{\mathbb{T}}_m = \mathbb{V}_{m+1}^L \diamond (A\mathbb{V}_m)$, and suppose that m steps of Algorithm 3 have been carried out. Then

$$A\mathbb{V}_m = \mathbb{V}_{m+1}(\overline{\mathbb{T}}_m \otimes I^{(s)}), \tag{16}$$

$$= \mathbb{V}_m(\mathbb{T}_m \otimes I^{(s)}) + V_{m+1}(T_{m+1,m}E_m^T \otimes I^{(s)}), \tag{17}$$

where $T_{i,j}$ is the 2×2 block (i,j) of $\mathbb{T}_m$ and $E_m^T=[0_{2\times 2(m-1)}, I^{(2)}]$, and $\mathbb{T}_m$ is obtained by removing the two last rows of $\overline{\mathbb{T}}_m$.

Proof. The proof is similar to the case for the classical Arnoldi process in [20]. □

As [36], in the following proposition, we derive some recursive relations, which can be used to significantly reduce the computational cost of the basic algorithm.

Proposition 6. Let $\overline{\mathbb{T}}_m=[t_{:,1},\ldots,t_{:,2m}]$ and $\overline{\mathbb{H}}_m=[h_{:,1},\ldots,h_{:,2m}]$ be two $2(m+1)\times 2m$ block upper Hessenberg matrices, let $\ell^{(k+1)}=(\ell_{i,j})=H_{k+1,k}^{-1}$, and let $r_{1,1}, r_{1,2}, r_{2,2}$ be as defined in Algorithm 3. Then for the odd columns, we have

$$t_{:,2j-1}=h_{:,2j-1}, \qquad j=1,\ldots,m,$$

and for the even columns, we have

$$\begin{aligned}
(k=1)\qquad t_{:,2} &= \frac{1}{r_{2,2}}(r_{1,1}e_1^{2(m+1)}-r_{1,2}t_{:,1}),\\
t_{:,4} &= (e_2^{2(m+1)}-\begin{bmatrix}\overline{\mathbb{T}}_1 h_{1:2,2}\\ 0_{(2m-2)\times 2}\end{bmatrix})\ell_{22}^{(2)},\\
\rho^{(2)} &= (\ell_{11}^{(2)})^{-1}\ell_{12}^{(2)},\\
(1<k\le m)\qquad t_{:,2k} &= t_{:,2k}+t_{:,2k-1}\rho^{(k)},\\
t_{:,2k+2} &= (e_{2k}^{2(m+1)}-\begin{bmatrix}\overline{\mathbb{T}}_k h_{1:2k,2k}\\ 0_{(2m-2k)\times 2}\end{bmatrix})\ell_{22}^{(k+1)},\\
\rho^{(k+1)} &= (\ell_{11}^{(k+1)})^{-1}\ell_{12}^{(k+1)}.
\end{aligned}$$

Proof. Starting from (15), we have

$$\begin{aligned}
AV_k^{(1)} &= V_{k+1}(H_{k+1,k}e_1^{(2)}\otimes I^{(s)})+\mathbb{V}_k(\mathbb{H}_k e_{2k-1}^{(2k)}\otimes I^{(s)})\\
&= \mathbb{V}_{k+1}(\overline{\mathbb{H}}_k e_{2k-1}^{(2k)}\otimes I^{(s)}).
\end{aligned}$$

Pre-multiplying the above relation by $\mathbb{V}_{m+1}^L$, we get

$$\begin{aligned}
\mathbb{V}_{m+1}^L\diamond AV_k^{(1)} &= \mathbb{V}_{m+1}^L\diamond\mathbb{V}_{k+1}(\overline{\mathbb{H}}_k e_{2k-1}^{(2k)}\otimes I^{(s)})\\
&= (\mathbb{V}_{m+1}^L\diamond\mathbb{V}_{k+1})\overline{\mathbb{H}}_k e_{2k-1}^{(2k)}\\
&= \begin{bmatrix} I^{(2k+2)}\\ 0_{(2m-2k)\times(2k+2)}\end{bmatrix}\overline{\mathbb{H}}_k e_{2k-1}^{(2k)}\\
&= \begin{bmatrix}\overline{\mathbb{H}}_k\\ 0_{(2m-2k)\times(2k+2)}\end{bmatrix}e_{2k-1}^{(2k)}.
\end{aligned}$$

Hence,

$$t_{:,2k-1} = h_{:,2k-1}, \qquad k = 1, \ldots, m.$$

From the lines 2 and 3 of Algorithm 3, we have

$$r_{2,2}V_1^{(2)} = r_{1,1}A^{-1}V_1^{(1)} - r_{1,2}V_1^{(1)}.$$

Pre-multiplying this relation by A, we get

$$r_{2,2}AV_1^{(2)} = r_{1,1}V_1^{(1)} - r_{1,2}AV_1^{(1)}.$$

Pre-multiplying the above relation by $\mathbb{V}_{m+1}^L$, we have

$$(\mathbb{V}_{m+1}^L \diamond AV_1^{(2)}) = \frac{1}{r_{2,2}}(r_{1,1}(\mathbb{V}_{m+1}^L \diamond V_1^{(1)}) - r_{1,2}(\mathbb{V}_{m+1}^L \diamond AV_1^{(1)})).$$

Consequently,

$$t_{:,2} = \frac{1}{r_{2,2}}(r_{1,1}e_1^{2(m+1)} - r_{1,2}h_{:,1}),$$

In addition, from (15), one gets

$$V_k^{(2)} = AV_{k+1}(H_{k+1,k}e_2^{(2)} \otimes I^{(s)}) + A\mathbb{V}_k(\mathbb{H}_k e_{2k}^{(2k)} \otimes I^{(s)}).$$

This relation implies that

$$\begin{aligned}
&\mathbb{V}_{m+1}^L \diamond AV_{k+1}(H_{k+1,k}e_2^{(2)} \otimes I^{(s)}) \\
&\quad = \mathbb{V}_{m+1}^L \diamond V_k^{(2)} - \mathbb{V}_{m+1}^L \diamond (A\mathbb{V}_k(\mathbb{H}_k e_{2k}^{(2k)} \otimes I^{(s)}) \\
&\quad = e_{2k}^{2(m+1)} - (\mathbb{V}_{m+1}^L \diamond AV_k)\mathbb{H}e_{2k}^{(2k)} \\
&\quad = e_{2k}^{2(m+1)} - \begin{bmatrix} \overline{\mathbb{T}}_k h_{1:2k,2k} \\ 0_{(2m-2k)\times 2k} \end{bmatrix}.
\end{aligned}$$

On the other hand, for the left-hand side of this relation, we deduce

$$\begin{aligned}
&\mathbb{V}_{m+1}^L \diamond AV_{k+1}(H_{k+1,k}e_2^{(2)} \otimes I^{(s)}) \\
&\quad = \mathbb{V}_{m+1}^L \diamond [AV_{k+1}^{(1)} \quad AV_{k+1}^{(2)}] \begin{bmatrix} h_{2k+1,2k}I^{(s)} \\ h_{2k+2,2k}I^{(s)} \end{bmatrix} \\
&\quad = h_{2k+1,2k}\mathbb{V}_{m+1}^L \diamond AV_{k+1}^{(1)} + h_{2k+2,2k}\mathbb{V}_{m+1}^L \diamond AV_{k+1}^{(2)} \\
&\quad = h_{2k+1,2k}t_{:,2k+1} + h_{2k+2,2k}t_{:,2k+2}.
\end{aligned}$$

Hence

$$t_{:,2k+2} = \frac{1}{h_{2k+2,2k}}(-h_{2k+1,2k}t_{:,2k+1} + e_{2k}^{2(m+1)} - \begin{bmatrix} \overline{\mathbb{T}}_k h_{1:2k,2k} \\ 0_{(2m-2k)\times 2k} \end{bmatrix}).$$

By using the inverse of the 2×2 upper triangular matrix $H_{k+1,k}$ and defining $\rho^{(k+1)} = (\ell_{11}^{(k+1)})^{-1}\ell_{12}^{(k+1)}$, this relation can be written as follows:

$$t_{:,2k+2} = t_{:,2k+1}\rho^{(k+1)} + (e_{2k}^{2(m+1)} - \begin{bmatrix} \overline{\mathbb{T}}_k h_{1:2k,2k} \\ 0_{(2m-2k)\times 2k)} \end{bmatrix})\ell_{22}^{(k+1)},$$

which completes the proof. □

4.1 Extended global Hessenberg process for low-rank Sylvester tensor equation

In this subsection, we consider the extended global Hessenberg process derived in the previous subsection for the pair $(A^{(i)}, B^{(i)})$, $i = 1, \ldots, N$. By applying Algorithm 3 with $s = R$ to the pair $(A^{(i)}, B^{(i)})$, $i = 1, \ldots, N$, the block matrices $\mathbb{V}_{m_i} = [V_1^{(i)}, \ldots, V_{m_i}^{(i)}], i = 1, \ldots, N$, are obtained and the following relation holds, for $i = 1, \ldots, N$,

$$\begin{aligned} A^{(i)}\mathbb{V}_{m_i} &= \mathbb{V}_{m_i+1}(\overline{\mathbb{T}}_{m_i} \otimes I^{(R)}) \\ &= \mathbb{V}_{m_i}(\mathbb{T}_{m_i} \otimes I^{(R)}) + V_{m_i+1}^{(i)}(T_{m_i+1,m_i}^{(i)} E_{m_i}^T \otimes I^{(R)}), \end{aligned} \tag{18}$$

where $E_{m_i}^T = \left[0_{2\times 2}, \ldots, 0_{2\times 2}, I^{(2)}\right] \in \mathbb{R}^{2\times 2m_i}$, and $\overline{\mathbb{T}}_{m_i} = (T_{i,j}^{(i)}) \in \mathbb{R}^{2(m_i+1)\times 2m_i}$ is the restriction of $A^{(i)}$ to the extended global Krylov subspace $\mathcal{K}_{m_i}^e(A^{(i)}, B^{(i)})$. Using Line 1 of Algorithm 3, we have

$$B^{(i)} = r_{11}^{(i)}(V_1^{(i)})^{(1)}, \qquad \text{for } \; i = 1, 2, \ldots, N.$$

As in the case of the global Hessenberg process, for the low-rank Sylvester tensor equation (1), we seek an approximate solution of the form

$$\mathcal{X}_m = (\mathcal{Y}_m \otimes \mathcal{I}_R) \times \{\mathbb{V}_m\}, \tag{19}$$

where $\{\mathbb{V}_m\}$ denotes a set of matrices $\mathbb{V}_{m_i} \in \mathbb{R}^{n\times 2Rm_i}$, $i = 1, \ldots, N$, and $\mathcal{Y}_m \in \mathbb{R}^{2m_1\times\cdots\times 2m_N}$ satisfies the low-dimensional Sylvester tensor equation

$$\sum_{i=1}^{N} \mathcal{Y}_m \times_i \mathbb{T}_{m_i} = \beta_m \mathcal{E}_m, \tag{20}$$

where $\beta_m = \prod_{i=1}^{N} r_{11}^{(i)}$ and $\mathcal{E}_m = (e_1^{(2m_1)} \circ \cdots \circ e_1^{(2m_N)})$. In this case, the residual corresponding to $\mathcal{X}_m$ can be written as

$$\mathcal{R}_m = -\sum_{i=1}^{N} (\mathcal{Y}_m \times_i T^{(i)}_{m_{i+1},m_i} E^T_{m_i}) \otimes \mathcal{I}_R \times_1 \mathbb{V}_{m_1} \ldots \times_i V^{(i)}_{m_i+1} \ldots \times_N \mathbb{V}_{m_N}. \tag{21}$$

We can easily obtain

$$\|\mathcal{R}_m\| \leq \sqrt{((2nR-2m+1)m)^N} \sqrt{\sum_{i=1}^{N} \|\mathcal{Y}_m \times_i T^{(i)}_{m_{i+1},m_i} E^T_{m_i}\|} \tag{22}$$

and

$$\|\mathcal{R}_m\| \leq \sqrt{(2nmR)^N} \sqrt{\sum_{i=1}^{N} \|\mathcal{Y}_m \times_i T^{(i)}_{m_{i+1},m_i} E^T_{m_i}\|}, \tag{23}$$

where $m = \max\limits_{1\leq i\leq N} m_i$. Finally, the following estimate is derived heuristically:

$$\|\mathcal{R}_m\| \approx E_m := \sqrt[N]{(2nmR)} \sqrt{\sum_{i=1}^{N} \|\mathcal{Y}_m \times_i T^{(i)}_{m_i+1,m_i} E^T_{m_i}\|}. \tag{24}$$

For the extended global Hessenberg process, the main part of Algorithm 2 remains the same except that the lines 6, 9, and 10 must be changed as follows:

6. Construct the basis $\left[V_{k_i+1}, \ldots, V_{k_i+k'_i}\right]$ and the matrix $\mathbb{T}_{m_i}$ by Algorithm 3 and the formulas of Proposition 6.
9. Solve the low-dimensional equation $\sum_{i=1}^{N} \mathcal{Y}_m \times_i \mathbb{T}_{m_i} = \beta_m \mathcal{E}_m$ by the recursive blocked algorithms presented in [11].
10. Compute the estimated residual norm of $\mathcal{R}_m$, that is, $E_m = \sqrt[N]{(2nmR)} \sqrt{\sum_{i=1}^{N} \|\mathcal{Y}_m \times_i T^{(i)}_{m_i+1,m_i} E^T_{m_i}\|^2}$.

5 Complexity consideration

In this section, we present the required number of operations to solve the low-rank Sylvester tensor equation (1) for $I_1 = I_2 = \cdots = I_N$. Let Nnz denote the number of nonzero elements of matrix A, and suppose that the LU decomposition of A is available for computing the block matrix $W = A^{-1}V$. We compare the required operations for the extended global Hessenberg process and the extended global Arnoldi process [18]. Algorithm 3 requires $(2n^2s + 4ns)$ operations for computing the block matrices $V_1^{(1)}$ and $V_1^{(2)}$. In addition, the iteration k of this algorithm involves

- $V_{k+1}^{(1)}$, which requires $2sNnz + ns(4k+1) - 4k^2$ operations,
- $V_{k+1}^{(2)}$, which requires $2n^2s + ns(4k+3) - (2k+1)^2$ operations.

Note that the global Arnoldi process (Algorithm 2 in [18]) requires $2n^2s+10ns$ operations for computing the global QR decomposition $[V, A^{-1}V]$, and the iteration k of this process involves

- $U = [AV_k^{(1)}, A^{-1}V_k^{(2)}]$, which requires $2sNnz + 2n^2s$ operations.
- $H_{i,j} = V_i^T \diamond U,\ U = U - V_i(H_{i,j} \otimes I^{(s)}),\ i = 1, 2, \ldots, k$, which require $16nsk$ operations.
- the global decomposition of U, that is, $U = V_{k+1}(H_{k+1,k} \otimes I^{(s)})$, which requires $10ns$ operations.

Therefore, for computing an approximation of the solution of Sylvester tensor equation (1), the total cost number of operations required to perform m iterations of the extended global versions of Arnoldi and Hessenberg processes is approximately shown in Table 1. In addition, the total cost number of operations required to perform m iterations of the global Hessenberg process (Algorithm 1) and the modified global Arnoldi process (Algorithm 2.2 in [23]) is presented in this table. According to Table 1, when solving the low-rank Sylvester tensor equation (1), the global and extended global Hessenberg processes are less expensive than the global and extended global Arnoldi ones. On the other hand, these Hessenberg processes use the maximum strategy. Hence they involve some data movement. However, these processes need slightly less storage than the Arnoldi processes per iteration.

Table 1: Operation count for the global and extended global versions of Hessenberg and Arnoldi processes.

Process	Number of operations
Global Arnoldi	$N(2mRNnz + (m+1)(2m+3)nR - (m(m+1))/2)$
Global Hessenberg	$N(2mRNnz + (m+1)^2nR - (m(m+1)(2m+1))/6)$
Extended Global Arnoldi	$N(2mRNnz + 2(m+1)n^2R + (m+1)(8m+10)nR)$
Extended Global Hessenberg	$N(2mRNnz + 2(m+1)n^2R + 4(m+1)^2nR - m(8m^2+18m+13)/3)$

6 Numerical experiments

In this section, some test problems with $N = 3$ are used to examine the robustness of two new presented methods for solving the low-rank Sylvester equation (1). All the numerical experiments were performed in double-precision floating-point arithmetic in MATLAB 2021a. The machine we have used is an Intel(R) Xeon(R) CPU E5-2680 v4@2.40 GHz, 128 GB of RAM, using the Tensor Toolbox [2]. We employ the recursive blocked algorithms

introduced in [11] to solve the low-dimensional Sylvester tensor equations (8) and (20). The step size parameter k' associated with one cycle is equal to 3. The algorithms stopped whenever $E_m \leq 10^{-7}$, where E_m is the estimate of $\|\mathcal{R}_m\|$. We also compare the numerical behavior of the methods in terms of the number of cycles (Cycle), the norm of residual $\|\mathcal{R}_m\|$, the norm of error $\|\mathcal{X}^* - \mathcal{X}_m\|$, where $\mathcal{X}^*$ is the exact solution, and the CPU time in seconds (CPU time) required only for constructing the Krylov subspace basis and the solution of reduced Sylvester tensor equation. Note that we use the procedure $cp\_als(\mathcal{B}, R)$ from the toolbox [2] to compute the CP decomposition of the right-hand side $\mathcal{B}$. In Table 2, we report $\|\mathcal{B} - \mathcal{B}_{cp}\|$, where the $\mathcal{B}_{cp}$ is the CP decomposition corresponding to the right-hand side tensor $\mathcal{B}$, using the procedure $cp\_als(\mathcal{B}, R)$. The results of examples are reported in Table 2. For each example, the rank R and the dimension n are presented in this table. In Figure 1 , by plotting the norm of residual $\|R_m\|_F$ versus the number of cycles, we display the convergence history of the global and extended global Arnoldi and Hessenberg algorithms for Examples 1–5.

Example 1. In this example, as in [5], we consider the matrices $A^{(i)}, i = 1, 2, 3$, corresponding to discretization of the operator

$$L(u) = \Delta u - f_1(x, y)\frac{\partial u}{\partial x} + f_2(x, y)\frac{\partial u}{\partial y} + g(x, y),$$

in the unit square $[0, 1] \times [0, 1]$ with Dirichlet homogeneous boundary conditions. The number of inner grid points in each direction is n_0 for the operator L. The discretization of the operator L yields matrices extracted from the Lyapack package [29], using the command fdm and denoted as

$$A^{(i)} = \text{fdm}(n_0, f_1(x, y), f_2(x, y), g(x, y)), \qquad i = 1, 2, 3,$$

with $f_1(x, y) = e^{xy}, f_2(x, y) = sin(x, y), g(x, y) = y^2 - x^2$, $n = n_0^2$. The right-hand side tensor is chosen in such a way that the exact solution of the Sylvester tensor equation (1) has the form $\mathcal{X}^* = x_1 \circ x_2 \circ x_3$, with $x_i = \text{rand}(n, 1)$, for $i = 1, 2, 3$.

Example 2. Assume that in the Sylvester tensor equation (1), the coefficient matrices are presented as [5]

$$A^{(i)} = \text{gallery}('\text{poisson}', n_0), \quad i = 1, 2, 3,$$

where $n = n_0^2$. The right-hand side tensor is constructed such that the exact solution $\mathcal{X}$ of the Sylvester tensor equation (1) is a tensor with entries equal to one.

Example 3. Let $A^{(i)}$, $i = 1, 2, 3$, be defined as [10]

$$A^{(i)} = \text{rand(n,n)} + \text{diag(ones(n,1)} * \text{alfa)},$$

where alfa = 8 and the right-hand side tensor is constructed as in Example 1.

Example 4. Consider the Sylvester equation (1) with the coefficient matrices generated by [38]

$$A^{(i)} = \text{diag(rand(n-1,1),-1)} + \text{diag}(2 + \text{diag(rand(n,n)})), \quad i = 1, 2, 3,$$

and the right-hand side tensor is constructed as in Example 1.

Example 5. The coefficient matrices $A^{(i)},\ i = 1, 2, 3$, for the Sylvester tensor equation (1) are defined as

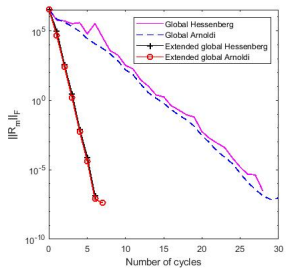
$$A^{(i)}(l, j) = \frac{1}{1 + |l - j|},$$

and the right-hand side tensor is constructed as in Example 1.

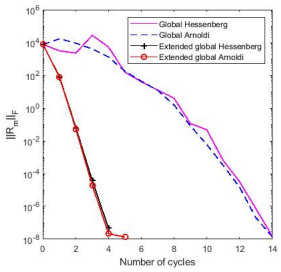
Table 2: Results of Examples 1–5.

Example	Algorithm	$\|\mathcal{B} - \mathcal{B}_{cp}\|$	$\|\mathcal{R}_m\|$	$\|\mathcal{X}^* - \mathcal{X}_m\|$	Cycle	CPU time
Example 1 $n = 400$,$R = 4$	Global Arnoldi	$3.655e-08$	$8.549e-08$	$9.903e-11$	30	2.879
	Global Hessenberg	$3.655e-08$	$2.901e-07$	$2.667e-10$	28	2.558
	Extended Global Arnoldi	$3.655e-08$	$4.197e-08$	$3.173e-11$	7	0.261
	Extended Global Hessenberg	$3.655e-08$	$1.411e-07$	$1.162e-10$	6	0.110
Example 2 $n = 400$, $R = 3$	Global Arnoldi	$1.355e-08$	$1.406e-08$	$1.560e-08$	14	0.138
	Global Hessenberg	$1.355e-08$	$1.573e-08$	$1.735e-08$	14	0.229
	Extended Global Arnoldi	$1.355e-08$	$1.375e-08$	$1.603e-08$	5	0.079
	Extended Global Hessenberg	$1.355e-08$	$4.528e-08$	$2.652e-08$	4	0.058
Example 3 $n = 500$, $R = 3$	Global Arnoldi	$1.532e-05$	$1.531e-05$	$3.731e-07$	19	0.612
	Global Hessenberg	$1.532e-05$	$1.530e-05$	$3.729e-07$	18	0.479
	Extended Global Arnoldi	$1.532e-05$	$1.531e-05$	$3.731e-07$	9	0.429
	Extended Global Hessenberg	$1.532e-05$	$1.531e-05$	$3.731e-07$	8	0.267
Example 4 $n = 500$,$R = 3$	Global Arnoldi	$1.980e-07$	$1.980e-07$	$2.698e-08$	5	0.049
	Global Hessenberg	$1.980e-07$	$1.984e-07$	$2.704e-08$	4	0.046
	Extended Global Arnoldi	$1.980e-07$	$1.980e-07$	$2.698e-08$	3	0.082
	Extended Global Hessenberg	$1.980e-07$	$1.980e-07$	$2.698e-08$	3	0.077
Example 5 $n = 500$, $R = 3$	Global Arnoldi	$1.038e-08$	$1.034e-08$	$2.567e-09$	12	0.120
	Global Hessenberg	$1.038e-08$	$1.161e-08$	$2.622e-09$	11	0.144
	Extended Global Arnoldi	$1.038e-08$	$1.042e-08$	$2.567e-09$	5	0.115
	Extended Global Hessenberg	$1.038e-08$	$1.034e-08$	$2.566e-09$	5	0.112

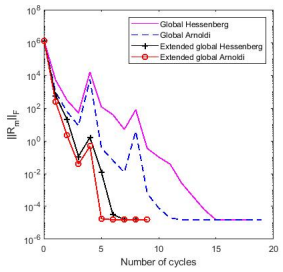
As can be seen from Table 2 and Figure 1, Global Arnoldi, Extended Global Arnoldi, and Global Hessenberg, Extended Global Hessenberg methods are shown a similar behavior. In addition, for all examples, the number of cycles of Extended Global Hessenberg is less than or equal to that of the other methods. In Examples 1, 2, 3, and 5, the CPU time of Extended Global Hessenberg method is less than the others. The results of Example 4 show that when the required number of cycles is small for Global Hessenberg method, this method outperforms the other methods in terms of CPU times.



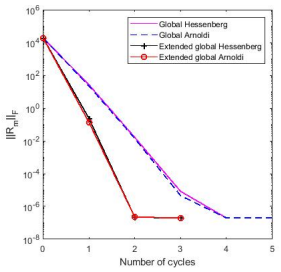
(a) Example 1



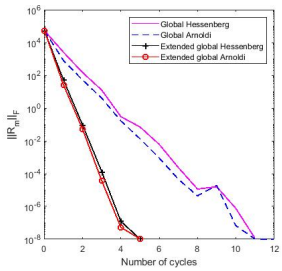
(b) Example 2



(c) Example 3



(d) Example 4



(e) Example 5

Figure 1: Convergence history of the global and extended global Arnoldi and Hessenberg algorithms for Examples 1–5.

7 Conclusion

In this study, for computing the approximate solutions of the Sylvester tensor equation (1) with the low-rank right-hand side, two new projection methods based on the Hessenberg process were proposed. The theoretical results of these methods were presented and analyzed as well. The global and extended global Hessenberg algorithms were compared, in terms of CPU times, cycles, and the number of operations, with the global and extended global Arnoldi

algorithms, respectively. Numerical examples showed that the global and extended global Hessenberg algorithms are efficient and feasible for solving the low-rank Sylvester tensor equation (1).

Acknowledgment

The authors would like to thank the anonymous referees for their comments and suggestions, which considerably improved the paper. All computations of this paper were carried out in the High-Performance Computing Center of Shahrekord university. The authors of this paper would like to express their gratitude for this support.

References

1. Addam, M., Elbouyahyaoui, L. and Heyouni, M. *On Hessenberg type methods for low-rank Lyapunov matrix equations,* Applicationes Mathematicae 45 (2018) 255–273.

2. Bader, B.W. and Kolda, T.G. *MATLAB tensor toolbox version 3.2.* http://www.sandia.gov/tgkolda/TensorToolbox/.

3. Ballani, J. and Grasedyck, L. *A projection method to solve linear systems in tensor format,* Numer. Linear Algebra Appl. 20 (1) (2013) 27–43.

4. Beik, F.P.A., Saberi Movahed, F. and Ahmadi-Asl, S. *On the krylov subspace methods based on tensor format for positive definite Sylvester tensor equations,* Numer. Linear Algebra Appl. 23 (3) (2016) 444–466.

5. Bentbib, A.H., El-Halouy, S. and Sadek, El M. *Krylov subspace projection method for Sylvester tensor equation with low rank right-hand side,* Numer. Algorithms 84 (4) (2020) 1411–1430.

6. Bentbib, A.H., El-Halouy, S. and Sadek, El M. *Extended Krylov subspace methods for solving Sylvester and Stein tensor equations,* Discrete Continuous Dyn. Syst.-s 15 (1) (2022) 41–56.

7. Bouyouli, R., Jbilou, K., Sadaka, R. and Sadok, H. *Convergence properties of some block Krylov subspace methods for multiple linear systems,* J. Comp. Appl. Math. 196 (2) (2006) 498–511.

8. Chang, K.C., Pearson, K. and Zhang, T. *Perron-Frobenius theorem for nonnegative tensors,* Commun. Math. Sci. 6 (2) (2008) 507–520.

9. Chen, Z. and Lu, L.Z. *A projection method and Kronecker product preconditioner for solving Sylvester tensor equations,* Sci. China Math. 55 (6) (2012) 1281–1292.

10. Chen, Z. and Lu, L.Z. *A gradient based iterative solutions for Sylvester tensor equations,* Math. Probl. Eng. (2013) Article ID 819479, 7 pp.

11. Chen, M. and Kressner, D. *Recursive blocked algorithms for linear systems with Kronecker product structure,* Numer. Algorithms 84 (3) (2020) 1199–1216.

12. Cheraghzadeh, T., Khoshsiar Ghaziani, R. and Toutounian, F. *Projection schemes based on Hessenberg process for Sylvester tensor equation with low-rank right-hand side,* Comput. Appl. Math. 41 (2022) Article number 311.

13. Colda, T.G. and Bader, B.W. *Tensor decompositions and applications,* SIAM Rev. 51 (3) (2009) 455–500.

14. Cichocki, A., Zdunek, R., Phan, A.H. and Amari, S.-I. *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation,* John Wiley and Sons, 2009.

15. Dehdezi, E.K. and Karimi, S. *Extended conjugate gradient squared and conjugate residual squared methods for solving the generalized coupled Sylvester tensor equations,* T. I. Meas. Control. 9(4)(2021) 645–664.

16. Dehdezi, E.K. and Karimi, S. *A gradient based iterative method and associated preconditioning technique for solving the large multilinear systems,* Calcolo. 58(4) (2021) 1–19.

17. Heyouni, M. *The global Hessenberg and global CMRH methods for linear systems with multiple right-hand sides,* Numer. Algorithms 26 (4) (2001) 317–332.

18. Heyouni, M. *Extended Arnoldi methods for large low-rank Sylvester matrix equations,* Appl. Numer. Math. 60 (11) (2010) 1171–1182.

19. Heyouni, M. and Essai, A. *Matrix Krylov subspace methods for linear systems with multiple right-hand sides,* Numer. Algorithms 40 (2) (2005) 137–156.

20. Heyouni M. and Jbilou, K. *An extended block Arnoldi algorithm for large-scale solution of the continuous-time algebraic Riccati equation,* Electron. Trans. Numer. Anal. 33 (2009) 53–62.

21. Heyouni, M., Saberi-Movahed, F. and Tajaddini, A. *A tensor format for the generalized Hessenberg method for solving Sylvester tensor equations,* J. Comput. Appl. Math. 377 (2020) 112878.

22. Huang, B., Xie, Y. and Ma, C. *Krylov subspace methods to solve a class of tensor equations via the Einstein product,* Numer. Linear Algebra Appl. 26 (2019) e2254.

23. Jbilou, K., Messaoudi, A. and Sadok, H. *Global FOM and GMRES algorithms for matrix equations,* Appl. Numer. Math. 31 (1) (1999) 49–63.

24. Karimi, S. and Dehghan, M. *Global least squares method based on tensor form to solve linear systems in Kronecker format,* Trans. Inst. Measur. Contr. 40 (7) (2018) 2378–2386.

25. Kressner, D. and Tobler, C. *Low-rank tensor Krylov subspace methods for parametrized linear systems,* SIAM J. Matrix Anal. Appl. 32 (4) (2011) 1288–1316.

26. Lee, N. and Cichocki, *A. Fundamental tensor operations for large-scale data analysis using tensor network formats,* Multidim. Syst. Sign. Process. 29 (3) (2018) 921–960.

27. Li, B.W., Tian, S., Sun, Y.S. and Hu, Z.M. *Schur decomposition for 3d matrix equations and its application in solving radiative discrete ordinates equations discretized by Chebyshev collocation spectral method,* J. Comput. Phys. 229 (4) (2010) 1198–1212 .

28. Malek, A. and Momeni-Masuleh, S.H. *A mixed collocation finite difference method for 3d microscopic heat transport problems,* J. Comput. Appl. Math. 217 (1) (2008) 137–147.

29. Penzl, T. et al., *A Matlab toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear quadratic optimal control problems,* https://www.tu-chemnitz.de/ sfb393/lyapack/(2000).

30. Mele, G. and Jarlebring, E. *On restarting for the tensor infinite Arnoldi method,* BIT 58 (1) (2018) 133–162.

31. Qi, L. *Eigenvalues of a real supersymmetric tensor,* J. Symbolic Comput. 40 (6) (2005) 1302–1324.

32. Ramezani, Z. and Toutounian, F. *Extended and rational Hessenberg methods for the evaluation of matrix functions,* BIT Numer. Math. 59, (2019) 523–545.

33. Saberi-Movahed, F., Tajaddini, A., Heyouni, M. and Elbouyahyaoui, L. *Some iterative approaches for Sylvester tensor equations, PartI: A tensor format of truncated Loose Simpler GMRES,* Appl. Numer. Math. 172 (2022) 428–445.

34. Saberi-Movahed, F., Tajaddini, A., Heyouni, M. and Elbouyahyaoui, L. *Some iterative approaches for Sylvester tensor equations, Part II: A tensor format of truncated Loose Simpler GMRES,* Appl. Numer. Math. 172 (2022) 428–445.

35. Sadok, H. *CMRH: A new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm,* Numer. Algorithms 20 (4) (1999) 303–321.

36. Simoncini, V. *A new iterative method for solving large-scale Lyapunov matrix equations,* SIAM J. Sci. Comp. 29 (3) (2007) 1268–1288.

37. Sun, Y., Ma, J. and Li, B.W. *Chebyshev collocation spectral method for three-dimensional transient coupled radiative conductive heat transfer,* J. Heat Transfer 134 (9) (2012):092701.

38. Xu, X. and Wang, Q.-W. *Extending Bi-CG and Bi-CR methods to solve the Stein tensor equation,* Comput. Math. Appl. 77 (12) (2019) 3117–3127.

39. Zhang, X.-F.and Wang, Q.-W. *Developing iterative algorithms to solve Sylvester tensor equations,* Appl. Math. Comput. 409 (2021) 126403.

How to cite this article
T. Cheraghzadeh, F. Toutounian and R. Khoshsiar Ghaziani Global and extended global Hessenberg processes for solving Sylvester tensor equation with low-rank right-hand side. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 658-679. doi: 10.22067/ijnao.2022.78966.1186.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 680–703
DOI:10.22067/ijnao.2022.78108.1166
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Shooting continuous Runge–Kutta method for delay optimal control problems

T. Jabbari-Khanbehbin, M. Gachpazan* , S. Effati and S.M. Miri

Abstract

In this paper, we present an efficient method to solve linear time-delay optimal control problems with a quadratic cost function. In this regard, first, by employing the Pontryagin maximum principle to time-delay systems, the original problem is converted into a sequence of two-point boundary value problems (TPBVPs) that have both advance and delay terms. Then, using the continuous Runge–Kutta (CRK) method, the resulting sequences are recursively solved by the shooting method to obtain an optimal control law. This obtained optimal control consists of a linear feedback term, which is obtained by solving a Riccati matrix differential equation, and a forward term, which is an infinite sum of adjoint vectors, that can be obtained by solving sequences of delay TPBVPs by the shooting CRK method. Finally, numerical results and their comparison with other available results illustrate the high accuracy and efficiency of our proposed method.

* Corresponding author

Received 6 August 2022; revised 12 September 2022; accepted 29 September 2022

Tahereh Jabbari-Khanbehbin
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran. e-mail: jabbari.tahere@gmail.com

Mortaza Gachpazan
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran. e-mail: gachpazan@um.ac.ir

Sohrab Effati
Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP), Ferdowsi University of Mashhad, Iran. e-mail: s-effati@um.ac.ir

Seyed Mohsen Miri
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran. e-mail: mohsenmiri80@gmail.com

AMS subject classifications (2020): 49M05; 34K35; 34K10; 34K28.

Keywords: Pontryagin maximum principle; Time-delay two-point boundary value problems; Time-delay optimal control problems; Continuous Runge–Kutta methods; Shooting method.

1 Introduction

In recent years, optimization and control of systems with time delay have been considered in much research because the time delay in many processes cannot be ignored. To more accurately express the behavior of a natural phenomenon, we need a more complex system. Some of the applications of these issues are in the chemical, electronic, medicine, engineering, biological, economy, and so on [22, 19, 12, 7, 8, 41].

In general, two methods are provided to solve optimal control problems (OCPs). The first approach involves the use of necessary and (or) sufficient conditions of optimality by applying the Pontryagin minimum (maximum) principle or optimality principle. The minimum principle was presented in 1956 by the Russian mathematician Lev Pontryagin and his students, and its primary application was to maximize the terminal velocity of a rocket. This result was obtained using the classical ideas of variational calculus. The equations obtained from these conditions can be solved numerically. This approach yields indirect methods, which are known as analytical-based methods; see [39, 43, 17, 13].

In another approach, an OCP is considered an optimization problem. instead of using the optimality conditions, the dynamic constraints are transformed into an algebraic equations system by discretizing the time interval and parameterizing the variables of the problem. Therefore, the OCP becomes a nonlinear programming problem of dimension finite. The resulting nonlinear programming problem can then be solved using optimization techniques. This approach yields direct methods. We refer the reader to [11, 2, 18, 26, 8]. Since direct methods do not need to calculate the optimality conditions, they can be used for a wide range of OCPs. However, the lack of guarantee for the optimal solution and the high amount of memory resources and time for producing a close approximation is among the disadvantages of these methods.

In the case of time-delay OCPs, in 1963, Oğuztöreli [35] was one of the pioneers in the analytical-based approach (also, see [36]). For the first time, Kharatishvili [24] generalized the Pontryagin maximum principle for OCPs with a constant delay in the state variable. Then in [25], he gave similar results on OCPs with delay in the control variable. After that, in 1968, a maximum principle for OCPs with multiple constant delays in state and control was proved by Halanay [16]. In 1972, Ray and Soliman [42] also obtained similar results. Guinn [14] transformed the delayed OCP with constant delay

in the state variable into a higher-dimensional undelayed OCP. Banks [3] derives a maximum principle for control systems with a time-dependent delay in the state variable.

The system resulted from the necessary conditions that Kharatishvili provided, which was a two-point boundary value problem involving both advance and delay terms. This type of problem does not have an exact solution, except in exceptional cases. Therefore, there are many attempts available in the literature to approximately solve this problem; for example, see [29, 44, 30, 31, 32, 20, 21, 6].

The following articles can be mentioned as the latest studies. For OCPs with time-invariant delayed systems, Mirhosseini-Alizamini, the second author, and Heydari [32] applied the variational iteration method and then obtained a suboptimal solution for the two-point boundary value problem (TPBVP). Moreover, Mirhosseini-Alizamini and the second author [31] investigated infinite horizon OCPs with time-variant delayed systems. Also, using a Hermite interpolation polynomial for delay terms and employing a second-order finite difference formula for the first-order derivatives, Jajarmi and Hajipour [21] converted the TPBVP obtained from the time-delay OCP into a system of linear algebraic equations and then solved it. Recently, using an algorithm based on the forward and backward difference approximation, Bouajaji et al. [6] solved the system obtained from the application of the Pontryagin maximum principle to a delayed OCP.

In this work, we investigate a family of time-delay OCPs with a quadratic cost functional that should be minimized subject to a linear time-delay system with constant delay in the state variable. Using the Pontryagin minimum principle for delayed systems from [24] and then applying continuous Runge–Kutta (CRK) methods, we convert a time-delay OCP into a sequence of linear TPBVPs and thereafter solve it recursively by the shooting method to obtain the optimal control law.

The rest of the paper is organized as follows: The CRK methods are presented in Section 2. After that, in Section 3, we introduce the Shooting CRK (SCRK) method and apply it to a delayed TPBVP. Then, in the continuation of this section, we present a basic algorithm for the proposed method. In the next section, we will use a generalization of this algorithm to solve a time-delay OCP. Section 4 describes the Pontryagin maximum principle for our delayed OCP and designs an algorithm based on the previous algorithm defined in Section 3 for solving the final system. In Section 5, we give several numerical examples to demonstrate the effectiveness and accuracy of the proposed technique. Finally, with the conclusion in Section 6, we end the article.

2 CRK methods

In this section, we describe the CRK methods. Consider $f(t,x(t)) \in C^0([t_0,t_f]\times\mathbb{R}^d,\mathbb{R}^d)$. The CRK methods were originally designed to treat the initial value problem for the following ordinary differential equation:

$$\begin{cases}\dot{x}(t)=f(t,x(t)), & t_0\le t\le t_f,\\ x(t_0)=x_0.\end{cases} \tag{1}$$

Some of the implicit Runge–Kutta methods are equivalent to collocation methods; see [46]. Thus, they sequentially provide a continuous extension of the approximate solution without any additional evaluation of f. Indeed, the next question is whether there is such a continuous extension for each Runge–Kutta process that is given sequentially by the method itself?

Nørsett and Wanner partially answered this question by proving that a large number of Runge–Kutta methods are the same as the somewhat pertubed collocation method that is somewhat perturbed. After that, Zennaro [47] presented a continuous extension of the solution provided by a Runge–Kutta method, which includes the collocation solution if it is equivalent to collocation and behaves similarly in other cases.

Let $\Delta=\{t_0,\dots,t_n,\dots,t_N=t_f\}$ be an arbitrary mesh. Then for the numerical solution of the ordinary differential equation (1), an s-stage discrete Runge–Kutta method has the form

$$x_{n+1}=x_n+h_{n+1}\sum_{i=1}^{s}b_ik_i, \tag{2}$$

$$k_i=f(t_n^i,x_n+h_{n+1}\sum_{j=1}^{s}a_{ij}k_j),\quad i=1,\dots,s, \tag{3}$$

where $c_i=\sum_{j=1}^{s}a_{ij}, t_n^i=t_n+c_ih_{n+1}, i=1,\dots,s$, and $h_{n+1}=t_{n+1}-t_n$. In addition, the Runge–Kutta method (2) and (3) is denoted by (A,b). Let the solution have advanced to the point $t=t_n$. Zennaro [47] showed that for this s-stage Runge–Kutta method of order p, there is a CRK method of degree d, if there exist s polynomials $b_i(\theta),\ i=1,\dots,s$, of degree less than or equal to d, independent of f. This method reads as follows:

$$\eta(t_n+\theta h_{n+1})=x_n+h_{n+1}\sum_{i=1}^{s}b_i(\theta)k_i,\quad 0\le\theta\le 1, \tag{4}$$

$$k_i=f(t_n^i,x_n+\sum_{j=1}^{s}a_{ij}k_j),\quad i=1,\dots,s, \tag{5}$$

where

$$\eta(t_n)=x_n,\qquad \eta(t_n+h_{n+1})=x_{n+1},$$

and x_n is an approximate solution obtained by applying the Runge–Kutta method for $x(t_n)$. This method, which is usually expressed as $(A, b(\theta))$, can also be related to the following CRK tableau:

$$\begin{array}{c|c} \mathrm{C} & \mathrm{A} \\ \hline & b^T(\theta) \end{array}.$$

In fact, $\{c_i, a_{ij}\}$'s are the same as the coefficients of the discrete Runge–Kutta method. Now, we recall the consistency of the discrete Runge–Kutta method from [5].

Definition 1. [5, Definition 5.1.3] Consider $p \geq 1$ the largest integer having the following property: For every mesh point and C^p-continuous right-hand-side function $f(t, x)$ in (1), the local solution $z_{n+1}(t)$ to the local problem

$$\begin{cases} z'_{n+1}(t) = f(t, z_{n+1}(t)), \quad t_n \leq t \leq t_{n+1}, \\ z_{n+1}(t) = x^*_n, \end{cases} \tag{6}$$

satisfies

$$\|z_{n+1}(t_{n+1}) - x_{n+1}\| = O(h^{p+1}_{n+1})$$

uniformly with respect to x^*_n belonging to a bounded subset of $\mathbb{R}^d$ and respect to $n = 0, \ldots, N-1$, Then we say that the discrete Runge–Kutta method (A, b) is consistent with order p.

Similarly, with the above notations, we say that the continuous extension (4) is consistent with uniform order q if $q \geq 1$ is the largest integer having the following property:

$$\max_{t_n \leq t \leq t_{n+1}} \|z_{n+1}(t) - \eta(t)\| = O(h^{q+1}_{n+1}),$$

for every mesh point and C^q-continuous right-hand-side function $f(t, x)$ in (1).

According to Definition 1, the convergence results in discrete and CRK methods for ordinary differential equations have been proved in the following theorem; see [5].

Theorem 1. [5, Theorem 5.1.4] Suppose that the Runge–Kutta method (2) and (3) is consistent with order p and that $f(t, x)$ defined in (1) is a right-hand-side C^p-continuous function. Then, on any bounded interval $[t_0, t_f]$, the method has discrete global order (or, equivalently, is convergent of order) p. In other words,

$$\max_{1 \leq n \leq N} \|x(t_n) - x_n\| = O(h^p),$$

in which $h = \max\limits_{1\leq n\leq N} h_n$.

Moreover, let the continuous extension (4) have the uniform order q. Then the CRK method (4) and (5) has the uniform global order (or, equivalently, uniformly convergent of order) $q' = \min(q+1, p)$, which means that

$$\max_{t_0\leq t\leq t_f} \|x(t) - \eta(t)\| = O(h^{q'}).$$

Then, Baker and Paul [1] generalized this idea for a CRK method to delay differential equations with a general delay differential equation of the form

$$\begin{cases} \dot{x}(t) = f(t, x(t), x(t-\tau(t)), & t > t_0, \\ x(t) = \phi(t), & t_0 - \tau(t_0) \leq t \leq t_0, \end{cases} \tag{7}$$

in which $f : \mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^n \to \mathbb{R}^n$ and $\tau(t) \geq 0$. Moreover, $\phi \in C^0[t_0 - \tau(t_0), t_0]$ denotes the initial information of the state variable x. For delay differential equations, Baker and Paul [1] modified (4) and (5) as follows :

$$\eta(t_n + \theta h_{n+1}) = x_n + h_{n+1}\sum_{i=1}^{s} b_i(\theta)k_i, \quad 0 \leq \theta \leq 1, \tag{8}$$

$$k_i = f(t_n^i, X_i, \eta(t_n^i - \tau(t_n^i))), \quad i = 1, \ldots, s, \tag{9}$$

$$X_i = x_n + h_{n+1}\sum_{j=1}^{s} a_{ij}k_j, \quad i = 1, \ldots, s. \tag{10}$$

When the delay is constant and $h_{n+1} \leq \tau$, then $\eta(t_n^i - \tau)$ is known for any i $(0 \leq c_i \leq 1)$. In this case, $\eta(t_n^i - \tau)$ is available from the past, so this method is an explicit CRK method. The pair formed by (A, b) and $(A, b(\theta))$ is called the underlying CRK method.

Theorem 2. [5, Theorem 6.3.1] Assuming the delay differential equation (7), suppose that $f(t, x, y) \in [t_0, t_f]\times\mathbb{R}^n\times\mathbb{R}^n$ is a C^p-continuous function. Then the delay $\tau(t) \in [t_0, t_f]\times\mathbb{R}^n$ is a C^p-continuous function and $\phi(t)$ is the initial C^p-continuous function. In addition, let $\Delta = \{t_0, t_1, \ldots, t_n, \ldots, t_N = t_f\}$ be the mesh containing all points of discontinuity with the order less than or equal to p being in $[t_0, t_f]$. Also, assume that the underlying CRK method has the uniform and discrete orders q and p, respectively. Then for the delay differential equation, the CRK method (8), (9), and (10) has uniform global and discrete global orders $q' = \min(p, q+1)$. In other words,

$$\max_{1\leq n\leq N}\|x(t) - \eta(t)\| = O(h^{q'}),$$

and

$$\max_{1\leq n\leq N}\|x(t_n) - x_n\| = O(h^{q'}),$$

where $h = \max_{1 \le n \le N} h_n$.

3 Outline of SCRK method for a delay TPBVP

In the present section, we first state details of the proposed method on a TPBVP with only a time-delay term. Therefore, consider the following basic form of a first-order TPBVP with a time delay:

$$\begin{cases} \dot{x}(t) = f_1(t, x(t), y(t), x(t-\tau), y(t-\tau)), & t_0 \le t \le t_f, \\ \dot{y}(t) = f_2(t, x(t), y(t), x(t-\tau), y(t-\tau)), & t_0 \le t \le t_f, \\ x(t) = \phi(t), & t_0 - \tau \le t \le t_0, \\ y(t_f) = \beta. \end{cases} \tag{11}$$

For solving this problem, we need to use the solutions to a sequence of initial value problems that are made by substituting the initial guess $y(t_0) = z$ instead of the terminal condition $y(t_f) = \beta$ in (11).

To approximate a solution to the boundary value problem (11), we involve a parameter z, by choosing the parameters $z = z_k$ such that

$$\lim_{k \to \infty} y(t_f, z_k) = y(t_f) = \beta,$$

where $y(t)$ is the solution to the boundary value problem (11) and $y(t, z_k)$ denotes the solutions to the constructed initial value problem with initial conditions $x(t) = \phi(t),\ t_0 - \tau \le t \le t_0$ and $y(t_0) = z_k$.

This technique is called the Shooting method. For starting, we choose a parameter z_1 such that it determines the initial evaluation at which the object is fired from the point $(t_0, \phi(t_0))$ and along the curve indicated by the solution to the problem

$$\begin{cases} \dot{x}(t) = f_1(t, x(t), y(t), x(t-\tau), y(t-\tau)), & t_0 \le t \le t_f, \\ \dot{y}(t) = f_2(t, x(t), y(t), x(t-\tau), y(t-\tau)), & t_0 \le t \le t_f, \\ x(t) = \phi(t), & t_0 - \tau \le t \le t_0, \\ y(t_0) = z_1. \end{cases} \tag{12}$$

If $y(t_f, z_1)$ is not sufficiently close to β, then we correct the approximation by choosing elevations z_2, z_3, and so on, until $y(t_f, z_k)$ is sufficiently close to β.

For determining the parameters z_k, we must solve this problem:

$$y(t_f, z) - \beta = 0. \tag{13}$$

To solve this nonlinear equation, we use the secant method. For this method, we need to choose initial approximations z_1 and z_2 and then generate the remaining terms of the sequence by the following formula:

Algorithm 1 SCRK method for time-delay TPBVP

Step 1. Set N (the number of subintervals), $h = \frac{t_f - t_0}{N}$, $K = 1$, M (the maximum number of iterations), and s (the number of stages of the CRK method), and choose z_1, z_2, and tolerance error bound ϵ.

Step 2. While $L \leq M$, do

- Set $x_0 = \alpha$ and $y(t_0) = z_1$,

Step 3. For $k = 1, 2, \cdots$,

- solve (12), using the CRK method (15).
- Set $x_0 = \alpha$ and $y(t_0) = z_1$,

Step 4. Check the stop condition,

- If $|y_N - \beta| < \epsilon$, then the procedure is complete, and jump to Step 7,
- else, go to the next step.

Step 5. If $k = 1$, then set $y(t_0) = z_2$ and back to Step 3,

- else, go to the next step,

Step 6. Calculate the next approximation for z_{k+1} from (14), set $y(t_0) = z_{k+1}$, and back to Step 3.

- end for

Step 7. Stop the algorithm and output (t_n, x_n, y_n).

- end while

Step 8. Output (maximum number of iterations exceeded).

- Stop

$$z_{k+1} = z_k - \frac{y(t_f, z_k) - \beta}{y(t_f, z_k) - y(t_f, z_{k-1})}(z_k - z_{k-1}), \quad k = 3, 4, \ldots. \tag{14}$$

To obtain $y(t_f, z_1)$ in (12), we use the CRK method (8), (9), and (10) for a system of delay differential equations. For a given mesh $\Delta = \{t_0, \ldots, t_n, \ldots, t_N = t_f\}$, let $h = \frac{t_f - t_0}{N}$. In each underlying mesh interval $[t_n, t_{n+1}]$, CRK formulas for (12) are as follows:

$$\begin{cases} k_{1,i} = f_1(t_n^i, X_i, Y_i, \eta_x(t_n^i - \tau_1), \eta_y(t_n^i - \tau_2)), & i = 1, \ldots, s, \\ k_{2,i} = f_2(t_n^i, X_i, Y_i, \eta_x(t_n^i - \tau_1), \eta_y(t_n^i - \tau_2)), & i = 1, \ldots, s, \\ X_i = x_n + h\sum_{j=1}^{s} a_{ij} k_{1,j}, & i = 1, \ldots, s, \\ Y_i = y_n + h\sum_{j=1}^{s} a_{ij} k_{2,j}, & i = 1, \ldots, s, \\ \eta_x(t_n + \theta h) = x_n + h\sum_{i=1}^{s} b_i(\theta) k_{1,i}, & 0 \leq \theta \leq 1, \\ \eta_y(t_n + \theta h) = y_n + h\sum_{i=1}^{s} b_i(\theta) k_{2,i}, & 0 \leq \theta \leq 1, \end{cases} \tag{15}$$

Note that at the endpoint of the interval, the stop condition must be checked. In the following algorithm, we describe an SCRK method for a time-delay TPBVP.

Example 1. Consider the following second-order delay boundary value problem:

$$\begin{cases} x''(t) = -\frac{1}{16}\sin x(t) - (t+1)x(t-1) + t, & 0 \le t \le 2, \\ x(t) = t - \frac{1}{2}, & t \le 0, \\ x(2) = -\frac{1}{2}. \end{cases} \tag{16}$$

With the new condition $x'(0) = z$, instead of solving (16), we need to solve a sequence of initial value problems of the form

$$\begin{cases} x''(t) = -\frac{1}{16}\sin x(t) - (t+1)x(t-1) + t, & 0 \le t \le 2, \\ x(t) = t - \frac{1}{2}, & t \le 0, \\ x'(0) = z. \end{cases} \tag{17}$$

Now, we try to make the value of $y(2, z)$ as close to $\beta = -\frac{1}{2}$ as possible by adjusting the value of z. Before that, by assuming $y(t) = x'(t)$, we turn the delay second-order system (17) into a delay first-order system as follows:

$$\begin{cases} x'(t) = y(t), & 0 \le t \le 2, \\ y'(t) = -\frac{1}{16}\sin x(t) - (t+1)x(t-1) + t, & \\ x(t) = t - \frac{1}{2}, & t \le 0, \\ y(0) = z. \end{cases} \tag{18}$$

We solve this problem by applying Algorithm 1. For this purpose, we use the explicit Runge–Kutta of discrete order $p = 4$ with the following coefficients:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \\ \frac{1}{2} & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Moreover, we set

$$b_1(\theta) = \frac{1}{2}\theta^2 + \frac{2}{3}\theta, \qquad b_3(\theta) = \frac{1}{3}\theta,$$
$$b_2(\theta) = \frac{1}{3}\theta, \qquad b_4(\theta) = \frac{1}{2}\theta^2 - \frac{1}{3}\theta.$$

Table 1 indicates a comparison between the approximate result of our SCRK method and the results obtained in [34].

Table 1: Approximation values of $x(t)$ in Example 1

	$x_n(1)$		$x_n(1.5)$		$x_n(2)$	
n	Ref. [34]	Proposed method	Ref. [34]	Proposed method	Ref. [34]	Proposed method
4	-1.854384	-1.983957	-1.719174	-1.884111	-0.499976	-0.499999
6	-2.018854	-2.032385	-1.896332	-1.922809	-0.499999	-0.500000
8	-2.066385	-2.052802	-1.946231	-1.939199	-0.499999	-0.500000
10	-2.078723	-2.063029	-1.959110	-1.947469	-0.499999	-0.500000
12	-2.081821	-2.068830	-1.962343	-1.952141	-0.500000	-0.500000

4 Design of SCRK method for an OCP with time delay in state variable

In this section, we first use the Pontryagin maximum principle to solve our delayed OCP. Then, for solving the final system, we describe an algorithm based on Algorithm 1. Through this section, by $PC^1([t_0, t_f], \mathbb{R}^n)$ we denote the class of continuous functions from $[t_0, t_f]$ into $\mathbb{R}^n$ whose first-order derivatives are piecewise continuous, and similarly, $PC([t_0, t_f], \mathbb{R}^n)$ denotes the class of piecewise continuous functions from $[t_0, t_f]$ into $\mathbb{R}^n$.

Consider the linear system with delay in the state variable

$$\begin{cases} \dot{x}(t) = Ax(t) + A_1x(t-\tau) + Bu(t), & t_0 \le t \le t_f, \\ x(t) = \phi(t), & t_0 - \tau \le t \le t_0, \end{cases} \tag{19}$$

where $u(t)$ in $PC([t_0, t_f], \mathbb{R}^n)$ and $x(t)$ in $PC^1([t_0 - \tau, t_f], \mathbb{R}^n)$ are the control and state variables, respectively. In fact, the parameter $\tau > 0$ is nonnegative and indicates the time delay. Furthermore, the initial state function $\phi(t)$ is continuous in $C([t_0 - \tau, t_0], \mathbb{R}^n)$, and finally, the matrices A, B, and A_1 are real constants with appropriate dimensions. For $t \in [t_0, t_f]$, our aim is to obtain, $u^*(t)$, the optimal control law minimizing the quadratic cost function

$$J = \frac{1}{2}\int_{t_0}^{t_f} (u^T(t)Ru(t) + x^T(t)Qx(t))dt + \frac{1}{2}x^T(t_f)Q_fx(t_f), \tag{20}$$

in which $R \in \mathbb{R}^{m\times n}$ is a positive definite matrix and Q and $Q_f \in \mathbb{R}^{n\times n}$ are positive semi-definite matrices.

For time-delay OCPs, it follows from [24] that the pontryagin maximum principle provides the necessary conditions of optimality for the problem (19) and (20) as follows:

$$\begin{cases} \dot{x}(t) = Ax(t) + A_1x(t-\tau) - BR^{-1}B^T\lambda(t), & t_0 \le t \le t_f, \\ \dot{\lambda}(t) = \begin{cases} -Qx(t) - A^T\lambda(t) - A_1^T\lambda(t+\tau), & t_0 \le t \le t_f - \tau, \\ -Qx(t) - A^T\lambda(t), & t_f - \tau < t \le t_f, \end{cases} \\ x(t) = \phi(t), \quad t_0 - \tau \le t \le t_0, \\ \lambda(t_f) = Q_fx(t_f). \end{cases} \tag{21}$$

The Hamiltonian function from which the above conditions are derived is

$$H(x,u,\lambda,t)=\lambda^T(t)[Ax(t)+Bu(t)+A_1x(t-\tau)+\frac{1}{2}x^T(t)Qx(t)+\frac{1}{2}u^T(t)Ru(t)], \tag{22}$$

where $\lambda(t)\in PC^1([t_0,t_f],\mathbb{R}^n)$ is called co-state vector. Moreover,

$$u^*(t)=-R^{-1}B^T\lambda(t), \tag{23}$$

for $t_0\le t\le t_f$, is the optimal control law. We recall that the system (21) is a TPBVP with both time-advance and time-delay terms. Unfortunately, in general, this problem does not have any analytical solution. Therefore, providing an efficient method for solving this difficult problem numerically is very important.

At first, we produce a sequence of TPBVP as

$$\begin{cases}\dot{x}^{(k)}(t)=-S\lambda^{(k)}(t)+Ax^{(k)}(t)+A_1x^{(k)}(t-\tau), & t_0\le t\le t_f,\\ \dot{\lambda}^{(k)}(t)=\begin{cases}-A^T\lambda^{(k)}(t)-Qx^{(k)}(t)-A_1^T\lambda^{(k-1)}(t+\tau), & t_0\le t\le t_f-\tau,\\ -A^T\lambda^{(k)}(t)-Qx^{(k)}(t), & t_f-\tau<t\le t_f,\end{cases}\\ x^{(k)}(t)=\phi(t), \quad t_0-\tau\le t\le t_0,\\ \lambda^{(k)}(t_f)=Q_fx^{(k)}(t_f),\\ x^{(0)}(t)\equiv 0, \quad \lambda^{(0)}(t)\equiv 0, \quad t_0\le t\le t_f,\end{cases} \tag{24}$$

where $S=BR^{-1}B^T$ and $k=1,2,\ldots$. Therefore,

$$u^{(k)}(t)=-R^{-1}B^T\lambda^{(k)}(t) \tag{25}$$

is the sequence of controls. Now, we are ready to obtain a closed-loop optimal control. We can define the co-state vector by

$$\lambda^{(k)}(t)=g^{(k)}(t)+P(t)x^{(k)}(t), \tag{26}$$

in which $g^{(k)}(t)\in\mathbb{R}^n$ is the kth adjoint vector and $P(t)\in\mathbb{R}^{n\times n}$ is an unknown function matrix with positive-definite property [45, 44].
Consider the following extended sequence of the TPBVP (24):

$$\begin{cases}\dot{x}^{(k)}(t)=[A-SP(t)]x^{(k)}(t)-Sg^{(k)}(t)+A_1x^{(k)}(t-\tau), & t_0\le t\le t_f,\\ \dot{g}^{(k)}(t)=\begin{cases}-P(t)A_1x^{(k)}(t-\tau)-[A-SP(t)]^Tg^{(k)}(t)\\ -A_1^TP(t+\tau)x^{(k-1)}(t+\tau)-A_1^Tg^{(k-1)}(t+\tau), & t_0\le t\le t_f-\tau,\\ -P(t)A_1x^{(k)}(t-\tau)-[A-SP(t)]^Tg^{(k)}(t), & t_f-\tau<t\le t_f,\end{cases}\\ x^{(k)}(t)=\phi(t), \quad t_0-\tau\le t\le t_0,\\ g^{(k)}(t_f)=0,\\ x^{(0)}(t)\equiv 0, \quad g^{(0)}(t)\equiv 0, \quad t_0\le t\le t_f.\end{cases} \tag{27}$$

We note that by substituting (26) into the first equation of (24), the kth optimal closed-loop system is constructed, which is the first equation of the system (27). Similarly, substituting (26) in the second equation of (24) and comparing the result with the derivative of (26), we obtain the second equation of the system (27). Also,

$$\begin{aligned} -\dot{P}(t) &= P(t)A + A^T P(t) - P(t)BR^{-1}B^T P(t) + Q, \\ P(t_f) &= Q_f, \end{aligned} \tag{28}$$

is a Riccati matrix differential equation.
Moreover, from (25) and (26), the sequence of controls is converted to

$$u^{(k)}(t) = -R^{-1}B^T(P(t)x^{(k)}(t) + g^{(k)}(t)), \quad k = 1, 2, \ldots. \tag{29}$$

The system (27) is similar to (11), except that (27) has advance terms in addition to the delay terms. Now, we want to use Algorithm 1 to solve this advance-delay TPBVP. By using the SCRK method, we have the following CRK iteration formula of (27) in the mesh interval $[t_n, t_{n+1}]$:

$$\begin{aligned} \eta_x^{(k)}(t_n + \theta h) =& x_n^{(k)} + h\sum_{i=1}^{s} b_i(\theta)[\Psi(t_n^i)(x_n^{(k)} + h\sum_{j=1}^{s} a_{ij}k_{1,j}) \\ & - S(g_n^{(k)} + h\sum_{j=1}^{s} a_{ij}k_{2,j}) + A_1\eta_x^{(k)}(t_n^i - \tau)], \quad t_0 \le t \le t_f, \end{aligned} \tag{30}$$

$$\eta_g^{(k)}(t_n + \theta h) = \begin{cases} g_n^{(k)} + h\sum_{i=1}^{s} b_i(\theta)[-\Psi^T(t_n^i)(g_n^{(k)} + h\sum_{j=1}^{s} a_{ij}k_{2,j}) \\ -P(t_n^i)A_1\eta_x^{(k)}(t_n^i - \tau) - A_1^T P(t_n^i + \tau)x^{(k-1)}(t_n^i + \tau) \\ -A_1^T g^{(k-1)}(t_n^i + \tau)], \qquad t_0 \le t \le t_f - \tau, \\ g_n^{(k)} + h\sum_{i=1}^{s} b_i(\theta)[-\Psi^T(t_n^i)(g_n^{(k)} + h\sum_{j=1}^{s} a_{ij}k_{2,j}) \\ -P(t_n^i)A_1\eta_x^{(k)}(t_n^i - \tau)], \qquad t_f - \tau < t \le t_f, \end{cases} \tag{31}$$

where $t_n^i = t_n + c_i h$, $\Psi(t_n^i) = A - SP(t_n^i)$, and $0 \le \theta \le 1$. Also,

$$\begin{cases} x^{(k)}(t) = \phi(t), & t_0 - \tau \le t \le t_0, \\ g^{(k)}(t_f) = 0, \end{cases} \tag{32}$$

are the known initial and final conditions.

As already mentioned, for the constant delay, if $0 \le c_i \le 1$ and $h \le \tau$, then $\eta(t_n^i - \tau)$ is known for any i. Hence, $\eta_x(t_n^i - \tau)$ in (30) and (31) is known, and there is no so-called overlapping. On the other hand, $x^{(k-1)}(t_n^i + \tau)$ and $g^{(k-1)}(t_n^i + \tau)$ are obtained from the previous iteration by the assumptions $x^{(0)}(t) \equiv 0$ and $g^{(0)}(t) \equiv 0$.

Theorem 3. Consider TPBVP (27).

i) Assume that the right-hand-side functions corresponding to $\dot{x}^{(k)}(t)$ and $\dot{g}^{(k)}(t)$ and the initial function $\phi(t)$ are C^p-continuous in their domains (p is the discrete order of the underlying CRK method). Then the sequences $\{\eta_x^{(k)}(t)\}$ and $\{\eta_g^{(k)}(t)\}$ obtained from CRK formulas (30) and (31) with initial and boundary conditions (32), converge uniformly to the solution of TPBVP (27).

ii) Under the assumptions of part (i), the sequences $\{u^{(k)}(t)\}$ and $\{J^{(k)}\}$, which are defined as follows

$$u^{(k)}(t) = -R^{-1}B^T[P(t)\eta_x^{(k)}(t) + \eta_g^{(k)}(t)], \tag{33}$$

$$J^{(k)} = \frac{1}{2}(\eta_x^{(k)}(t_f))^T Q_f \eta_x^{(k)}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} [(\eta_x^{(k)}(t))^T Q \eta_x^{(k)}(t) + (u^{(k)}(t))^T R u^{(k)}(t)]dt, \tag{34}$$

converge to optimal control $u^*(t)$ and the optimal value of objective function, J^*, respectively.

Proof. i) Consider the vector function F as follows:

$$F(t, x, g, u, v, w, z) = (\dot{x}(t), \dot{g}(t))^T, \quad t_0 \le t \le t_f,$$

and u, v, w, z denote delay and advance terms corresponding to the variables $x(t)$ and $g(t)$. Also, $\dot{x}(t)$ and $\dot{g}(t)$ are the functions defined in (27). Because F and ϕ are C^p-continuous functions and τ is a constant delay, according to Theorem 2, the sequences $\{\eta_x^{(k)}(t)\}$ and $\{\eta_g^{(k)}(t)\}$ from the CRK method are uniformly convergence to the exact solutions of (27).

ii) Suppose that $\{\eta_x^{(k)}(t)\}$ and $\{\eta_g^{(k)}(t)\}$ are solution sequences produced by the CRK method, which are convergence to $\hat{\eta}_x(t)$ and $\hat{\eta}_g(t)$ under the assumptions of part (i). We take the limit from the (33) as $k \to \infty$,

$$\begin{aligned}\hat{u}(t) &:= \lim_{k\to\infty} u^{(k)}(t) = -R^{-1}B^T[P(t)(\lim_{k\to\infty}\eta_x^{(k)}(t)) + \lim_{k\to\infty}\eta_g^{(k)}(t)] \\ &= -R^{-1}B^T[P(t)\hat{\eta}_x(t) + \hat{\eta}_g(t)].\end{aligned}$$

Since $\hat{\eta}_x(t)$ and $\hat{\eta}_g(t)$ are the exact solutions of necessary conditions (27), so $\hat{u}(t)$ is the optimal control $u^*(t)$.
Similarly, we take the limit from the (34) as follows:

$$\begin{aligned}\hat{J} &:= \lim_{k\to\infty} J^{(k)} \\ &= \lim_{k\to\infty}((\frac{1}{2}(\eta_x^{(k)}(t_f))^T Q_f \eta_x^{(k)}(t_f))\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{2}\lim_{k\to\infty}\Big(\int_{t_0}^{t_f}[(\eta_x^{(k)}(t))^T Q\eta_x^{(k)}(t)+(u^{(k)}(t))^T Ru^{(k)}(t)]dt\Big)\\
=&\frac{1}{2}\Big(\lim_{k\to\infty}(\eta_x^{(k)}(t_f))^T Q_f\Big(\lim_{k\to\infty}\eta_x^{(k)}(t_f)\Big)\\
&+\frac{1}{2}\int_{t_0}^{t_f}\Big[\Big(\lim_{k\to\infty}(\eta_x^{(k)}(t))^T\Big)Q\Big(\lim_{k\to\infty}\eta_x^{(k)}(t)\Big)\\
&+\Big(\lim_{k\to\infty}(u^{(k)}(t))^T\Big)R\Big(\lim_{k\to\infty}u^{(k)}(t)\Big)\Big]dt\\
=&\frac{1}{2}\hat{\eta}_x^T(t_f)Q_f\hat{\eta}_x(t_f)+\frac{1}{2}\int_{t_0}^{t_f}[\hat{\eta}_x^T(t)Q\hat{\eta}_x(t)+\hat{u}^T(t)R\hat{u}(t)]dt,
\end{aligned}
$$

so, $\hat{J}$ is the optimal value of the performance index J.

□

According to Theorem 3, it can be concluded that for enough iterations of the CRK method, for example, N iterations, where N depends on a given error criterion, we can obtain a suboptimal control as follows:

$$u^{(N)}(t) = -R^{-1}B^T[P(t)\eta_x^{(N)}(t)+\eta_g^{(N)}(t)]. \tag{35}$$

In this case, the continuous suboptimal state function is as $\eta_x(t)\cong\eta_x^{(N)}(t)$. To calculate a more accurate state function, the suboptimal control function resulting from equation (35), can be placed in (19), and we then solve the obtained initial value problem. Finally, by placing this pair of suboptimal control and state in the objective function, we have

$$
\begin{aligned}
J^{(N)} = \frac{1}{2}(\eta_x^{(N)}(t_f))^T Q_f\eta_x^{(N)}(t_f)+\frac{1}{2}\int_{t_0}^{t_f}((\eta_x^{(N)}(t))^T Q\eta_x^{(N)}(t)\\
+(u^{(N)}(t))^T Ru^{(N)}(t))dt.
\end{aligned}
\tag{36}
$$

For given $\varepsilon>0$, if the stop condition,

$$\Big|\frac{J^{(N)}-J^{(N-1)}}{J^{(N)}}\Big|<\varepsilon,$$

is satisfied, then the suboptimal control (35) will have the desired accuracy. Now, to implement the above method, we provide the following simple algorithm.

5 Numerical examples

Now, we are ready to present several examples for showing the efficiency of the SCRK method.

Algorithm 2 SCRK method for time-delay OCPs

Step 1. Solve $P(t)$ from (28).
Step 2. Put $k = 1$, $x^{(0)} \equiv 0$, and $g^{(0)} \equiv 0$. Then obtain a continuous approximation for $x^{(k)}(t)$ and $g^{(k)}(t)$ from the kth TPBVP (30), (31), and (32) with the shooting method (Algorithm 1).
Step 3. Let $N = k$ and obtain $u^{(N)}(t)$ from (35).
Step 4. Obtain $J^{(N)}$ from (36).
Step 5. If $\left|\frac{J^{(N)}-J^{(N-1)}}{J^{(N)}}\right| < \varepsilon$, then the procedure is complete, and go to the next step;
- else, let $k := k + 1$, and back to Step 2.

Step 6. Stop the algorithm and consider the output $u^{(N)}(t)$ as the desired closed-loop suboptimal control law.

Example 2. Consider the delay system

$$\begin{cases} \dot{x} = x(t) + u(t) + x(t-1), & t \geq 0, \\ x(t) = 1, & -1 \leq t \leq 0, \end{cases} \tag{37}$$

to minimize this quadratic cost functional

$$J = \frac{3}{2}x^2(2) + \frac{1}{2}\int_0^2 u^2(t)dt. \tag{38}$$

It follows from [4] that the exact solution for $u(t)$ is

$$u^*(t) = \begin{cases} \delta(e^{2-t} + (1-t)e^{1-t}), & 0 \leq t \leq 1, \\ \delta e^{2-t}, & 1 \leq t \leq 2, \end{cases} \tag{39}$$

and that $J^* = 3.1017$, where $\delta = -0.3932$. According to (37) and (38), we have $Q = 0,\ R = 1,\ Q_f = 3,\ A = 1,\ B = 1$, and $A_1 = 1$. Hence, (28) can be rewritten as

$$\begin{cases} \dot{p}(t) + 2p(t) - p^2(t) = 0, \\ p(2) = 3, \end{cases} \tag{40}$$

which has the unique solution

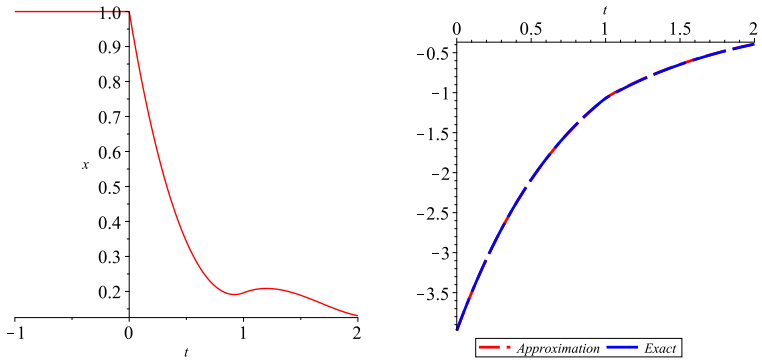
$$p(t) = \frac{6e^{4-2t}}{2 - 3(1 - e^{4-2t})}. \tag{41}$$

For the first time, Banks and Burns [4] proposed a numerical method to solve this problem based on averaging approximations. Then Pananisamy and Rao [38] solved it by using the Walsh functions. After that, Mirhosseini-Alizamini, the second author, and Heydari [32] used the variational iteration method. Furthermore, Jajarmi and Hajipour [20] employed a finite difference method for solving this problem. We apply our proposed method according to Algorithm 2 to this example. Comparison results of the optimal values

of J obtained by our proposed technique and other mentioned methods are listed in Table 2. The curves depicted from the obtained approximations for the state and control variables of problems (37) and (38) are shown in Figure 6.

Table 2: Value of cost functional for various methods in Example 2

Method	J
Banks and Burns [4]	3.0833
Pananismay and Rao [38]	3.0879
Mirhosseini-Alizamini, Effati, and Heydari [32]	3.1091
Jajarmi and Hajipour [20]	3.101717
Proposed SCRK method	3.101667
Optimal cost J^*	3.1017



(a) State variable $x(t)$ (b) Control variable $u(t)$

Figure 1: Simulated curves of (a) state variable and (b) approximation and exact values of control variable for Example 2

Now, we give another example.

Example 3. Consider the time-delay system

$$\begin{cases} \dot{x} = u(t) - x(t-1), & 0 \le t \le 1, \\ x(t) = 1, & -1 \le t \le 0, \end{cases} \tag{42}$$

to minimize this quadratic cost functional

$$J = \int_0^1 [\frac{1}{2}x^2(t) + \frac{1}{2}u^2(t)]dt. \tag{43}$$

Now, our aim is to obtain the optimal control, $u(t)$, subject to (42) that minimizes (43). Moreover, the Riccati equation for this example is

$$\begin{cases} \dot{p}(t) - p^2(t) + 1 = 0, \\ p(1) = 0, \end{cases} \tag{44}$$

and has the unique solution

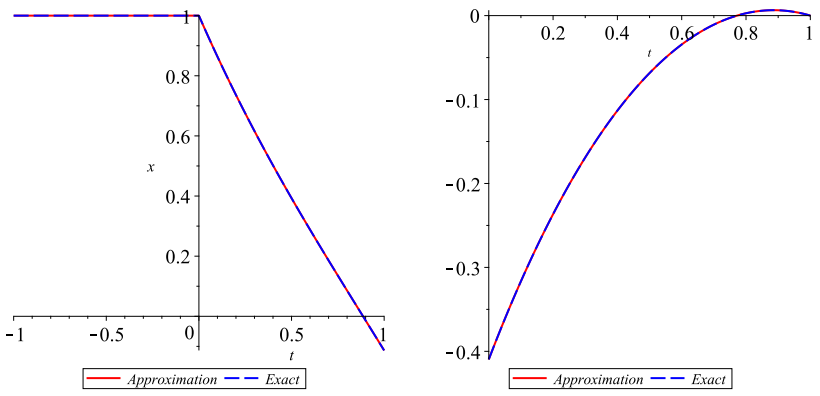
$$p(t) = -\tanh(t-1) \tag{45}$$

The exact solutions for $u(t)$ and $x(t)$ are, respectively, obtained as follows:

$$u^*(t) = 1 + \frac{1}{\cosh(1)}(\sinh(t-1) - \cosh(t)), \tag{46}$$

$$x^*(t) = \frac{1}{\cosh(1)}(\cosh(t-1) - \sinh(t)). \tag{47}$$

Moreover, it follows from [33] that the optimal value of cost functional is $J^* = 0.1480542786$. It can be shown that the approximate value of the cost functional calculated by the proposed SCRK method is equal to $J = 0.1480542988$. It is clear that the approximate value of J is very close to the optimal value. Also, we depict the simulation curves of the trajectory of $x(t)$, control variable $u(t)$, and their exact values in Figure 2.



(a) State variable $x(t)$ (b) Control variable $u(t)$

Figure 2: Approximation and exact values of state and control variables for Example 3

For the first time, Eller, Aggarwal, and Banks [10] presented the next example and then studied by other authors in [23, 37, 9, 40].

Example 4. Consider the linear time-varying delay system

$$\begin{cases} \dot{x} = x(t) + u(t) + x(t-1), & 0 \le t \le 2, \\ x(t) = 1, & -1 \le t \le 0, \end{cases} \tag{48}$$

to minimize this quadratic functional

$$J = \int_0^2 [x^2(t) + u^2(t)]dt. \tag{49}$$

Therefore, the Riccati equation for this example is

$$\begin{cases} \dot{p}(t) + 2p(t) - \frac{1}{2}p^2(t) + 2 = 0, \\ p(2) = 0, \end{cases} \tag{50}$$

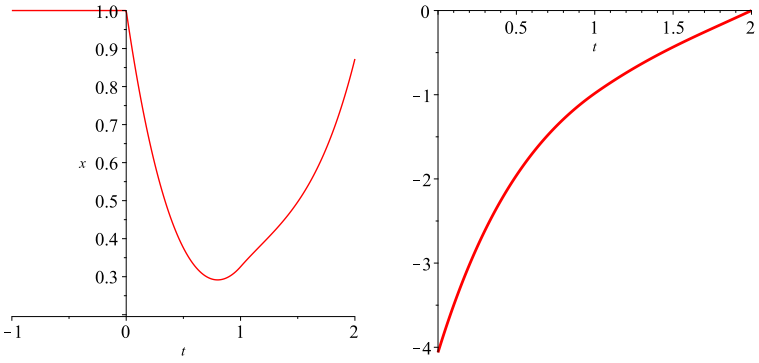
and the unique solution for this Riccati equation is

$$p(t) = 2 - 2\sqrt{2}\tanh(\sqrt{2}t + \tanh^{-1}(\frac{\sqrt{2}}{2}) - 2\sqrt{2}). \tag{51}$$

In Table 3, we compare the results of the suggested method with the reported results in [10, 23, 37, 9, 40, 21]. Figure 3 shows the approximate values of the state and control variables of the problem (48) and (49).

Table 3: Values of cost functional for various methods in Example 4

Method	J
Eller, Aggarwal, and Banks [10]	6.45
Dadebo and luus [9]	6.26775
Oh and Luus [37]	6.23711
Jamshidi and malek-Zavarei [23]	6.5
Santos and Sanchez-Diaz [40]	6.97
Jajarmi and Hajipour [21]	6.219615
Proposed SCRK method	6.200623



(a) State variable $x(t)$ (b) Control variable $u(t)$

Figure 3: Simulated curves of (a) state and (b) control variables for Example 4

Example 5. In this example, we want to minimize the cost functional

$$J = 5x_1^2(2) + \frac{1}{2}\int_0^2 u^2(t)dt, \tag{52}$$

with the following two-dimensional delay system:

$$\begin{cases} \dot{x}_1(t) = x_2(t), & 0 \le t \le 2, \\ \dot{x}_2(t) = -x_1(t) - x_2(t-1) + u(t), & 0 \le t \le 2, \\ x_1(0) = 10, \quad x_2(0) = 0, & -1 \le t \le 0. \end{cases} \tag{53}$$

Now, our aim is to obtain the optimal control $u^*(t)$ subject to (53) that minimizes (52). It follows from [4] that this problem has the exact solution

$$u^*(t) = \begin{cases} \delta \sin(2-t) + \frac{\delta}{2}(1-t)\sin(t-1), & 0 \le t \le 1, \\ \delta \sin(2-t), & 1 \le t \le 2, \end{cases} \tag{54}$$

in which the optimal cost is $J^* = 3.3991$ and $\delta = 2.5599$. In this two-dimensional example, we have $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $A_1 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $Q_f = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}$, and $R = 1$.

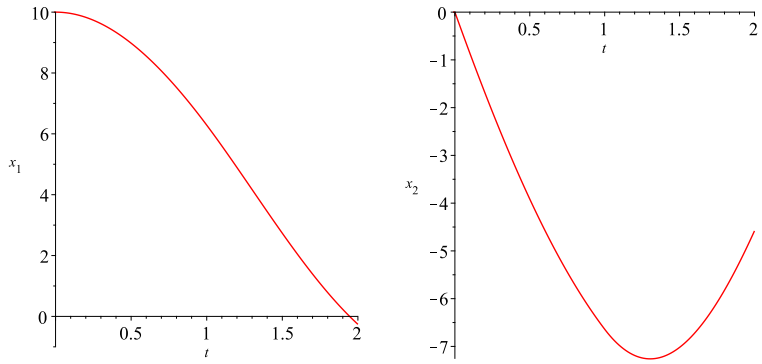
Thus, instead of the Riccati equation, we have a system consisting of four equations and four variables. After applying the proposed method to this example, we obtained the minimum value of $J = 3.3993$. In Table 4, the comparison of the result obtained with our proposed method and the result based on the techniques presented in [4, 28, 27, 15, 32] is shown. Also, Figures 3 and 5 show the corresponding state trajectories of $x_1(t)$, $x_2(t)$ and control variable $u(t)$, respectively.

Table 4: Cost functional values of various methods for Example 5

Method	J
Banks and Burns [4]	3.2587
Lee [28]	3.4827
Khellat [27]	3.43254
Haddadi, Ordokhani, and Razzaghi [15]	3.21663
Mirhosseini-Alizamini, Effati, and Heydari [32]	3.3991
Proposed SCRK method	3.3993

6 Conclusion

We employed The CRK method to solve a class of time-delay OCPs with delay in the state variable and with quadratic cost functional in this paper. At first, by employing the Pontryagin maximum principle for time-delay systems, the delay OCP was converted to a sequence of TPBVPs that have both delays and advance terms. After that, by applying the CRK method together with the shooting method, we constructed two sequences in which the delay



(a) State variable $x_1(t)$ (b) State variable $x_2(t)$

Figure 4: Simulated curves of state variables for Example 5

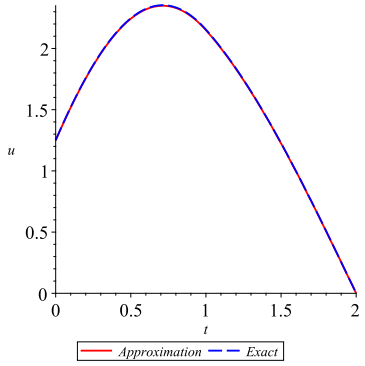


Figure 5: Control variable $u(t)$ for Example 5

and advance terms are known. Then we showed that by establishing the continuity condition, these sequences converge to the exact solution of the problem. The numerical results were presented to illustrate the high accuracy and efficiency of our proposed approach. Further research can be done on the extension of the SCRK method for solving time-delay OCPs with time-dependent delays in the control and state variables.

References

1. Baker, T. and Paul, C. *Parallel continuous Runge-Kutta methods and vanishing lag delay differential equations,* Adv. Comput. Math. 1 (3) (1993), 367–394.

2. Balochian, S. and Baloochian, H. *Social mimic optimization algorithm and engineering applications,* Expert Syst. Appl. 134 (2019), 178–191.

3. Banks, H. *Necessary conditions for control problems with variable time lags,* SIAM J. Control Optim. 6 (1) (1968), 9–47.

4. Banks, H. and Burns, J.A. *Hereditary control problems: Numerical methods based on averaging approximations,* SIAM J. Control Optim. 16 (2) (1978), 169–208.

5. Bellen, A. and Zennaro, M. *Numerical methods for delay differential equations, Numerical Mathematics and Scientific Computation*, Oxford University Press, Oxford, 2013.

6. Bouajaji, R., Abta, A., Laarabi, H. and Rachik, M. *Optimal control of a delayed alcoholism model with saturated treatment,* Differ. Equ. Dyn. Syst. (2021), 1–16.

7. Chen, L. and Wu, Z. *Stochastic optimal control problem in advertising model with delay,* J. Syst. Sci. Complex 33 (4) (2020), 968–987.

8. Chongyang, L., Zhaohua, G., Kok Lay, T. and Wang, S. *Modelling and optimal state-delay control in microbial batch process,* Appl. Math. Model, 89 (2021), 792–801.

9. Dadebo, S. and Luus, R. *Optimal control of time-delay systems by dynamic programming,* Optim. Control. Appl. Methods. 13 (1) (1992), 29–41.

10. Eller, D., Aggarwal, J. and Banks, H. *Optimal control of linear time-delay systems,* IEEE Trans. Automat. Contr. 14 (6) (1969), 678–687.

11. Ghomanjani, F., Farahi, M. H. and Gachpazan, M. *Optimal control of time-varying linear delay systems based on the bezier curves,* Int. J. Comput. Appl. Math. 33 (3) (2014), 687–715.

12. Göllmann, L. and Maurer, H. *Optimal control problems with time delays: Two case studies in biomedicine,* Math. Biosci. Eng. 15 (5) (2009), 11–37.

13. Gooran Orimi, A., Effati, S. and Farahi,M.H. *A suboptimal control of linear time-delay problems via dynamic programming,* IMA J. Math. Control Inform., 2022.

14. Guinn, T. *Reduction of delayed optimal control problems to nondelayed problems,* J. Optim. Theory Appl. 18 (3) (1976), 371–377.

15. Haddadi, N., Ordokhani, Y. and Razzaghi, M. *Optimal control of delay systems by using a hybrid functions approximation,* J. Optim. Theory Appl. 153 (2) (2012), 338–356.

16. Halanay, A., *Optimal controls for systems with time lag,* SIAM J. Control Optim. 6 (2) (1968), 215–234.

17. Hou, L., Chen, D. and He, C. *Finite-time h_∞ bounded control of networked control systems with mixed delays and stochastic nonlinearities*, Adv. Diff. Equ. 1 (2020), 1–23.

18. Huang, M., Gao, W. and Jiang, Z. P. *Connected cruise control with delayed feedback and disturbance: An adaptive dynamic programming approach,* Int. J. Adapt. Control Signal Process.33 (2) (2019), 356–370.

19. Ivanov, Anatoli F and Swishchuk, Anatoly V. *Optimal control of stochastic differential delay equations with application in economics,* International Journal of Qualitative Theory of Differential Equations and Applications 2 (2) (2008), 201–213.

20. Jajarmi, A. and Hajipour, M. *An efficient recursive shooting method for the optimal control of time-varying systems with state time-delay,* Appl. Math. Model. 40 (4) (2016), 2756–2769.

21. Jajarmi, A. and Hajipour, M. *An efficient finite difference method for the time-delay optimal control problems with time-varying delay,* Asian J. Control. 19 (2) (2017), 554–563.

22. Jamshidi, M. and Wang, C.M. *A computational algorithm for large-scale nonlinear time-delay systems,* IEEE Trans. Syst. Man Cybern. Syst. 1(1984), 2–9.

23. Jamshidi, M. and Zavarei, M. *Suboptimal design of linear control systems with time delay,* Proc. Inst. Electr. Eng. 119 (1972), 1743–1746.

24. Kharatishvili, GL. *The maximum principle in the theory of optimal processes involving delay,* Dokl. Akad. Nauk. 136 (1961), 39–42.

25. Kharatishvili, GL. *A maximum principle in extremal problems with delays,* Mathematical Theory of Control (1967), 26–34.

26. Kheirabadi, A. A. Mahmoudzadeh Vaziri, and S. Effati, *Linear optimal control of time delay systems via hermite wavelet,* Numer. Algebra Control Optim. 10 (2) (2020), 143.

27. Khellat, F. *Optimal control of linear time-delayed systems by linear Legendre multiwavelets,* J. Optim. Theory Appl. 143 (1) (2009), 107–121.

28. Lee, Y. *Numerical solution of time-delayed optimal control problems with terminal inequality constraints,* Optim. Control Appl. Methods. 14 (3) (1993), 203–210.

29. Malek-Zavarel, L. and Jamshidi, M. *Time-delay systems: analysis, optimization and applications*, Elsevier Science Inc., 1987.

30. Mansoori, M. and Nazemi, A. R. *Solving infinite-horizon optimal control problems of the time-delayed systems by Haar wavelet collocation method,* Int. J. Comput. Appl. Math. 35 (1) (2016), 97–117.

31. Mirhosseini-Alizamini, A. M. and Effati, S. *An iterative method for suboptimal control of a class of nonlinear time-delayed systems,* Int. J. Control. 92 (12) (2019), 2869–2885.

32. Mirhosseini-Alizamini, S. M., Effati, S. and Heydari, A. *An iterative method for suboptimal control of linear time-delayed systems,* Syst. Control. Lett.82 (2015), 40–50.

33. Mueller, T. *Optimal control of linear systems with time lag,* Third Annual Allerton Conf. on Circuit and System Theory. (1965), 339–345.

34. Nevers, K. D. and Schmitt, K. *An application of the shooting method to boundary value problems for second order delay equations,* Aust. J. Math. Anal. Appl. 36 (3) (1971), 588–597.

35. Oğuztöreli, M.N. *A time optimal control problem for systems described by differential difference equations,* SIAM J. Appl. Math., Series A: Control. 1 (3)(1963), 290–310.

36. Oğuztöreli, M.N. *Time-lag control systems,* Mathematics in Science and Engineering, 24 Academic Press, New York-London 1966.

37. Oh, S. and Luus, R. *Optimal feedback control of time-delay systems,* AIChE J. 22 (1) (1976), 140–147.

38. Palanisamy, K. and Prasada, R. *Optimal control of linear systems with delays in state and control via Walsh functions*, In IEE Proceedings D-Control Theory and Applications. 130 (1983), 300–312.

39. Santos, O., Mondié S. and Kharitonov, V. *Linear quadratic suboptimal control for time delays systems,* Int. J. Control. 82 (1) (2009), 147–154.

40. Santos, O. and Sanchez-Diaz, G. *Suboptimal control based on hill-climbing method for time delay systems,* IET Control Theory Appl. 1 (5) (2007), 1441–1450.

41. Silva, C.J., Cruz, C., Torres, D.F., Muñuzuri, A.P., Carballosa, R., Area, I., Nieto, J.J., Fonseca-Pinto, R., Passadouro, R., Santos, E.S.D., Abreu, W. *Optimal control of the Covid-19 pandemic: controlled sanitary deconfinement in portugal,* Scientific reports. 11 (1)(2021), 1–15.

42. Soleiman, MA and Ray, WH. *On the optimal control of systems having pure time delays and singular arcsI, Some necessary conditions for optimality,* Int. J. Control., 16 (5) (1972), 963–976.

43. Song, R., Xiao, W. and Wei, Q. *Multi-objective optimal control for a class of nonlinear time-delay systems via adaptive dynamic programming,* Soft Comput. 17 (11) (2013), 2109–2115.

44. Tang, G. and Luo, Z. *Suboptimal control of linear systems with state time-delay,* In IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028) , 5(1999), 104–109.

45. Tang, G.Y. and Zhao, Y.D. *Optimal control of nonlinear time-delay systems with persistent disturbances,* J. Optim. Theory Appl. 132 (2) (2007), 307–320.

46. Wright, K. *Some relationships between implicit Runge-Kutta, collocation and Lanczoѕ methods, and their stability properties,* BIT Numer. Math. 10 (2)(1970), 217–227.

47. Zennaro, M. *Natural continuous extensions of Runge-Kutta methods,* Math. Comp. 46 (173) (1986), 119–133.

How to cite this article
T. Jabbari-Khanbehbin, M. Gachpazan, S. Effati and S.M. Miri Shooting continuous Runge–Kutta method for delay optimal control problems. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 680-703. doi: 10.22067/ijnao.2022.78108.1166.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 704–718
DOI:10.22067/ijnao.2022.77745.1163
https://ijnao.um.ac.ir/

How to cite this article
Research Article

A new iteration method for solving space fractional coupled nonlinear Schrödinger equations

H. Aslani, D. Khojasteh Salkuyeh* and M. Taghipour

Abstract

A linearly implicit difference scheme for the space fractional coupled nonlinear Schrödinger equation is proposed. The resulting coefficient matrix of the discretized linear system consists of the sum of a complex scaled identity and a symmetric positive definite, diagonal-plus-Toeplitz, matrix. An efficient block Gauss–Seidel over-relaxation (BGSOR) method has been established to solve the discretized linear system. It is worth noting that the proposed method solves the linear equations without the need for any system solution, which is beneficial for reducing computational cost and memory requirements. Theoretical analysis implies that the BGSOR method is convergent under a suitable condition. Moreover, an appropriate approach to compute the optimal parameter in the BGSOR method is exploited. Finally, the theoretical analysis is validated by some numerical experiments.

AMS subject classifications (2020): 65F10, 81Q05, 81V99.

Keywords: The space fractional Schrödinger equations, Toeplitz matrix, Block Gauss-Seidel over-relaxation method, Convergence analysis.

---

* Corresponding author

Received 17 July 2022; revised 24 September 2022; accepted 25 September 2022

Hamed Aslani
Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran. email: hamedaslani525@gmail.com

Davod Khojasteh Salkuyeh
Faculty of Mathematical Sciences, and Center of Excellence for Mathematical Modelling Optimization and Combinational Computing (MMOCC), University of Guilan, Rasht, Iran. email: khojasteh@guilan.ac.ir

Mehran Taghipour
Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran. email: mtp20222@yahoo.com

1 Introduction

The Schrödinger equation is a crucial equation in quantum mechanics, a science that studies submicroscopic phenomena. It can arise from the Brownian path integral. In [6], the path integral method to the Lévy-α process was generalized, and the space fractional equations were derived.

Consider the space fractional coupled nonlinear Schrödinger (CNLS) equations

$$\begin{cases} \imath \mathrm{u}_t + \xi(-\Delta)^{\frac{\alpha}{2}}\mathrm{u} + \eta\left(|\mathrm{u}|^2 + \theta|\mathrm{v}|^2\right)\mathrm{u} = 0, \\ \imath \mathrm{v}_t + \xi(-\Delta)^{\frac{\alpha}{2}}\mathrm{v} + \eta\left(|\mathrm{v}|^2 + \theta|\mathrm{u}|^2\right)\mathrm{v} = 0, \end{cases} \qquad a_1 \le x \le a_2, \quad 0 < t < T. \tag{1}$$

Given the conditions of the initial boundary value as follows:

$$\begin{aligned} &\mathrm{u}(x,0) = \mathrm{u}_0(x), \qquad \mathrm{v}(x,0) = \mathrm{v}_0(x), \qquad a_1 \le x \le a_2, \\ &\mathrm{v}(a_1,t) = \mathrm{u}(a_2,t) = 0, \quad \mathrm{v}(a_1,t) = \mathrm{v}(a_2,t) = 0, \qquad 0 \le t \le T, \end{aligned}$$

where $\imath$ is the imaginary unit, $\xi > 0$, $\eta > 0$, $\theta \ge 0$ are some constants, and $1 < \alpha < 2$. In [5], the fractional Laplacian was designated as

$$(-\Delta)^{\frac{\alpha}{2}}\mathrm{u}(x,t) = \mathscr{H}^{-1}\left(|\phi|^{\alpha}\mathscr{H}(\mathrm{u}(x,t))\right),$$

in which $\mathscr{H}$ stands for the Fourier transform applied to the spatial variable x. Assuming that ${}_{-\infty}D_x^{\alpha}\mathrm{u}(x,t)$ and ${}_xD_{+\infty}^{\alpha}\mathrm{u}(x,t)$ are the left and right Riemann–Liouville fractional derivatives of order $\alpha \in \mathbb{R}^+$ given by

$${}_{-\infty}D_x^{\alpha}\mathrm{u}(x,t) = \frac{1}{\Gamma(n-\alpha)}\frac{\partial^n}{\partial x^n}\int_{-\infty}^{x}(x-\mu)^{n-1-\alpha}\mathrm{u}(\mu,t)d\mu,$$

$${}_xD_{+\infty}^{\alpha}\mathrm{u}(x,t) = \frac{1}{\Gamma(n-\alpha)}\frac{\partial^n}{\partial x^n}\int_{x}^{+\infty}(\mu-x)^{n-1-\alpha}\mathrm{u}(\mu,t)d\mu,$$

respectively, the Riesz fractional derivative can be calculated as

$$\frac{\partial^{\alpha}}{\partial|x|^{\alpha}}\mathrm{u}(x,t) = -(-\Delta)^{\frac{\alpha}{2}}\mathrm{u}(x,t) = -\frac{1}{2\cos\frac{\pi\alpha}{2}}\left[{}_{-\infty}D_x^{\alpha}\mathrm{u}(x,t) + {}_xD_{+\infty}^{\alpha}\mathrm{u}(x,t)\right].$$

In general, analyzing and understanding the behavior of the exact solutions of the space fractional CNLS equations is so challenging. In recent years, some numerical methods have been proposed to solve the CNLS equations. The difference method [12, 13, 11], the Crank–Nickelson scheme [1], and the collocation method [2] have been presented to solve the CNLS equations.

The discretization of the CNLS equations leads to the solution of the complex symmetric linear systems. The coefficient matrix consists of the sum of the symmetric positive definite, diagonal-plus-Toeplitz, matrix and the complex identity scaled matrix. Recently, Dai and Wu [4] developed a suited 2×2

linear system and employed the block Gauss–Seidel (BGS) iteration scheme to solve the resulting linear systems. Then they analyzed the convergence of the BGS scheme for the corresponding 2×2 linear system. In this work, we establish a fast block Gauss–Seidel over-relaxation (BGSOR) scheme for solving the two-by-two linear system that arises from the discretization of CNLS equations. Notably, the new method allows the corresponding systems to be solved without the need to compute the inverse of the coefficient matrices. Moreover, it should be pointed out that the BGS method can be regarded as a special case of the new method when the relaxation parameter is set to be one.

The arrangement of this work is as follows. In Section 2, the model problem will be studied, and a linearly implicit difference technique will be presented. Application, convergence theory, and finding the optimal parameter for the BGSOR method are proposed in Section 3. Section 4 is devoted to giving some numerical examinations. In Section 5, we finally made some conclusions.

2 Model problem and a linearly implicit difference scheme

The domain $\Omega = (a_1, a_2)\times(0, T)$ is divided into a uniform grid of mesh points (x_j, t_k), where

$$x_j = a_1 + jh, \quad h = \frac{a_2 - a_1}{m+1}, \quad 0 \le j \le m+1,$$

and

$$t_k = k\tau, \quad \tau = \frac{T}{n}, \quad 0 \le k \le n.$$

At grid points, the values of functions $\mathrm{u}(x,t), \mathrm{v}(x,t)$ are, respectively, denoted by $\mathrm{u}_j^k = \mathrm{u}(x_j, t_k)$, $\mathrm{v}_j^k = \mathrm{v}(x_j, t_k)$, and $\mathscr{U}_j^k, \mathscr{V}_j^k$ are the approximate solutions of (1).

The following equation gives a discrete approximation to $\frac{\partial^\alpha}{\partial|x|^\alpha}\mathrm{u}(x,t)$ [10]:

$$\frac{\partial^\alpha}{\partial|x|^\alpha}\mathrm{u}(x_j,t_k) = -\frac{\Psi_\alpha}{h^\alpha}\left[\sum_{l=0}^{\infty}\tilde{w}_k^{(\alpha)}\mathrm{u}(x_{j-l+1},t_k)+\sum_{l=0}^{\infty}\tilde{w}_k^{(\alpha)}\mathrm{u}(x_{j+l-1},t_k)\right]+\mathcal{O}(h^2), \tag{2}$$

where $\Psi_\alpha = \frac{1}{2\cos(\frac{\pi\alpha}{2})}$ and $\{\tilde{w}_k^\alpha\}$ is defined as follows:

$$\tilde{w}_0^{(\alpha)} = \frac{\alpha}{2}g_0^{(\alpha)}, \qquad \tilde{w}_l^{(\alpha)} = \frac{\alpha}{2}g_l^{(\alpha)} + \left(1-\frac{\alpha}{2}\right)g_{l-1}^{(\alpha)}, \qquad l \ge 1,$$

$$g_0^{(\alpha)} = 1, \quad g_l^{(\alpha)} = \left(1-\frac{\alpha+1}{l}\right)g_{l-1}^{(\alpha)}, \qquad l = 1, 2, \ldots.$$

Ortigueira [7] proposed the following fractional central difference operator:

$$\Delta_h^\alpha \mathrm{u}(x) = \sum_{l=-\infty}^{\infty} \hat{g}_l^{(\alpha)} \mathrm{u}(x - lh),$$

where

$$\hat{g}_l^{(\alpha)} = \frac{(-1)^k \Gamma(\alpha+1)}{\Gamma(\frac{\alpha}{2} - l + 1)\Gamma(\frac{\alpha}{2} + l + 1)}.$$

As stated in [7], the coefficient $\{\hat{g}_l^{(\alpha)}\}$ satisfies

$$\left| 2\sin\left(\frac{x}{2}\right) \right|^2 = \sum_{l=-\infty}^{\infty} \hat{g}_l^{(\alpha)} e^{\imath l x}, \qquad x \in \mathbb{R}.$$

When $\alpha > -1$, the recursive relations for $\{\hat{g}_l^{(\alpha)}\}$ are as follows:

$$\hat{g}_0^{(\alpha)} = \frac{\Gamma(\alpha+1)}{\Gamma^2(\alpha/2+1)}, \qquad \hat{g}_l^{(\alpha)} = \left(1 - \frac{\alpha+1}{\alpha/2+l}\right)\hat{g}_{l-1}^{(\alpha)}, \qquad l \geq 1;$$
$$\hat{g}_{-l}^{(\alpha)} = \hat{g}_l^{(\alpha)}, \qquad l \geq 1.$$

Lemma 1. [10] Assume that $u(x) \in \mathbb{C}^5(\mathbb{R})$ and that its all derivatives of order up to 5 belong to $L^1(\mathbb{R})$. Then, it holds

$$-\frac{\Delta_h^\alpha \mathrm{u}(x)}{h^\alpha} = \frac{\partial^\alpha \mathrm{u}(x)}{\partial |x|^\alpha} + \mathcal{O}(h^2). \tag{3}$$

From Lemma 1, it follows that

$$(-\Delta)^{\frac{\alpha}{2}} \mathrm{u}(x_j, t_k) = \frac{\Delta_h^\alpha \mathrm{u}(x)}{h^\alpha} + \mathcal{O}(h^2) = \frac{1}{h^\alpha} \sum_{l=1}^{M} \hat{g}_{j-l}^{(\alpha)}(x_j, t_k) + \mathcal{O}(h^2).$$

Now, we consider the following numerical scheme for solving (1) [12]:

$$\begin{aligned}
&\imath \frac{\mathscr{U}_j^{k+1} - \mathscr{U}_j^{k-1}}{2\tau} + \frac{\gamma}{h^\alpha} \sum_{l=1}^{m} \hat{g}_{j-l}^{(\alpha)} \left(\frac{\mathscr{U}_l^{k+1} + \mathscr{U}_l^{k-1}}{2} \right) + \rho\big(|\mathscr{U}_j^k|^2 + \beta|\mathscr{V}_j^k|^2\big) \\
&\qquad + \frac{\mathscr{U}_l^{k+1} + \mathscr{U}_l^{k-1}}{2} = 0, \\
&\imath \frac{\mathscr{V}_j^{k+1} - \mathscr{V}_j^{k-1}}{2\tau} + \frac{\gamma}{h^\alpha} \sum_{l=1}^{m} \hat{g}_{j-l}^{(\alpha)} \left(\frac{\mathscr{V}_l^{k+1} + \mathscr{V}_l^{k-1}}{2} \right) + \rho\big(|\mathscr{V}_j^k|^2 + \beta|\mathscr{U}_j^k|^2\big) \\
&\qquad + \frac{\mathscr{V}_l^{k+1} + \mathscr{V}_l^{k-1}}{2} = 0,
\end{aligned} \tag{4}$$

where $1 \le j \le m,\ 1 \le k \le n-1$. Another scheme should be provided for the numerical solution at $k = 1$. We consider the following scheme for this purpose (see [3]):

$$
\begin{aligned}
&\imath\frac{\mathscr{U}_j^1 - \mathscr{U}_j^0}{\tau} + \frac{\gamma}{h^\alpha}\sum_{l=1}^{m}\hat{g}_{j-l}^{(\alpha)}\mathscr{U}_l^{(1)} + \rho\big(|\mathscr{U}_j^0|^2 + \beta|\mathscr{V}_j^0|^2\big)\mathscr{U}_j^1 = 0,\\
&\imath\frac{\mathscr{V}_j^1 - \mathscr{V}_j^0}{\tau} + \frac{\gamma}{h^\alpha}\sum_{l=1}^{m}\hat{g}_{j-l}^{(\alpha)}\mathscr{V}_l^1 + \rho\big(|\mathscr{V}_j^0|^2 + \beta|\mathscr{U}_j^0|^2\big)\mathscr{V}_j^{(1)} = 0,\\
&\imath\frac{\mathscr{U}_j^1 - \mathscr{U}_j^0}{\tau} + \frac{\gamma}{h^\alpha}\sum_{l=1}^{m}\hat{g}_{j-l}^{(\alpha)}\left(\frac{\mathscr{U}_l^1 + \mathscr{U}_l^0}{2}\right)\\
&+ \rho\left(\frac{3}{2}|\mathscr{U}_j^1|^2 - \frac{1}{2}|\mathscr{U}_j^0|^2 + \beta\left(\frac{3}{2}|\mathscr{V}_j^{(1)}|^2 - \frac{1}{2}|\mathscr{V}_j^0|^2\right)\right)\frac{\mathscr{U}_j^1 + \mathscr{U}_j^0}{2} = 0,\\
&\imath\frac{\mathscr{V}_j^1 - \mathscr{V}_j^0}{\tau} + \frac{\gamma}{h^\alpha}\sum_{l=1}^{m}\hat{g}_{j-l}^{(\alpha)}\left(\frac{\mathscr{V}_l^1 + \mathscr{V}_l^0}{2}\right)\\
&+ \rho\left(\frac{3}{2}|\mathscr{V}_j^1|^2 - \frac{1}{2}|\mathscr{V}_j^0|^2 + \beta\left(\frac{3}{2}|\mathscr{U}_j^1|^2 - \frac{1}{2}|\mathscr{U}_j^0|^2\right)\right)\frac{\mathscr{V}_j^1 + \mathscr{V}_j^0}{2} = 0.
\end{aligned}
$$

The structure of the first and second difference equations in (4) is the same. Set

$$
\begin{aligned}
\mathscr{U}^{k+1} =& [\mathscr{U}_1^{k+1}, \dots, \mathscr{U}_m^{k+1}]^T, \quad b^{k+1} = [b_1^{k+1}, \dots, b_m^{k+1}]^T,\\
\mu =& \frac{\xi\tau}{h^\alpha}, \quad d_j^{k+1} = \eta\tau\big(|\mathscr{U}_j^k|^2 + \beta|\mathscr{V}_j^k|^2\big), \quad D^{k+1} = \operatorname{diag}(d_1^{k+1}, \dots, d_m^{k+1}).
\end{aligned}
$$

So, at each time step, we need to solve the following systems of linear equations:

$$
\begin{aligned}
A^{k+1}\mathscr{U}^{k+1} &= b^{k+1}, \qquad 1 \le k \le n-1,\\
B^{k+1}\mathscr{V}^{k+1} &= c^{k+1}, \qquad 1 \le k \le n-1,
\end{aligned}
\tag{5}
$$

where $A^{k+1} = T + D^{k+1} + \imath I$ and b^{k+1} is as follows:

$$
b^{k+1} = \begin{pmatrix}
\imath\mathscr{U}_1^{k-1} - \mu\sum_{l=1}^{m}\hat{g}_{1-l}^{(\alpha)}\mathscr{U}_l^{k-1} - d_1^{k+1}\mathscr{U}_1^{k-1}\\
\imath\mathscr{U}_2^{k-1} - \mu\sum_{l=1}^{m}\hat{g}_{2-l}^{(\alpha)}\mathscr{U}_l^{k-1} - d_2^{k+1}\mathscr{U}_2^{k-1}\\
\vdots\\
\imath\mathscr{U}_{m-1}^{k-1} - \mu\sum_{l=1}^{m}\hat{g}_{m-1-l}^{(\alpha)}\mathscr{U}_l^{k-1} - d_{m-1}^{k+1}\mathscr{U}_{m-1}^{k-1}\\
\imath\mathscr{U}_m^{k-1} - \mu\sum_{l=1}^{m}\hat{g}_{m-l}^{(\alpha)}\mathscr{U}_l^{k-1} - d_m^{k+1}\mathscr{U}_m^{k-1}
\end{pmatrix}.
$$

Moreover, T is the Toeplitz matrix, which has the following structure:

$$T=\mu\begin{pmatrix}\hat{g}_0^{(\alpha)} & \hat{g}_{-1}^{(\alpha)} & \cdots & \hat{g}_{2-m}^{(\alpha)} & \hat{g}_{1-m}^{(\alpha)}\\ \hat{g}_1^{(\alpha)} & \hat{g}_0^{(\alpha)} & \cdots & \hat{g}_{3-m}^{(\alpha)} & \hat{g}_{2-m}^{(\alpha)}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \hat{g}_{m-2}^{(\alpha)} & \hat{g}_{m-3}^{(\alpha)} & \cdots & \hat{g}_0^{(\alpha)} & \hat{g}_{-1}^{(\alpha)}\\ \hat{g}_{m-1}^{(\alpha)} & \hat{g}_{m-2}^{(\alpha)} & \cdots & \hat{g}_1^{(\alpha)} & \hat{g}_0^{(\alpha)}\end{pmatrix}. \tag{6}$$

Also, it should be noted that B^{k+1} and c^{k+1} can be obtained by changing the roles of $\mathscr{U}$ and $\mathscr{V}$ in A^{k+1} and b^{k+1}.

3 The BGSOR iteration method

To establish the BGSOR iteration method, we need to give some preliminaries. Let us first consider the iterative solution of the linear equation

$$A\mathscr{U}=b, \tag{7}$$

in which $A\in\mathbb{C}^{\ell\times\ell}$ is a nonsingular complex symmetric matrix as follows:

$$A=T+D+\imath I,$$

where T is the symmetric positive definite (SPD) and Toeplitz matrix designated in (6), $D=\operatorname{diag}(d_1,d_2,\ldots,d_\ell)$ with $d_i\geq 0$, $i=1,2,\ldots,\ell$, is the diagonal matrix, $U,b\in\mathbb{C}^{\ell}$. Let $U=x+\imath y$ and $b=f+\imath g$ be complex vectors, where $y,z,p,q\in\mathbb{R}^{\ell}$. So, the system can be rewritten as a particular form, namely,

$$\mathscr{A}\mathbf{x}\equiv\begin{pmatrix}-I & Q\\ Q & I\end{pmatrix}\begin{pmatrix}y\\ x\end{pmatrix}=\begin{pmatrix}f\\ g\end{pmatrix}\equiv\mathscr{P}, \tag{8}$$

where $Q=D+T$. We are now in a position to design a new method for solving (8).

To introduce the BGSOR iteration method, we consider the following decomposition for the coefficient matrix (8):

$$\mathscr{A}=(\omega\mathscr{D}-\mathscr{E})-(\mathscr{E}^T-(1-\omega)\mathscr{D})=:\mathscr{M}-\mathscr{N}, \tag{9}$$

where

$$\mathscr{D}=\begin{pmatrix}-I & 0\\ 0 & I\end{pmatrix},\qquad \mathscr{E}=\begin{pmatrix}0 & 0\\ -Q & 0\end{pmatrix},$$

and ω is a positive parameter, which is known as the relaxation parameter. Using the decomposition (9), the BGSOR iteration method is stated as

$$\mathscr{M}\mathbf{z}^{(k+1)}=\mathscr{N}\mathbf{z}^{(k)}+\mathscr{P},\quad k=0,1,2,\ldots,$$

where $\mathscr{M}$ and $\mathscr{N}$ are defined as (9), and $\mathbf{z}^{(k)} = (\mathbf{y}^{(k)}; \mathbf{x}^{(k)})$. Note that $\mathbf{y}^{(k)}$ and $\mathbf{x}^{(k)}$ are two M-vectors that stand for the iterations. Also, $\mathbf{z}^{(0)}$ is an arbitrary initial guess. Thereupon, the iterations take the following procedure:

$$\begin{cases} \mathbf{y}^{(k+1)} = \frac{1}{\omega}\left((\omega-1)\mathbf{y}^{(k)} + Q\mathbf{x}^{(k)} - f\right), \\ \mathbf{x}^{(k+1)} = \frac{1}{\omega}\left((\omega-1)\mathbf{x}^{(k)} + g - Q\mathbf{y}^{(k+1)}\right). \end{cases} \tag{10}$$

As can be seen, there is not any system solution in each iteration, and only two matrix-vector multiplication are needed. This can be very important because the new scheme requires insignificant computational efforts and just contains the matrix-vector multiplications. Furthermore, if $\omega = 1$, then the iteration scheme (10) reduces to

$$\begin{cases} \mathbf{y}^{(k+1)} = Q\mathbf{x}^{(k)} - f, \\ \mathbf{x}^{(k+1)} = g - Q\mathbf{y}^{(k+1)}, \end{cases} \tag{11}$$

which is presented in [4] and known as the BGS iteration method. Therefore, the BGS iteration method is a special case of the BGSOR iteration method.

Next, we investigate the convergence of the BGSOR method for solving (8), and then we obtain the optimal value of the relaxation parameter ω. In the following, we recall a result that will be useful later.

Lemma 2. [14] Suppose that the quadratic equation $x^2 - px + q = 0$, where p and q are real numbers. Both roots of the equation are less than one in modulus if and only if $|q| < 1$ and $|p| < 1 + q$.

Theorem 1. Consider $A = D + T + \imath I \in \mathbb{R}^{\ell\times\ell}$ as a matrix, where D and T are diagonal and Toeplitz SPD matrices, respectively. The necessary and sufficient condition for convergence of the BGSOR iteration method to the solution of (8) for any initial guess, is

$$\omega > \frac{1 + \mu_{\max}(Q)}{2},$$

where $\mu_{\max}(Q)$ is the largest eigenvalue of Q.

Proof. Let λ be an eigenvalue of the iteration matrix $\mathscr{G} = \mathscr{M}^{-1}\mathscr{N}$, and let $\mathbf{x} = [\mathbf{u}; \mathbf{v}]$ be the corresponding eigenvector. Without loss of generality, let $\lambda \neq 0$. Then

$$(\mathscr{D} - \omega\mathscr{E})^{-1}(\mathscr{E}^T - (1-\alpha\mathscr{D}))\mathbf{x} = \lambda\mathbf{x},$$

equivalently,

$$(1-\omega)\mathbf{u} - Q\mathbf{v} = -\lambda\omega\mathbf{u}, \tag{12}$$

$$(\omega-1)\mathbf{v} = \lambda(Q\mathbf{u} + \omega\mathbf{v}). \tag{13}$$

We can derive from (12) and the positive definiteness of Q that

$$\mathbf{v} = ((\lambda-1)\omega + 1)Q^{-1}\mathbf{u}. \tag{14}$$

Substituting (14) into (13), gives

$$-\lambda Q^2\mathbf{u} = ((\lambda-1)\omega+1)^2\mathbf{u}. \tag{15}$$

This shows that if μ is an eigenvalue of Q, then

$$\lambda\mu^2 = -\left((\lambda-1)\,\omega+1\right)^2 \tag{16}$$

$$= -\left(\lambda^2\omega^2+2\omega(1-\omega)\lambda+(\omega-1)^2\right). \tag{17}$$

From (17), we get

$$\lambda^2-\left(\frac{2\omega^2-2\omega-\mu^2}{\omega^2}\right)\lambda+(\frac{\omega-1}{\omega})^2=0. \tag{18}$$

Now it follows from Lemma 2 that $|\lambda|<1$ if and only if

$$|\omega-1|<\omega,$$
$$|2\omega^2-2\omega-\mu^2|7<2\omega^2-2\omega+1.$$

It is straightforward to see that $|\omega-1|<\omega$ is equivalent to $\omega>\frac{1}{2}$. By some easy manipulations, we can observe, whenever

$$(2\omega-1)^2>\mu^2, \tag{19}$$

the second inequality holds. The inequality (19) is ensured, if

$$|2\omega-1|>\mu \quad \text{or} \quad |2\omega-1|<-\mu,$$

equivalently,

$$\omega<\frac{1-\mu}{2} \quad \text{or} \quad \omega>\frac{1+\mu}{2}. \tag{20}$$

Evidently, the first inequality of (20) cannot be true. On the other hand, holding the second inequality of (20) ensures $\omega>\frac{1}{2}$, and then it completes the proof. □

In the following, we would like to find the optimal value of the relaxation parameter ω, denoted by ω^*. To do so, ω^* should be computed to minimize the spectral radius of the iteration matrix of the BGSOR method, that is,

$$\rho\left(\mathscr{G}_{\omega^*}\right)=\arg\min_{\omega>\frac{1+\mu_{\max}(Q)}{2}}\rho\left(\mathscr{G}_{\omega}\right).$$

To compute the optimal value of w, we state and prove the next theorem.

Theorem 2. Assume that the hypothesis of Theorem 1 are met. Then the optimal value of the relaxation parameter and the corresponding optimal convergence factor in the BGSOR iteration method are as follows:

$$\omega^* = \frac{1}{2}\left(1 + \sqrt{1+\rho^2(Q)}\right), \tag{21}$$

and

$$\rho\left(\mathscr{G}_{\omega^*}\right) = 1 - \frac{1}{\omega^*} = \left(\frac{\rho(Q)}{1+\sqrt{1+\rho^2(Q)}}\right)^2.$$

Proof. If λ is an eigenvalue of the iteration matrix $\mathscr{G}_\omega$, then $\lambda < 0$ or $\lambda \in \mathbb{C} \setminus \mathbb{R}$, according to (16). First, we consider the case $\lambda < 0$. So, there exists an eigenvalue μ of Q such that (18) holds true. The discriminant of this quadratic equation is

$$\Delta = \left(\frac{2\omega^2 - 2\omega - \mu^2}{\omega^2}\right)^2 - 4\left(\frac{\omega-1}{\omega}\right)^2,$$

and the roots of (18) are as follows:

$$\lambda_{1,2}(\omega) = \frac{2\omega^2 - 2\omega - \mu^2}{2\omega^2} \pm \frac{\sqrt{\Delta}}{2}.$$

From (16), we get

$$(\lambda - 1)\omega + 1 = \pm\mu\sqrt{-\lambda}. \tag{22}$$

Set

$$\begin{aligned} f_\omega(\lambda) =& (\lambda-1)\omega + 1 = \omega\lambda + 1 - \omega, \\ g(\lambda) =& \pm\,\mu\sqrt{-\lambda}. \end{aligned}$$

Clearly, the function f_ω passes through the point $(1,1)$, that is, $f_\omega(1) = 1$ and the slope of $f_\omega(\lambda)$ is ω. Figure 1 displays the points of intersections of the functions $f_\omega(\lambda)$ and $g(\lambda)$ for an arbitrary value of ω. This figure shows that by increasing ω, the maximum of absolute values of the abscissas of the points of intersection of the functions $f_\omega(\lambda)$ and $g(\lambda)$, that is, $\max\{\lambda_1, \lambda_2\}$, decrease, while $f_\omega(\lambda)$ gets tangent to $g(\lambda)$. In the tangent case, we have $\lambda_1 = \lambda_2$, and it indicates that $\Delta = 0$. From $\Delta = 0$, it is straightforward to verify that $\mu = 0$ or $4\omega^2 - 4\omega - \mu^2 = 0$. The case $\mu = 0$ is impossible, because of the positive definiteness of Q. Thus, $4\omega^2 - 4\omega - \mu^2 = 0$. This quadratic equation has two roots, as follows:

$$\omega_\pm = \frac{1}{2}\left(1 \pm \sqrt{1+\mu^2}\right).$$

Due to the condition $\omega > \frac{1+\mu_{\max}(Q)}{2}$, ω_- is not acceptable. So, we consider

$$\omega_+ = \frac{1}{2}\left(1 + \sqrt{1+\mu^2}\right),$$

and in this case, we have

$$\lambda_1 = \lambda_2 = \lambda_+ = \frac{1}{\omega_+} - 1.$$

Now suppose that $\omega > \omega_+$. In this case, the roots of the quadratic equation (18) are complex and conjugate, which are as follows:

$$\lambda_{1,2}(\omega) = \frac{-2\omega^2 + 2\omega + \mu^2}{2\omega^2} \pm \imath \frac{\sqrt{\Delta'}}{2},$$

where

$$\Delta' = 4\left(\frac{\omega - 1}{\omega}\right)^2 - \left(\frac{2\omega^2 - 2\omega - \mu^2}{\omega^2}\right)^2.$$

Then

$$|\lambda_{1,2}| = 1 - \frac{1}{\omega}.$$

By recalling that $\omega > \omega_+$ and having in mind that $w_+ > 1$, we have

$$1 - \frac{1}{\omega_+} < 1 - \frac{1}{\omega},$$

and this shows that ω_+ is the best choice for ω. On the other hand, the curve $g(\lambda) = \pm\rho(Q)\sqrt{-\lambda}$ serves an upper bound for each curve as $\pm\mu\sqrt{-\lambda}$, where $0 \leq \mu \leq \rho(Q)$. Summarizing the above results, we see that

$$\rho\left(\mathscr{G}_{\omega^*}\right) = \min_{\omega} \max_{\omega > \frac{1+\mu_{\max}}{2}} |1 - \frac{1}{\omega}| = 1 - \frac{1}{\omega^*} = \left(\frac{\rho(Q)}{1 + \sqrt{1 + \rho^2(Q)}}\right)^2,$$

where ω^* was considered as in (21). □

Remark 1. In Theorem 2, for computing ω^*, we need to compute $\rho(q)$. One may use a few iterations of the power method to compute $\lambda_{\max}(Q)$. On the other hand, because of positive definiteness of Q, we have

$$\rho(Q) = \lambda_{\max}(Q) = \|Q\|_2.$$

So, we can compute $\|Q\|_2$ instead of $\rho(Q)$. In practice, the `normest` command of MATLAB can be used to compute an estimation of $\|Q\|_2$.

4 Numerical experiments

This section is devoted to numerical experiments to evaluate the effectiveness of the BGSOR iteration scheme for solving linear systems (8). The numerical results of the proposed method are compared with those of the GMRES [8, 9] and the BGS methods. In all the test problems, we use the restart version of GMRES with a restarting number 10. The initial guess is assumed to be

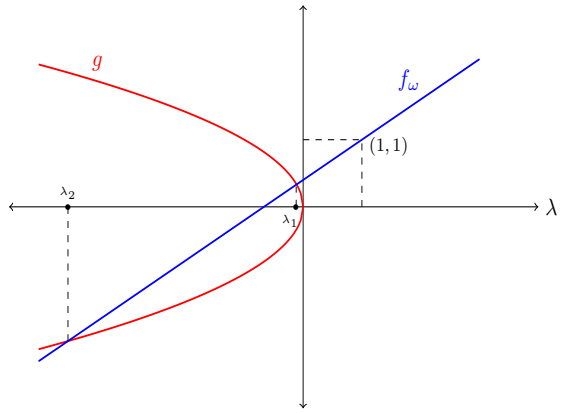


Figure 1: The graph of the functions $f_\omega(\lambda)$ and $g(\lambda)$.

a random vector, and iterations are terminated when

$$Res = \frac{\|r_k\|_2}{\|r_0\|_2} < 10^{-9},$$

where $r_k = \mathscr{P} - \mathscr{A}\mathbf{z}^{(k)}$ is the residual at the kth iteration or if the maximum number of iterations $maxit = 1000$ is exceeded. The terms "IT" and "CPU" in the tables refer to the total number of iterations and the elapsed CPU time in seconds for convergence, respectively. We comment that five runs were performed for each test, and then the average of CPU times and iterations are reported (The average of the iteration numbers were rounded). For the BGSOR method, the optimal parameter is computed according to the rule (21). The numerical results were carried out under MATLAB-R2017 on a laptop running Windows 10 and an Intel (R) Core(TM) i5-8265U CPU @ 1.60 GHz 8 GB.

Example 1. Let $\theta = 0$. The system (1) is then decoupled and becomes

$$\imath \mathrm{u}_t + (-\Delta)^{\frac{\alpha}{2}}\mathrm{u} + 2|\mathrm{u}|^2\mathrm{u} = 0,$$

when the initial value

$$\mathrm{u}(x, 0) = \mathrm{sech}(x) \cdot \exp(2\imath x),$$

is applied. In this example, the original problem was truncated in $[-20, 20]$. Set $u(-20, t) = u(20, t) = 0$. For this problem, we choose the parameters $\xi = 1.3$ and $\eta = 1.2$.

We set $m = 800, 1600, 3200, 6400$ and examine two values of α, $\alpha = 1.3, 1.6$. When $\alpha = 1.3$, we set $n = 4m$; otherwise, we choose $n = 6m$. The

Table 1: The optimal parameters ω^* for BGSOR method with $\alpha = 1.3$ and $n = 4m$ at $t = 2$ for Example 1.

ℓ	800	1600	3200	6400
ω^*	1.002	1.004	1.006	1.009

optimal values of the relaxation parameter in the BGSOR method for $\alpha = 1.3$ are given in Table 1, and the ones for $\alpha = 1.6$ are given in Table 3.

In Tables 2 and 4, we have listed the numerical results at $t = 2$. From these tables, we observe that the BGSOR method is superior to the examined methods in terms of both the iterations and the elapsed CPU times.

Table 2: Numerical results with $\alpha = 1.3$ and $n = 4m$ at $t = 2$ for Example 1.

Method	ℓ	800	1600	3200	6400
BGSOR	IT	5	5	5	5
	CPU	0.016	0.051	0.171	0.955
BGS	IT	5	6	6	7
	CPU	0.018	0.072	0.228	1.705
GMRES(10)	IT	6	7	7	7
	CPU	0.080	0.112	0.352	3.610

Example 2. For the following coupled system with $\theta \neq 0$:

$$\begin{cases} \imath \mathrm{u}_t + (-\Delta)^{\frac{\alpha}{2}} \mathrm{u} + 2\left(|\mathrm{u}|^2 + |\mathrm{v}|^2\right)\mathrm{u} = 0, \\ \imath \mathrm{v}_t + (-\Delta)^{\frac{\alpha}{2}} \mathrm{v} + 2\left(|\mathrm{v}|^2 + |\mathrm{v}|^2\right)\mathrm{v} = 0, \end{cases} \quad -20 \leq x \leq 20, 0 < t \leq 2. \tag{23}$$

We will use

$$\begin{cases} \mathrm{u}(x,0) = \operatorname{sech}(x + D_0) \cdot \exp(\imath \mathrm{v}_0 x), \quad \mathrm{v}(x,0) = \operatorname{sech}(x - D_0) \cdot \exp(-\imath \mathrm{v}_0 x), \\ \mathrm{u}(-20,0) = \mathrm{u}(20,0) = 0, \quad \mathrm{v}(-20,0) = \mathrm{v}(20,0) = 0, \end{cases} \tag{24}$$

as the initial conditions. In this case, we choose the parameters $D_0 = 1$, $\mathrm{v}_0 = 2$, $\xi = 1.4$, and $\eta = 1.2$.

The discretization of the coupled system of (23) leads to the solution of the linear systems of equations of the form (5). We assume that these coefficient matrices are A and B. These matrices have the same structure. Tables 5 and 7 show the optimal values of the relaxation parameter of A and B in the BGSOR method for different values of α and m.

In Tables 6 and 8, we report the results for the BGSOR, BGS, and GMRES(10) iterative methods at $t = 2$. These results clearly show that the BGSOR method leads to a faster overall convergence time than the other examined methods. Besides, the BGSOR method gets less iteration numbers.

Table 3: The optimal parameters ω^* for the BGSOR method with $\alpha = 1.6$ and $n = 6m$ at $t = 2$ for Example 1.

ℓ	800	1600	3200	6400
ω^*	1.010	1.022	1.050	1.108

Table 4: Numerical results with $\alpha = 1.6$ and $n = 6m$ at $t = 2$ for Example 1.

Method	ℓ	800	1600	3200	6400
BGSOR	IT	6	7	8	10
	CPU	0.018	0.068	0.311	2.015
BGS	IT	7	9	14	28
	CPU	0.022	0.093	0.571	4.462
GMRES(10)	IT	8	9	10	13
	CPU	0.112	0.185	0.235	6.941

Table 5: The optimal parameters ω^* of A and B for the BGSOR method with $\alpha = 1.3$ and $n = 4m$ at $t = 2$ for Example 2.

ℓ	800	1600	3200	6400
$\omega^*(A)$	1.002	1.004	1.006	1.008
$\omega^*(B)$	1.002	1.004	1.006	1.008

Table 6: Numerical results with $\alpha = 1.3$ and $n = 4m$ at $t = 2$ for Example 2.

Method	ℓ	800		1600		3200		6400	
		A	B	A	B	A	B	A	B
BGSOR	IT	5	5	5	5	5	5	5	5
	CPU	0.013	0.010	0.052	0.023	0.173	0.145	0.938	0.841
BGS	IT	5	5	6	6	6	6	7	7
	CPU	0.020	0.014	0.069	0.064	0.213	0.228	1.641	1.145
GMRES(10)	IT	6	6	7	7	7	7	8	8
	CPU	0.064	0.017	0.093	0.049	0.155	0.139	2.812	1.377

5 Conclusion

In this paper, the BGSOR scheme has been presented to solve the complex symmetric linear systems deriving from the discretization of the space fractional CNLS equation. We have analyzed the convergence theory of the method, and we have shown that the method is convergent under a suitable condition. The optimal value of the relaxation parameter and the rate of convergence factor for the BGSOR method were also provided. Our results have verified that the BGSOR method performs better than some existing methods.

Table 7: The optimal parameters ω^* of A and B for the BGSOR method with $\alpha = 1.6$ and $n = 6m$ at $t = 2$ for Example 2.

	800	1600	3200	6400
ℓ	1.010	1.022	1.050	1.122
$\omega^*(A)$	1.010	1.022	1.050	1.122
$\omega^*(B)$				

Table 8: Numerical results with $\alpha = 1.6$ and $n = 6m$ at $t = 2$ for Example 2.

Method	ℓ	800		1600		3200		6400	
		A	B	A	B	A	B	A	B
BGSOR	IT	6	6	7	7	9	9	10	10
	CPU	0.021	0.017	0.071	0.069	0.346	0.248	1.941	2.003
BGS	IT	7	7	10	10	15	15	35	35
	CPU	0.025	0.020	0.106	0.112	0.607	0.592	6.832	5.483
GMRES(10)	IT	8	8	9	9	11	11	13	13
	CPU	0.088	0.061	0.093	0.082	0.448	0.412	3.376	3.251

Acknowledgements

We would like to thank the referees for their helpful comments and suggestions.

This paper is dedicated to Prof. Faezeh Toutounian and Prof. Ali Vahidan Kamyad for their many contributions to numerical linear algebra and optimization theory.

References

1. Atangana, A. and Cloot, A.H. *Stability and convergence of the space fractional variable-order Schrödinger equation,* Adv. Diff. Equ. 2013 (2013) 80.

2. Amore, P., Fernández, F.M., Hofmann, C.P. and Sáenz, R.A. *Collocation method for fractional quantum mechanics,* J. Math. Phys. 51 (2010) 122101.

3. Chang, Y. and Chen, H. *Fourth-order finite difference scheme and efficient algorithm for nonlinear fractional Schrödinger equations,* Adv. Differ. Equ. 4 (2020) 1–8.

4. Dai, P. and Wu, Q. *An efficient block Gauss-Seidel iteration method for the space fractional coupled nonlinear Schrödinger equations,* Appl. Math. Lett. 117 (2021) 107–116.

5. Demengel, F. and Demengel, G. *Fractional sobolev spaces, in: Functional spaces for the theory of elliptic partial differential equations,* Springer, London, (2012) 179–228.

6. Laskin, N. *Fractional quantum mechanics and Lévy path integrals,* Phys. Lett. A 268 (2000) 298–305.

7. Ortigueira, M.D. *Riesz potential operators and inverses via fractional centred derivatives,* Int. J. Math. Math. Sci. (2006) 1–12.

8. Saad, Y. *Iterative methods for sparse linear systems*, Second edition PWS, New York, 1995.

9. Saad, Y. and Schultz, M.H. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. and Stat. Comput. 7 (1986) 856–869.

10. Sun, Z.Z. and Gao, G.h. *Fractional differential equations,* De Gruyter, 2020.

11. Wang, D., Xiao, A. and Yang, W. *Crank-Nicolson difference scheme for the coupled nonlinear Schrödinger equations with the Riesz space fractional derivative,* J. Comput. Phys. 242 (2013) 670–681.

12. Wang, D., Xiao, A. and Yang, W. *A linearly implicit conservative difference scheme for the space fractional coupled nonlinear Schrödinger equations,* J. Comput. Phys. 272 (2014) 644–655.

13. Wang, D., Xiao, A. and Yang, W. *Maximum-norm error analysis of a difference scheme for the space fractional CNLS,* Appl. Math. Comput. 257 (2015) 241–251.

14. Young, D.M. *Iterative solution or large linear systems,* Academic Press, New York, 1971.

How to cite this article
H. Aslani, D. Khojasteh Salkuyeh and M. Taghipour A new iteration method for solving space fractional coupled nonlinear Schrödinger equations. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 704-718. doi: 10.22067/ijnao.2022.77745.1163.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 3 (Special Issue), 2022, pp 719–738
DOI:10.22067/ijnao.2022.78220.1168
https://ijnao.um.ac.ir/

How to cite this article
Research Article

An efficient design for solving discrete optimal control problem with time-varying multi-delays

S.M. Abdolkhaleghzade, S. Effati* and S.A. Rakhshan

Abstract

The focus of this article is on the study of discrete optimal control problems (DOCPs) governed by time-varying systems, including time-varying delays in control and state variables. DOCPs arise naturally in many multi-stage control and inventory problems where time enters discretely in a natural fashion. Here, the Euler–Lagrange formulation (which are two-point boundary values with time-varying multi-delays) is employed as an efficient technique to solve DOCPs with time-varying multi-delays. The main feature of the procedure is converting the complex version of the discrete-time optimal control problem into a simple form of differential equations. Since the main problem is in discrete form, then the Euler–Lagrange equation changes to an algebraic system with initial and final conditions. The graphic representation of numerical simulation results shows that the proposed method can effectively and reliably solve DOCPs with time-varying multi-delays.

AMS subject classifications (2020): Primary 49M25; Secondary 93C55 , 37N35.

---

* Corresponding author

Received 15 August 2022; revised 3 October 2022; accepted 9 October 2022

Seyed Mostafa Abdolkhaleghzade
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. Email: mostafa.khaleghzade@gmail.com

Sohrab Effati
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran.
Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad, Mashhad, Iran Email: s-effati@um.ac.ir

Seyed Ali Rakhshan
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. Email: seyedalirakhshan@yahoo.com

Keywords: Discrete-time optimal control problem with time-varying delay, Euler–Lagrange equations, Pontryagin maximum principle.

1 Introduction

It is well known that discrete calculus is an important tool for describing natural phenomenas, which is expanded from classic calculus [18, 3, 7, 13]. By employing discrete calculus in optimal control problem (OCP), also well-known as discrete optimal control problem (DOCP), one can uniquely discover how to model natural phenomena. Discrete differential equations govern the dynamics of a dynamical system in a DOCP are one of the newest exciting mathematical challenges [16, 21, 22, 12].

The primary difference between continuous and discrete-time systems arises from the necessity to convert analog signals to digital values, as well as the time required for a computer system to calculate and execute the corrective action to the output.

A discrete time-control study on COVID-19 to address the quarantine and vital environmental loads has been explored in [2]. Mehraeen et al. [14] proposed an approach to obtain the optimal solutions based on the Hamilton–Jacobi–Isaacs equation for the discrete-time nonlinear system by using neural networks. In [11], the authors proposed an improved stability analysis method called a delay-mode-based functional method by weakening a condition in the Lyapunov–Krasovskii functional method. Adaptive dynamic programming as an effective intelligent control method has played an important role in seeking solutions for optimal control. Approximate dynamic programming techniques are used to solve the value function, and hence the optimal control policy, in discrete-time nonlinear OCPs having continuous state and action spaces; see([1, 5]). The adaptive dynamic programming algorithm was introduced in [20] for solving infinite-horizon undiscounted OCPs in discrete-time systems.

Discrete-time OCPs occur in many multi-stage control and scheduling problems, as may be expected. Originally, continuous-time OCPs can also be discretized suitably and subsequently formalized as discrete-time OCPs. Although due to the expansion of mathematical methods for solving continuous-time OCPs, this is not currently necessary. There are efficient methods for discrete-time OCPs in the literature.

To solve combined discrete-time OCPs and optimal parameter selection problems concerned with general constraints, a computational method was introduced in [4]. The DOCP for discrete-time linear system control constraint was investigated in [23], in which the control input is a one-dimensional variable whose range is contained in a bounded closed interval. Li, Teo, and Duan [10] considered a class of DDTOCP that contains nonlinear inequality constraints on both the state and control. In [19], authors discussed a

delay optimal tracking control for discrete-time systems with quadratic performance indexes when they are affected by persisting disturbances.

This paper presents a novel approach to solving DOCP, including time-varying delays. A general formula of the structure for DOCP with time-varying delays can be considered as follows:

$$J(u(\cdot)) = \sum_{k=k_0}^{k_f-1} F(x_k, u_k, k), \tag{1}$$

subject to time-varying delay in a dynamic system

$$x_{k+1} = G(x_k, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k), \qquad k_0 \le k \le k_f, \tag{2}$$

with initial conditions:

$$\begin{aligned} x_k &= \phi_k, \qquad k_0 - \tau_{k_0} \le k \le k_0, \\ u_k &= \Theta_k, \qquad k_0 - \omega_{k_0} \le k \le k_0, \end{aligned} \tag{3}$$

where $x(\cdot)$ is the state variable vector, $u(\cdot)$ is the control variable vector, k represents the time, F and G are given functionals, k_0 and k_f are fixed, ϕ_k and Θ_k are specific functions, $\tau_k \ge 0$ is delay function for state variable, and $\omega_k \ge 0$ is delay function for control variable.

Whenever the associated dynamic system of DOCP depends on prior information at a particular time, it can be considered that it is the DOCP with time-varying delays. A realistic distributed assumption, instead of a traditional point-wise assumption, creates interesting cases of delays [17]. Discrete derivatives are essential for explaining physical phenomena with memories, as previous information about predators and even prey can have an impact on birth rates, rather than the current model of predator-prey relationships and hereditary traits; thus, DOCP with time-varying delays is applied to all physical processes with realistic distribution assumptions and experiences [6]. As it can be seen, the problem satisfying (1)–(3) includes the delay system. A delay system is a specific form of partial differential equation with infinite dimensions. Therefore, these types of mathematical problems are very important in engendering and physical sciences.

Generally, time-delays systems can be found in control systems, lasers, traffic models, metal cutting, transmission lines, epidemiology, cell cycle, protein, production population dynamics, and neuroscience. Therefore, it is important to propose a beneficial method for solving time-delays systems. Also, solving optimal control problems is complicated in normal mode, especially in non-linear modes. As a result, they become much more complicated in modes whose systems have time delays. So it is very valuable to work on such issues.

As a review of this paper, the framework of this paper is organized as follows:

Section 2 includes the proposed technique for solving DOCP with time-varying delays in state and control variables. Finally, Section 3 contains a number of numerical examples that demonstrate the model's effectiveness. We conclude in the last section.

2 Main results

There are several kinds of variational problems in calculus [9, 8]. Here, we propose the two-boundary value problem based on classical Euler–Lagrange equations to solve DOCP with time-varying delay. Therefore, we review some necessary definitions and theoretical concepts to derive our efficient technique.

Definition 1. Suppose that x_k (respectively, x_{k+1}) takes on variations δx_k (respectively, δx_{k+1}) from their optimal values $\bar{x}_k$ (respectively, $\bar{x}_{k+1}$) satisfying

$$x_k = \bar{x}_k + \delta x_k, \qquad x_{k+1} = \bar{x}_{k+1} + \delta x_{k+1}. \tag{4}$$

Now with these variations, the performance index (1) becomes

$$\begin{aligned} \hat{J} = J(\bar{x}_{k_o}, k_o) &= \sum_{k=k_0}^{k_f-1} F(\bar{x}_k, \bar{x}_{k+1}, k) \\ J = J(x_{k_o}, k_o) &= \sum_{k=k_0}^{k_f-1} F(\bar{x}_k + \delta x_k, \bar{x}_{k+1} + \delta x_{k+1}, k). \end{aligned} \tag{5}$$

Definition 2. The first variation δJ is the first order approximation of the increment $\Delta J = J - \hat{J}$. So, applying the Taylor series expansion of (5), we obtain

$$\delta J = \sum_{k=k_0}^{k_f-1} \frac{\partial F(\bar{x}_k, \bar{x}_{k+1}, k)}{\partial \bar{x}_k} \delta x_k + \frac{\partial F(\bar{x}_k, \bar{x}_{k+1}, k)}{\partial \bar{x}_{k+1}} \delta x_{k+1}. \tag{6}$$

Theorem 1. For x_k to be a contender for an optimum, the first variation of J must be zero on x_k, that is, $\delta J(x_k, \delta x_k) = 0$ for all admissible values of δx_k. This is a necessary condition. As a sufficient condition for minimum, we have the second variation $\delta^2 J > 0$, and for maximum, $\delta^2 J < 0$.

Proof. The researchers can consider the proof in detail in (see [8, p. 37]). □

Lemma 1. Suppose that g_k is a function in which the domain and range are each a discrete set of values. If g_k is a discrete function satisfying

$$\sum_{k=k_0}^{k_f} g_k \delta x_k = 0, \tag{7}$$

where the function δx_k is discrete in the interval $[k_0, k_f]$, then $g_k = 0$ for every $k \in [k_0, k_f]$.

Proof. Let $g_{k_0} \neq 0$ for some k_0. Assume that $\delta x_s = 0$ if $s \neq k_0$ and $\delta x_{k_0} = 1$. Then δ is a discrete function. In addition, $\sum_k \delta x_k g_k = g_{k_0} = 0$, which is a contradiction. □

Definition 3 (Gateaux derivative)**.** Suppose that X and Y are locally convex topological vector spaces, $U \subset X$ is open, and $f : X \to Y$. The Gateaux differential of f at $u \in U$ in the direction $\psi \in X$, denoted by $df(u;\psi)$, is defined as

$$df(u;\psi) = \lim_{k \to 0} \frac{f(u + k\psi) - f(u)}{k} = \frac{d}{dk} f(u + k\psi)\big|_{k=0}. \tag{8}$$

If the limit (8) exists for every $\psi \in X$, then the function f is called Gateaux differentiable at u [15].

This paper investigates a structured strategy for finding the necessary optimality condition for the problem (1)–(3). It means that the DOCP with time-varying delays is analyzed in order to find the optimal control $u(\cdot)$ with the minimum performance index (1). Therefore, we investigate the necessary optimally condition of the DOCP with time-varying delays as follows.

Theorem 2 (Necessary conditions for DOCP with time-varying delays)**.** Suppose that the DOCP defined by (1)–(3) with k_0, x_{k_0}, and k_f is fixed. Also, suppose that X is a locally convex topological vector spaces, and that $U \subset X$ is an open subset. In addition, assume that the following regularity conditions are satisfied:

R1. $x_k, x_{k-\tau_k} \in X$;

R2. $u_k, u_{k-\omega_k} \in U$;

R3. $\tau_k : \mathbb{N} \longrightarrow \mathbb{N}$ and $\omega_k : \mathbb{N} \longrightarrow \mathbb{N}$ are natural-valued functions, and $\tau(\cdot), \omega(\cdot) \geq 0$;

R4. $k_0 \in \mathbb{Z}$, $k_f \in \mathbb{Z}$, $\Theta : \mathbb{Z} \longrightarrow \mathbb{Z}$, and $\phi : \mathbb{Z} \longrightarrow \mathbb{Z}$ are known;

R5. J is Gateaux differentiable at u_k;

R6. F and G are locally convex topological vector spaces.

Then any solution $u(\cdot) \in U$ must satisfy the following conditions:

N1. The state dynamics, for $k_0 \leq k \leq k_f$:

$$x_{k+1} = G\Big(x_k, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k\Big), \qquad k_0 \leq k \leq k_f. \tag{9}$$

N2. The adjoint dynamics:

$$\begin{cases} \dfrac{\partial F}{\partial x_k} - \lambda_k + \lambda_{k+1}^T \dfrac{\partial G}{\partial x_k} + \lambda_{k+1}^T \psi_k = 0, \; k > \tau_k, \\ \dfrac{\partial F}{\partial x_k} - \lambda_k + \lambda_{k+1}^T \dfrac{\partial G}{\partial x_k} = 0, \qquad\qquad O.W., \end{cases} \tag{10}$$

where

$$\begin{aligned} &\psi_k = \frac{\partial G}{\partial x_{k-\tau_k}}, \\ &F = F(x_k, u_k, k), \\ &G = G\Big(x_k, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k\Big). \end{aligned}$$

N3. The optimal control dynamics:

$$\begin{cases} \dfrac{\partial F}{\partial u_k} + \lambda_{k+1}^T \dfrac{\partial G}{\partial u_k} + \lambda_{k+1}^T \eta_k = 0 \qquad , k > \omega_k, \\ \dfrac{\partial F}{\partial u_k} + \lambda_{k+1}^T \dfrac{\partial G}{\partial u_k} = 0, \qquad\qquad O.W., \end{cases} \tag{11}$$

where $\eta_k = \dfrac{\partial G}{\partial u_{k-\omega_k}}$.

N4. The Boundary conditions:

$$x_k = \phi_k, \qquad k \le k_0, \tag{12}$$

$$u_k = \Theta_k, \qquad k \le k_0, \tag{13}$$

$$\frac{\partial \mathcal{L}\big(x_{k-1}, x_{k-\tau_{k-1}-1}, x_k, u_{k-1}, u_{k-\omega_{k-1}-1}, \lambda_k\big)}{\partial x_k}\Big|_{k=k_f} = 0. \tag{14}$$

Proof. The required condition for the DOCP with time-varying delays is found by utilizing the variational method. Suppose that

$$\bar{J}(u(\cdot)) = \sum_{k=k_0}^{k_f-1} F(x_k, u_k, k) + \lambda_{k+1}^T \Big(G(x_k, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k) - x_{k+1}\Big), \tag{15}$$

where $\lambda(\cdot)$ is the Lagrange multiplier. Let δx_k, δu_k, $\delta x_{k-\tau_k}$, $\delta u_{k-\omega_k}$, and $\delta\lambda_k$ be the variation of x_k, u_k, $x_{k-\tau_k}$, $u_{k-\omega_k}$, and λ_k, respectively. We then define a family of curves as follows:

$$\begin{cases} x_k^\epsilon = x_k + \epsilon\delta x_k, \\ x_{k+1}^\epsilon = x_{k+1} + \epsilon\delta x_{k+1}, \\ x_{k-\tau_k}^\epsilon = x_{k-\tau_k} + \epsilon\delta x_{k-\tau_k}, \\ u_k^\epsilon = u_k + \epsilon\delta u_k, \\ u_{k-\omega_k}^\epsilon = u_{k-\omega_k} + \epsilon\delta u_{k-\omega_k}, \\ \lambda_{k+1}^\epsilon = \lambda_{k+1} + \epsilon\delta\lambda_{k+1}. \end{cases} \tag{16}$$

Let

$$\begin{aligned} L(k) &= L\big(x_k, x_{k+1}, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k\big) \\ &= F(x_k, u_k, k) + \lambda_{k+1}^T\Big(G(x_k, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k) - x_{k+1}\Big), \end{aligned} \tag{17}$$

and

$$\begin{aligned} L^\epsilon(k) &= L(x_k^\epsilon, x_{k+1}^\epsilon, x_{k-\tau_k}^\epsilon, u_k^\epsilon, u_{k-\omega_k}^\epsilon, k) \\ &= F(x_k^\epsilon, u_k^\epsilon, k) + (\lambda_{k+1}^\epsilon)^T\Big(G(x_k^\epsilon, x_{k-\tau_k}^\epsilon, u_k^\epsilon, u_{k-\omega_k}^\epsilon, k) - x_{k+1}^\epsilon\Big). \end{aligned} \tag{18}$$

Also note that according to Definition 3, we get

$$\begin{aligned} \delta\bar{J}(u_k;\delta u_k) &= \lim_{\epsilon\to 0}\frac{J(u_k+\epsilon\delta u_k) - J(u_k)}{\epsilon} \\ &= \sum_{k=k_0}^{k_f}\lim_{\epsilon\to 0}\frac{L^\epsilon(k) - L(k)}{\epsilon} = \sum_{k_0}^{k_f}\frac{d}{d\epsilon}L^\epsilon(k)\mid_{\epsilon=0}. \end{aligned} \tag{19}$$

The variational of functional $\bar{J}(u(\cdot))$ is given as

$$\begin{aligned} \delta\bar{J}(u(\cdot)) = \sum_{k=k_0}^{k_f-1}\frac{d}{d\epsilon}L^\epsilon(k)\mid_{\epsilon=0} = \sum_{k=k_0}^{k_f-1}\Big[&\frac{\partial L^\epsilon(k)}{\partial x_k^\epsilon}\frac{dx_k^\epsilon}{d\epsilon} + \frac{\partial L^\epsilon(k)}{\partial x_{k+1}^\epsilon}\frac{dx_{k+1}^\epsilon}{d\epsilon} \\ &+ \frac{\partial L^\epsilon(k)}{\partial x_{k-\tau_k}^\epsilon}\frac{dx_{k-\tau_k}^\epsilon}{d\epsilon} + \frac{\partial L^\epsilon(k)}{\partial u_k^\epsilon}\frac{du_k^\epsilon}{d\epsilon} \\ &+ \frac{\partial L^\epsilon(k)}{\partial u_{k-\omega_k}^\epsilon}\frac{du_{k-\omega_k}^\epsilon}{d\epsilon} + \frac{\partial L^\epsilon(k)}{\partial \lambda_{k+1}^\epsilon}\frac{d\lambda_{k+1}^\epsilon}{d\epsilon}\Big]\mid_{\epsilon=0}. \end{aligned} \tag{20}$$

Also, according to (16), we have

$$\begin{aligned} &\frac{dx_k^\epsilon}{d\epsilon} = \delta x_k, \quad \frac{dx_{k+1}^\epsilon}{d\epsilon} = \delta x_{k+1}, \quad \frac{du_k^\epsilon}{d\epsilon} = \delta u_k, \\ &\frac{d\lambda_{k+1}^\epsilon}{d\epsilon} = \delta\lambda_{k+1}, \quad \frac{dx_{k-\tau_k}^\epsilon}{d\epsilon} = \delta x_{k-\tau_k}, \frac{du_{k-\omega_k}^\epsilon}{d\epsilon} = \delta u_{k-\omega_k}. \end{aligned} \tag{21}$$

Therefore,

$$\delta\bar{J}(u(\cdot)) = \sum_{k=k_0}^{k_f-1} \big[\frac{\partial L(k)}{\partial x_k}\delta x_k + \frac{\partial L(k)}{\partial x_{k+1}}\delta x_{k+1} + \frac{\partial L(k)}{\partial x_{k-\tau_k}}\delta x_{k-\tau_k} + \frac{\partial L(k)}{\partial u_k}\delta u_k + \frac{\partial L(k)}{\partial u_{k-\omega_k}}\delta u_{k-\omega_k} + \frac{\partial L(k)}{\partial \lambda_{k+1}}\delta\lambda_{k+1}\big]. \tag{22}$$

Also, we get from (17) that

$$\begin{aligned}
&\frac{\partial L(k)}{\partial x_k} = \frac{\partial F(x_k,u_k,k)}{\partial x_k} + \lambda_{k+1}^T \frac{\partial G(x_k,x_{k-\tau_k},u_k,u_{k-\omega_k},k)}{\partial x_k},\\
&\frac{\partial L(k)}{\partial x_{k-\tau_k}} = \lambda_{k+1}^T \frac{\partial G(x_k,x_{k-\tau_k},u_k,u_{k-\omega_k},k)}{\partial x_{k-\tau_k}},\\
&\frac{\partial L(k)}{\partial u_k} = \frac{\partial F(x_k,u_k,k)}{\partial u_k} + \lambda_{k+1}^T \frac{\partial G(x_k,x_{k-\tau_k},u_k,u_{k-\omega_k},k)}{\partial u_k},\\
&\frac{\partial L(k)}{\partial u_{k-\omega_k}} = \lambda_{k+1}^T \frac{\partial G(x_k,x_{k-\tau_k},u_k,u_{k-\omega_k},k)}{\partial u_{k-\omega_k}},\\
&\frac{\partial L(k)}{\partial \lambda_{k+1}} = G(x_k,x_{k-\tau_k},u_k,u_{k-\omega_k},k) - x_{k+1},\\
&\frac{\partial L(k)}{\partial x_{k+1}} = -\lambda_{k+1}.
\end{aligned} \tag{23}$$

Also, we can rearrange the term, including x_{k+1} in (22), as follows:

$$\begin{aligned}
&\sum_{k=k_0}^{k_f-1} \frac{\partial \mathcal{L}(x_k,x_{k-\tau_k},x_{k+1},u_k,u_{k-\omega_k},\lambda_{k+1})}{\partial x_{k+1}}\delta x_{k+1}\\
&\quad= \frac{\partial \mathcal{L}\big(x_{k_f-1},x_{k_f-\tau_{k-1}-1},x_{k_f},u_{k_f-1},u_{k_f-\omega_{k-1}-1},\lambda_{k_f}\big)}{\partial x_{k_f}}\delta x_{k_f}\\
&\qquad- \frac{\partial \mathcal{L}\big(x_{k_0-1},x_{k_0-\tau_{k-1}-1},x_{k_0},u_{k_0-1},u_{k_0-\omega_{k-1}-1},\lambda_{k_0}\big)}{\partial x_{k_0}}\delta x_{k_0}\\
&\qquad+ \sum_{k=k_0}^{k_f-1} \frac{\partial \mathcal{L}\big(x_{k-1},x_{k-\tau_{k-1}-1},x_k,u_{k-1},u_{k-\omega_{k-1}-1},\lambda_k\big)}{\partial x_k}\delta x_k\\
&\quad= \big[\frac{\partial \mathcal{L}(x_{k-1},x_{k-\tau_{k-1}-1},x_k,u_{k-1},u_{k-\omega_{k-1}-1},\lambda_k)}{\partial x_k}\delta x_k\big]\,\big|_{k=k_0}^{k=k_f}\\
&\qquad+ \sum_{k=k_0}^{k_f-1} \frac{\partial \mathcal{L}\big(x_{k-1},x_{k-\tau_{k-1}-1},x_k,u_{k-1},u_{k-\omega_{k-1}-1},\lambda_k\big)}{\partial x_k}\delta x_k.
\end{aligned} \tag{24}$$

We then conclude the first variation of $\bar{J}(u(\cdot))$ from equations (22)–(24) as

$$\delta\bar{J}(u(\cdot)) = \Big[\frac{\partial\mathcal{L}\big(x_{k-1}, x_{k-\tau_{k-1}-1}, x_k, u_{k-1}, u_{k-\omega_{k-1}-1}, \lambda_k\big)}{\partial x_k}\delta x_k\Big]\,|_{k=k_0}^{k=k_f} + \sum_{k=k_0}^{k_f-1}\Big(\frac{\partial L(k)}{\partial x_k}\delta x_k + \frac{\partial\mathcal{L}\big(x_{k-1}, x_{k-\tau_{k-1}-1}, x_k, u_{k-1}, u_{k-\omega_{k-1}-1}, \lambda_k\big)}{\partial x_k}\delta x_k + \frac{\partial L(k)}{\partial x_{k-\tau_k}}\delta x_{k-\tau_k} + \frac{\partial L(k)}{\partial u_k}\delta u_k + \frac{\partial L(k)}{\partial u_{k-\omega_k}}\delta u_{k-\omega_k} + \frac{\partial L(k)}{\partial\lambda_{k+1}}\delta\lambda_{k+1}\Big). \quad (25)$$

Therefore, the first variation is obtained as follows:

$$\delta\bar{J}(u(\cdot)) = \Big[\frac{\partial\mathcal{L}\big(x_{k-1}, x_{k-\tau_{k-1}-1}, x_k, u_{k-1}, u_{k-\omega_{k-1}-1}, \lambda_k\big)}{\partial x_k}\delta x_k\Big]\,|_{k=k_0}^{k=k_f} + \sum_{k=k_0}^{k_f-1}\Big[\Big(\frac{\partial F}{\partial x_k} - \lambda_k\Big)\delta x_k + \frac{\partial F}{\partial u_k}\delta u_k + \delta\lambda_{k+1}\big(G(x_k, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k) - x_{k+1}\big) + \lambda_{k+1}^T\Big(\frac{\partial G}{\partial x_k}\delta x_k + \frac{\partial G}{\partial u_k}\delta u_k + \frac{\partial G}{\partial x_{k-\tau_k}}\delta x_{k-\tau_k} + \frac{\partial G}{\partial u_{k-\omega_k}}\delta u_{k-\omega_k}\Big)\Big]. \quad (26)$$

Let

$$\psi_k = \frac{\partial G}{\partial x_{k-\tau_k}}, \qquad \eta_k = \frac{\partial G}{\partial u_{k-\omega_k}}. \quad (27)$$

Since x_k is specified function for $k \le k_0$, and $\tau_k : \mathbb{N} \longrightarrow \mathbb{N}$, then

$$\delta x_{k_i-\tau_{k_i}} = 0, \qquad \text{for all } k_i \in [k_0, k_f - 1] \textbf{ and } k_i - \tau_{k_i} \le 0; \quad (28)$$

otherwise,

$$\lambda_{k_i+1}^T\psi_{k_i} = 0, \qquad k_i - \tau_{k_i} > 0. \quad (29)$$

Similar to equations (28) and (29), we have

$$\delta u_{k_i-\omega_{k_i}} = 0, \qquad \text{for all } k_i \in [k_0, k_f - 1] \textbf{ and } k_i - \omega_{k_i} \le 0; \quad (30)$$

otherwise,

$$\lambda_{k_i+1}^T\eta_{k_i} = 0, \qquad k_i - \omega_{k_i} > 0. \quad (31)$$

Equation (26) can be rewritten as follows:

$$\delta\bar{J}(u(\cdot)) = \Big[\frac{\partial\mathcal{L}\big(x_{k-1}, x_{k-\tau_{k-1}-1}, x_k, u_{k-1}, u_{k-\omega_{k-1}-1}, \lambda_k\big)}{\partial x_k}\delta x_k\Big]\,|_{k=k_0}^{k=k_f} + \sum_{k=k_0}^{k_f-1}\Big[\Big(\frac{\partial F}{\partial x_k} - \lambda_k + \lambda_{k+1}^T\frac{\partial G}{\partial x_k}\Big)\delta x_k + \Big(\frac{\partial F}{\partial u_k} + \lambda_{k+1}^T\frac{\partial G}{\partial u_k}\Big)\delta u_k + \delta\lambda_{k+1}\Big(G(x_k, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k) - x_{k+1}\Big)$$

$$+\lambda_{k+1}^T\psi_k\delta x_{k-\tau_k}+\lambda_{k+1}^T\eta_k\delta u_{k-\omega_k}\big]. \tag{32}$$

In (32), the coefficients $\delta\lambda_k$, δx_k, and δu_k must be zero in order to gain the minimization of $\bar{J}(u(\cdot))$ and $J(u(\cdot))$. Also, Euler–Lagrange equations are derived from (29) and (31) as follows:

$$x_{k+1}=G\Big(x_k,x_{k-\tau_k},u_k,u_{k-\omega_k},k\Big),\qquad k_0\le k\le k_f, \tag{33}$$

$$\begin{cases}\dfrac{\partial F}{\partial x_k}-\lambda_k+\lambda_{k+1}^T\dfrac{\partial G}{\partial x_k}+\lambda_{k+1}^T\psi_k=0,\ k-\tau_k>0,\\ \dfrac{\partial F}{\partial x_k}-\lambda_k+\lambda_{k+1}^T\dfrac{\partial G}{\partial x_k}=0, \qquad\qquad O.W.,\end{cases} \tag{34}$$

$$\begin{cases}\dfrac{\partial F}{\partial u_k}+\lambda_{k+1}^T\dfrac{\partial G}{\partial u_k}+\lambda_{k+1}^T\eta_k=0,\ k-\omega_k>0,\\ \dfrac{\partial F}{\partial u_k}+\lambda_{k+1}^T\dfrac{\partial G}{\partial u_k}=0, \qquad\qquad O.W.,\end{cases} \tag{35}$$

with the following conditions:

$$x_k=\phi_k,\qquad k_0-\tau_{k_0}\le k\le k_0, \tag{36}$$

$$u_k=\Theta_k,\qquad k_0-\omega_{k_0}\le k\le k_0, \tag{37}$$

$$\frac{\partial\mathcal{L}\big(x_{k-1},x_{k-\tau_{k-1}-1},x_k,u_{k-1},u_{k-\omega_{k-1}-1},\lambda_k\big)}{\partial x_k}\ |_{k=k_f}=0. \tag{38}$$

□

3 Numerical examples

Some of the proposed features, including the efficiency and applicability of the technique, are discussed in this section with numerical examples. Our first example uses a non-autonomous DOCP with a time-varying state variable to implement the suggested method. In the second example, we also present the results of solving an autonomous DOCP with constant delays for state and control variables by the introduced method, indicating that we can solve optimal control problems with delays efficiently by this method.

Example 1. Consider the following cost functional:

$$J=\sum_{k=0}^{14}\big(x_k^2+u_k^2\big), \tag{39}$$

subject to non-autonomous recursive equation with time-varying delays

$$x_{k+1} = A_k x_k + A_{1_k} x_{k-\tau_k} + B_k u_k, \qquad 0 \le k \le 14, \tag{40}$$

and the following condition

$$x_k = 1, \qquad k \le 0, \tag{41}$$

where τ_k is the delay function satisfying $\tau_k > 0$ for $0 \le k \le 14$, and $A_k = k$, $A_{1_k} = 1$, and $B_k = 1$. The approach presented in this article has been applied to solve the DOCP with time-varying delays (39)–(41). The numerical results of this example are shown when $\tau_k = 3 - k^2$. The Lagrange function L is defined as follows:

$$\begin{aligned} L(k) =& L\big(x_k, x_{k+1}, x_{k-\tau_k}, u_k, k\big) \\ =& x_k^2 + u_k^2 + \lambda_{k+1}(x_k + k x_{k-3+k^2} + u_k - x_{k+1}). \end{aligned} \tag{42}$$

Therefore, the necessary conditions for the problem (39)–(41) are obtained as follows:

$$\begin{cases} x_{k+1} = x_k + k x_{k-\tau_k} + u_k, & 0 \le k \le 14, \\ \begin{cases} 2x_k - \lambda_k + \lambda_{k+1} k + \lambda_{k+1} = 0, & k - 3 + k^2 \le 0, \\ 2x_k - \lambda_k + \lambda_{k+1} k = 0, & O.W., \end{cases} \\ 2u_k + \lambda_{k+1} = 0, \quad 0 \le k \le 14. \end{cases} \tag{43}$$

Additionally, the following conditions contribute to obtain the solution:

$$x_k = 1, \qquad k \le 0, \tag{44}$$

$$\lambda_{15} = 0. \tag{45}$$

The numerical results of state and control variables of Example 1 are shown in Figure 1 when $\tau_k = 3 - k^2$. Also, we show the convergence curve of the performance index function to illustrate the performance of the proposed method, in Figure 2.

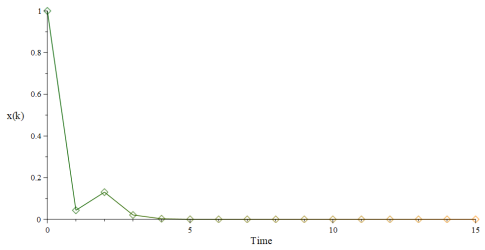
Example 2. Consider the following linear multi-delays time invariant problem to minimize the following functional:

$$J(u) = \frac{1}{2} \sum_{k=0}^{100} (x_k^2 + \frac{1}{2} u_k^2), \tag{46}$$

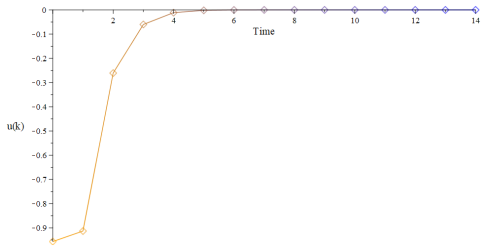
subject to

$$x_k = -x_k + x_{k-\tau} + u_k - \frac{1}{2} u_{k-\omega}, \qquad 0 \le k \le 100, \tag{47}$$

and the following condition



(a) State variable x_k



(b) Control variable u_k

Figure 1: Approximation of state and control variable of Example 1.

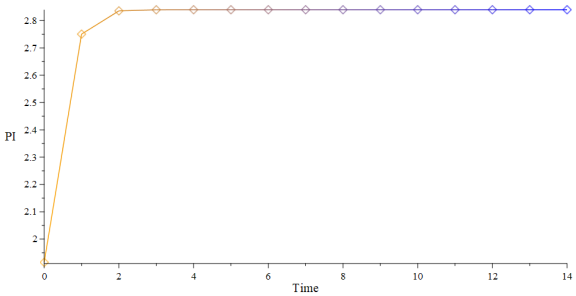


Figure 2: The convergence of performance index function of Example 1.

$$x_k = 1, \qquad k \leq 0, \tag{48}$$
$$u_k = 0, \qquad k \leq 0. \tag{49}$$

Note that in this example,

$$\tau = 6, \qquad \omega = 8.$$

The Lagrange function is defined as follows:

$$\begin{aligned} L(k) =& L\big(x_k, x_{k+1}, x_{k-\tau_k}, u_k, u_{k-\omega_k}, k\big) \\ =& \frac{1}{2}x_k^2 + \frac{1}{4}u_k^2 + \lambda_{k+1}(-x_k + x_{k-6} + u_k - \frac{1}{2}u_{k-8} - x_{k+1}). \end{aligned} \tag{50}$$

The following equations give the optimal solution:

$$\begin{cases} x_{k+1} = -x_k + x_{k-6} + u_k - \dfrac{1}{2}u_{k-8} \qquad 0 \leq k \leq 100, \\ \begin{cases} x_k - \lambda_k - \lambda_{k+1} + \lambda_{k+1} = 0, & k-6 \leq 0, \\ x_k - \lambda_k - \lambda_{k+1} = 0, & O.W., \end{cases} \\ \begin{cases} \frac{1}{2}u_k + \lambda_{k+1} - \frac{1}{2}\lambda_{k+1} = 0, & k-8 \leq 0, \\ \frac{1}{2}u_k + \lambda_{k+1} = 0, & O.W., \end{cases} \end{cases} \tag{51}$$

with the boundary conditions:

$$x_k = 1, \qquad k \leq 0, \tag{52}$$
$$u_k = 0, \qquad k \leq 0, \tag{53}$$
$$\lambda_{100} = 0. \tag{54}$$

The analytic solution to this problem is not available. In Figure 3, the state and control variables of problem (46)–(48) are depicted. To demonstrate the performance of the proposed method, we show the convergence curve of the performance index function in Figure 4.

Example 3. Consider the following two-Dimensional nonlinear time-delays autonomous problem to minimize the following functional:

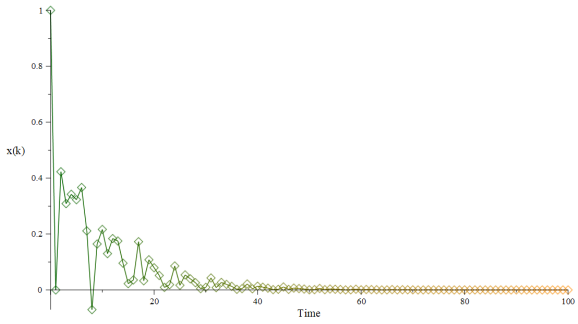
$$J(u_1(\cdot), u_2(\cdot)) = \sum_{k=0}^{k_f-1} (x_1^2(k) + x_2^2(k) + u_1^2(k) + u_2^2(k)), \tag{55}$$

subject to

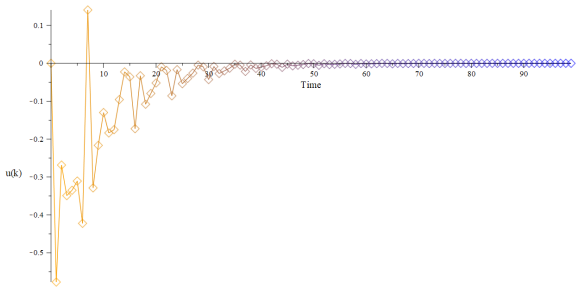
$$x_1(k+1) = x_2^2(k-2) - 0.2u_1(k), \qquad 0 \leq k \leq k_f - 1, \tag{56}$$
$$x_2(k+1) = x_1^2(k-2) - 0.2u_2(k), \qquad 0 \leq k \leq k_f - 1, \tag{57}$$

and the following conditions

$$x_1(k) = 1, \qquad -2 \leq k \leq 0, \tag{58}$$



(a) State variable x_k



(b) Control variable u_k

Figure 3: Approximation of state and control variable of Example 2

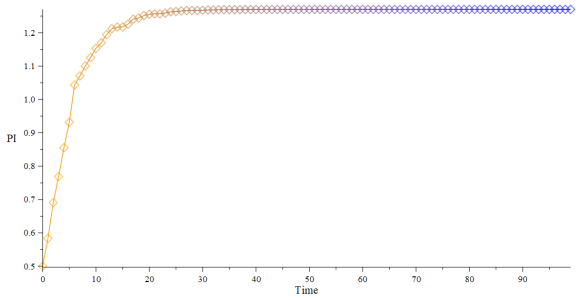


Figure 4: Convergence of performance index function of Example 2.

$$x_2(k) = -1, \qquad -2 \le k \le 0. \tag{59}$$

The following equations give the optimal solution:

$$\begin{aligned} L(k) =& L\big(x_k, x_{k+1}, x_{k-\tau_k}, u_k, k\big) \\ =& x_1^2(k) + x_2^2(k) + u_1^2(k) + u_2^2(k) + \lambda_1(k+1)(x_2^2(k-2) - 0.2u_1(k)) \\ &+ \lambda_2(k+1)(x_1^2(k-2) - 0.2u_2(k)) - \lambda_1(k+1)x_1(k+1) \\ &- \lambda_2(k+1)x_2(k+1), \end{aligned} \tag{60}$$

From equation (34)–(38), we get

$$\begin{cases} \dfrac{\partial F}{\partial x_1(k)} - \lambda_1(k) + \lambda_1^T(k+1)\dfrac{\partial G}{\partial x_1(k)} = 0,\ k-2 \le 0, \\ \dfrac{\partial F}{\partial x_2(k)} - \lambda_2(k) + \lambda_2^T(k+1)\dfrac{\partial G}{\partial x_2(k)} = 0,\ 0 < k-2, \end{cases} \tag{61}$$

$$\begin{cases} 2x_1(k) - \lambda_1(k) + \lambda_1(k+1)(2x_2(k-2)) = 0,\ k-2 \le 0, \\ 2x_1(k) - \lambda_1(k) = 0, \qquad\qquad\qquad\qquad\qquad 0 < k-2, \end{cases} \tag{62}$$

$$\begin{cases} 2x_2(k) - \lambda_2(k) + \lambda_2(k+1)(2x_1(k-2)) = 0,\ k-2 \le 0, \\ 2x_2(k) - \lambda_2(k) = 0, \qquad\qquad\qquad\qquad\qquad 0 < k-2, \end{cases} \tag{63}$$

$$\begin{cases} 2u_1(k) - 0.2\lambda_1(k+1) = 0,\ k-2 \le 0, \\ 2u_2(k) - 0.2\lambda_2(k+1) = 0,\ 0 < k-2, \end{cases} \tag{64}$$

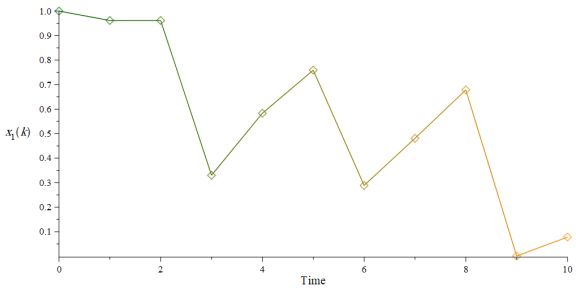
$$\begin{cases} x_1(k+1) = x_2^2(k-2) - 0.2u_1(k), \\ x_2(k+1) = x_1^2(k-2) - 0.2u_1(k), \end{cases} \tag{65}$$

with the boundary conditions:

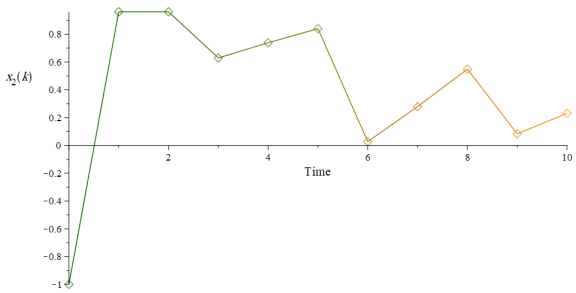
$$x_1(k) = 1, \qquad -2 \le k \le 0, \tag{66}$$

$$x_2(k) = -1, \qquad -2 \le k \le 0. \tag{67}$$

The analytic solution to this problem is not available. In Figure 5, the state variable of the problem (55)–(58) is depicted. Similar to the previous examples, we show the convergence curve of the performance index function in Figure 7. Also, the control variable is illustrated in Figure 6.

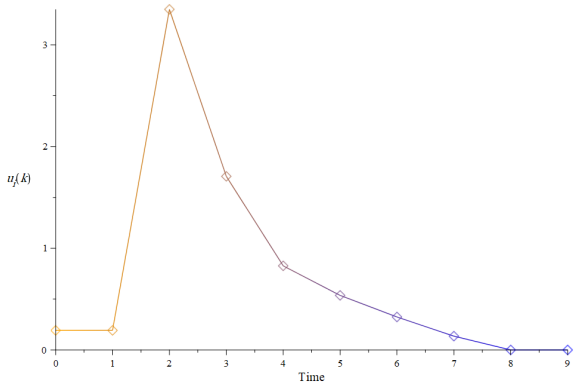


(a) State variable $x_1(k)$

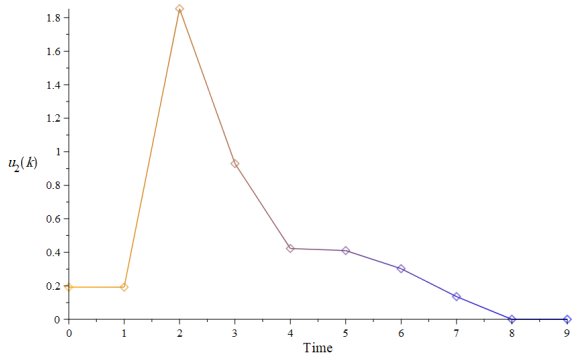


(b) State variable $x_2(k)$

Figure 5: Approximation of state variable of Example 3



(a) Control variable $u_1(k)$



(b) Control variable $u_2(k)$

Figure 6: Approximation of control variable of Example 3

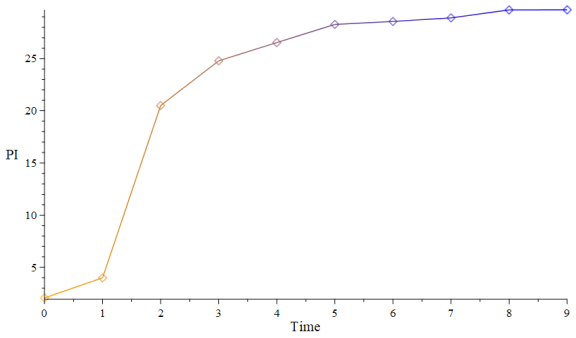


Figure 7: Convergence of performance index function of Example 3.

4 Conclusion

By introducing a new Lagrange multiplier, the original DOCP with time-varying delays problem has been transformed into DOCP problems without time-delay terms to avoid solving the DOCP problem with time-delay terms. In this regard, we utilized the discrete method to derive the new Euler–Lagrange delay formula with a two-point boundary to solve DOCP with time-varying delays. It is important to give a way to solve DOCP with time-varying delays, according to its application. In this technique, we utilized the variation method to construct the Euler–Lagrange formula with a two-point boundary in order to solve DOCP with time-varying delays, which has not been done before. Moreover, two illustrations were supplied to demonstrate how the technique could be used. The performance index influenced the DOCP problem of discrete time-delay systems, and also an approximate regulator was proposed. The simulation results showed that it is simple to implement and robust.

Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Al-Tamimi, A., Lewis, F.L., and Abu-Khalaf, M. *Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof.* IEEE Trans. Syst. Man Cybern. Part B (Cybernetics), 38(4), (2008) 943–949.

2. Essounaini, A., Labzai, A., Laarabi, H., and Rachik, M. *Mathematical modeling and optimal control strategy for a discrete time model of COVID-19 variants.* Commun. Math. Biol. Neurosci. (2022) Article-ID 2022.

3. Ferreira, R.A. *Discrete Calculus. In Discrete Fractional Calculus and Fractional Difference Equations* (pp. 1–14). Springer, Cham, 2022.

4. Fisher, M.E., and Jennings, L.S. *Discrete-time optimal control problems with general constraints.* ACM Trans. Math. Softw. (TOMS), 18(4), (1992) 401–413.

5. Haddad, W.M., Lee, J., and Bhat, S.P. *Asymptotic and finite time semistability for nonlinear discrete-time systems with application to network consensus.* IEEE Trans. Automat. Contr. (2022).

6. Hale, J. K., and Lunel, S.V. *Stability and control of feedback systems with time delays.* Int. J. Syst. Sci. 34(8-9), (2003) 497–504.

7. Hasegawa, Y. *Control problems of discrete-time dynamical systems* (Vol. 447) Springer, 2013.

8. Kot, M. *A first course in the calculus of variations* (Vol. 72). American Mathematical Society, 2014.

9. Lewis, F. L., Vrabie, D., and Syrmos, V.L. *Optimal control.* John Wiley and Sons, 2012.

10. Li, B., Teo, K. L., and Duan, G.R. *Optimal control computation for discrete time time-delayed optimal control problem with all-time-step inequality constraints.* Int. J. Innov. Comput. Inf. Control. 6(3), (2010) 521–532.

11. Li, X., Wang, R., Du, S., and Li, T. *An improved exponential stability analysis method for discrete-time systems with a time-varying delay.* Int. J. Robust Nonlinear Control. 32(2), (2022) 669–681.

12. Lu, J., Wei, Q., Wang, Z., Zhou, T., and Wang, F.Y. *Event-triggered optimal control for discrete-time multi-player non-zero-sum games using parallel control.* Inf. Sci. 584, (2022) 519–535.

13. Mariconda, C., and Tonolo, A. *Discrete calculus. Methods for counting* Springer, 2016.

14. Mehraeen, S., Dierks, T., Jagannathan, S., and Crow, M.L. *Zero-sum two-player game theoretic formulation of affine nonlinear discrete-time systems using neural networks.* IEEE Trans. Cybern. 43(6), (2012) 1641–1655.

15. Miller, F.P., Vandome, A.F., and McBrewster, J. *Gâteaux Derivative: Directional Derivative, Differential Calculus, World War I, Locally Convex Topological Vector Space, Topological Vector Space, Banach Space, Fréchet Derivative, Functional Derivative,* Alphascript Publishing, 2010.

16. Naz, R. *A current-value Hamiltonian approach to discrete-time optimal control problems in economic growth theory.* J. Differ. Equ. Appl. (2022) 1–11.

17. Park, J. H., Lee, T.H., Liu, Y., and Chen, J. *Dynamic systems with time delays: Stability and control.* Singapore: Springer, 2019.

18. Rovelli, C., and Zatloukal, V. *Natural discrete differential calculus in physics.* Found. Phys. 49(7), (2019) 693–699.

19. Tang, G., Sun, H., and Pang, H. *Approximately optimal tracking control for discrete time-delay systems with disturbances.* Prog. Nat. Sci. 18(2), (2008) 225–231.

20. Wei, Q., Liu, D., and Lin, H. *Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems.* IEEE Trans. Cybern. 46(3), (2015) 840–853.

21. Xu, J., Wang, J., Rao, J., Zhong, Y., and Wang, H. *Adaptive dynamic programming for optimal control of discrete-time nonlinear system with state constraints based on control barrier function.* Int. J. Robust Nonlinear Control. 32(6), (2022) 3408–3424.

22. Zanma, T., Yamamoto, N., Koiwa, K. and Liu, K.Z. *Optimal control input for discrete-time networked control systems with data dropout.* IET Cyber-Phys. Syst.: Theory Appl. 2022.

23. Zhao, H., Chen, D., and Hou, L. *The optimal control of delay discrete-time linear system with control constraint.* IEEE ICCA 2010 (pp. 1699–1704). IEEE.

How to cite this article
S.M. Abdolkhaleghzade, S. Effati and S.A. Rakhshan An efficient design for solving discrete optimal control problem with time-varying multi-delay. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(3 (Special Issue), 2022): 719-738. doi: 10.22067/ijnao.2022.78220.1168.

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

All submissions to IJNAO are first evaluated by the journal's Editor-in-Chief or one of the journal's Associate Editors for their appropriateness to the scope and objectives of IJNAO. If deemed appropriate, the paper is sent out for review using a single blind process. Manuscripts are reviewed simultaneously by reviewers who are experts in their respective fields. The first review of every manuscript is performed by at least two anonymous referees. Upon the receipt of the referee's reports, the paper is accepted, rejected, or sent back to the author(s) for revision. Revised papers are assigned to an Associate Editor who makes an evaluation of the acceptability of the revision. Based upon the Associate Editor's evaluation, the paper is accepted, rejected, or returned to the author(s) for another revision. The second revision is then evaluated by the Editor-in-Chief, possibly in consultation with the Associate Editor who handled the original paper and the first revision, for a usually final resolution.

The authors can track their submissions and the process of peer review via: http://ijnao.um.ac.ir

All manuscripts submitted to IJNAO are tracked by using "iThenticate" for possible plagiarism before acceptance.

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] 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.

[2] Stoer, J. and Bulirsch, R. *Introduction to Numerical Analysis*, Springer-verlag, New York, 2002.

Contents