Ferdowsi University of Mashhad

Iranian Journal of Numerical Analysis and Optimization

Volume 12 , Number 1

Winter 2022

Serial Number: 21

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 1, Winter 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
- zbMath
- EBSCO
- ISC
- SID
- Civilica
- Magiran
- DOAJ
- OAJI
- CiteFactor
- AcademicKeys
- ROAD
- Mendeley
- Academia.edu
- Linkedin
- GoogleScholar
- ULRICHS WEB
- INFOBASE

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

Iranian Journal of Numerical Analysis and Optimization

Volume 12, Number 1, Winter 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.

Ali R. Soheili

Contents

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 1-36
DOI:10.22067/ijnao.2021.69185.1020
https://ijnao.um.ac.ir/

Modified ADMM algorithm for solving proximal bound formulation of multi-delay optimal control problem with bounded control

K.A. Dawodu

Abstract

This study presents an algorithm for solving optimal control problems with the objective function of the Lagrange-type and multiple delays on both the state and control variables of the constraints, with bounds on the control variable. The full discretization of the objective functional and the multiple delay constraints is carried out by using the Simpson numerical scheme. The discrete recurrence relations generated from the discretization of both the objective functional and constraints are used to develop the matrix operators, which satisfy the basic spectral properties. The primal-dual residuals of the algorithm are derived in order to ascertain the rate of convergence of the algorithm, which performs faster when relaxed with an accelerator variant in the sense of Nesterov. The direct numerical approach for handling the multi-delay control problem is observed to obtain an accurate result at a faster rate of convergence when over-relaxed with an accelerator variant. This research problem is limited to linear constraints and objective functional of the Lagrange-type and can address real-life models with multiple delays as applicable to quadratic optimization of intensity modulated radiation theory planning. The novelty of this research paper lies in the method of discretization and its adaptation to handle linearly and proximal bound-constrained program formulated from the multiple delay optimal control problems.

AMS subject classifications (2020): 49M29, 93C43.

Keywords: Multi-delay; ADMM; Primal-dual; Proximal operator.

Received 2 March 2021; revised 4 June 2021; accepted 12 July 2021
Kazeem Adebowale Dawodu
Department of Mathematical Sciences, The Federal University of technology Akure, Ondo-State, Nigeria. e-mail: kadawodu@futa.edu.ng

1 Introduction

Optimal control problems (OCPs) with delays in state and/or control variables play important roles in the modeling of real-life scenarios. Time delay effects in economical, physical, chemical, and biological processes involving optimal control systems cannot be neglected, especially when they involve the transmission of information between different parts of the system. These hereditary effects of constant delays have been deliberated upon by Xue and Duanin [27], Rihan and Anwar [21], Laarabi, Abta, and Hattaf, [14] and Bashier and Patidar [1]. Gollman, Kern and Maurer [9] did extensive work on mixed control-state inequality constraints with single delay on both state and delay variables with the initial and terminal boundary conditions in a general mixed form. Later, Gollmann and Maurer [10] extended their work to multiple time delays in the control and state variables with mixed control-state constraints. Olotu and Dawodu [18] and Olotu, Dawodu, and Yusuf [19] worked on one-dimensional and generalized control problems, respectively, with multiple delay constants and unbounded control variable (vector) using modified alternating direction method of multipliers (ADMM), while their earlier work in [17] on the delay proportional control system was done by the method of Quasi-Newton embedded augmented Lagrangian functional.

The Douglas–Rachford (D-R) splitting method called the ADMM (*Alternating Direction Method of Multipliers*) had been deployed by a plethora of authors in solving OCPs; though very few have deployed it to solving multi-delay OCPs with control bounds. The patronage is based on the fact that ADMM enjoys the strong convergence properties of the method of multipliers and the decomposability property of dual ascent as in the works of Boley [2], Boyd and Vandenberghe [4], Sun, Toh, and Yang [23], and Yao et. al [28]; hence its relevance in solving decentralized and block-convex optimization problems like the large-scaled power or distributed systems cannot be overemphasized. Yao et al. [28] discovered that the ADMM could be used as a tool in solving a vehicle routing problem structured as an integer programming problem provided; the quadratic penalty terms used in ADMM can be reduced into simple linear functions. He and Yuan [13] and He, Hong, and Yuan [12] worked on the proximal Jacobian decomposition of the augmented Lagrangian method for the multiple-block separable convex minimization problem. It was observed that it has a strong affinity for ADMM.

Many authors have researched constrained convex quadratic problems with simple bounds. Voglis and Lagaris [24] was among the early authors that developed an algorithm for quadratic programming problems with box constrained using the *active set strategy* with applications mainly to circus tent, random and bi-harmonic problems. However, the scalability, simplicity, and potential of the ADMM in solving large-scaled structured convex optimization problems made it widely acceptable. The ADMM has the ability to solve the primal-dual feasibility updates in parallel and is well amenable to

the Gauss–Seidel accelerator or relaxation in order to improve its efficiency and rate of convergence as in Nesterov [16].

Pasquale and Gerardo [20] also reviewed the main results on global optimality conditions for establishing the global minimization of a quadratic problem with box constraints. However, Carreira-Perpiñán [5] worked on the proximal bound-constrained quadratic problem subject to simple box constraints. The ADMM algorithm was considered using the convex proximal operator by Moreau [15], Rockafellar [22], and Combettes and Pesquet in [6]. The algorithm was found to be particularly efficient in solving a collection of proximal operators that share a same quadratic form with a high rate of convergence and accuracy. Wu and Shang [25] further presented a global optimization method for solving general nonlinear programming problems subject to box constraints using a differential flow on the dual feasible space. The criteria for the global optimality and existence of complete solutions to the original problems were also given.

Xinmin et al. [26] researched the application of the ADMM algorithm to the constrained quadratic optimization of intensity modulated radiation theory planning. A graph form convex optimization model was formulated as $\min h(y)$ such that $y = Ax$, $0 \leq x \leq u_x$, $l_y \leq x \leq u_y$, where the meaning and conditions for the bounds were well-defined. This approach involves the construction of a *proximal operator* and the formulation of the optimization problem in the graph form of the ADMM algorithm. In the implementation of the algorithm, a clinically acceptable solution was obtained with fewer memory requirements when compared with other optimization solvers.

2 Statement of problem

The generalized form of the multiple delay OCP with bounded (box) control is expressed below with quadratic continuous functional F and linear dynamical function f.

2.1 Optimal control model

Considering the multi-delay OCP with bounded control below:

$$\min J(x,u) = \frac{1}{2}\int_{t_0}^{T} F(t,x(t),u(t))\,dt \tag{1}$$

$$\text{s.t. } \dot{x}(t) \leq Ax + Bu + \sum_{j=1}^{d} \alpha_j x(t-r_j) + \sum_{l=1}^{e} \beta_l u(t-q_l) \tag{2}$$

$$\text{s.t. } \dot{x}(t) \leq f(t, x(t), u(t), x^h(t - r_j), u^h(t - q_l)), \quad j(l) = 1, 2, \ldots, d(e), \tag{3}$$
$$x(t_0) = x_0, \tag{4}$$
$$x(t) = \phi(t), \qquad t_0 - r \leq t \leq t_0, \tag{5}$$
$$u(t) = \psi(t), \qquad t_0 - q \leq t \leq t_0, \tag{6}$$
$$\gamma \leq u(t) \leq \sigma, \tag{7}$$

where

$$\begin{aligned}
x &= (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n, \quad \phi = (\phi_1, \phi_2, \ldots, \phi_n)^T \in \mathbb{R}^n, \\
u &= (u_1, u_2, \ldots, u_m)^T \in \mathbb{R}^m, \\
x^h(t - r_j) &= (x(t - r_1), x(t - r_2), \ldots, x(t - r_d)) \in \mathbb{R}^{dn}, \\
u^h(t - q_l) &= (u(t - q_1), u(t - q_2), \ldots, u(t - q_e)) \in \mathbb{R}^{em}, \\
\psi &= (\psi_1, \psi_2, \ldots, \psi_m)^T \in \mathbb{R}^m, \\
\gamma &= (\gamma_1, \ldots, \gamma_m)^T \in \mathbb{R}^m, \\
\sigma &= (\sigma_1, \ldots, \sigma_m)^T \in \mathbb{R}^m, \\
r &= \max\{r_j\}_{j=1}^d, \text{ and } q = \max\{q_l\}_{l=1}^e.
\end{aligned}$$

2.2 Assumptions and properties of the model

The underlining assumptions and properties of the functions are stated below:

(i) The *time interval* $I = [t_0, T] \subseteq \mathbb{R}$ is a subset of real numbers $\mathbb{R}$;

(ii) The *state function* $x(t) : I \to X$ of the optimal control system is absolutely continuous on I provided the subset X is closed and bounded in $\mathbb{R}^n$;

(iii) The *control function* $u(t) : I \to U$ of the control system is piece-wise continuous and Lebesgue measurable on I provided the subset U is closed and bounded in $\mathbb{R}^m$;

(iv) The numbers $r_1, r_2, \ldots, r_d$ and $q_1, q_2, \ldots, q_e$ are the known *monotonically increasing positive delay constants* for the state and control variables, respectively; that is, $r_j \leq r_{j+1}$ for all $r_j \in \mathbb{R}$ for $j = 1, 2, \ldots, d$ and $q_l < q_{l+1}$ for all $q_l \in \mathbb{R}$ for $l = 1, 2, \ldots, e$;

(v) The *initial (pre-shaped) functions* $\varphi(t) : [t_0 - r, t_0] \to \mathbb{R}^n$ and $\psi(t) : [t_0 - q, t_0] \to \mathbb{R}^m$ for the state and control variables, respectively, are known and piecewise continuous;

(vi) The numbers $r = \max\{r_j\}_{j=1}^d$ and $q = \max\{q_l\}_{l=1}^e$ represent the values of the *maximum delay constants* for the state and control variables, respectively;

(vii) The *cost functional* $F : [t_0, T] \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ is nonlinear, piecewise continuous and measurable;

(viii) The *constraint functional* $f : [t_0, T] \times \mathbb{R}^{(1+d)n} \times \mathbb{R}^{(1+e)m} \to \mathbb{R}^n$ is linear and piece-wise continuous;

(ix) There exist an *admissible pair* $p = (x(\cdot), u(\cdot)) \in \Omega := ([t_0, T], \mathbb{R}^n) \times ([t_0, T], \mathbb{R}^m)$ that satisfies (1)–(7) of the multi-delay OCP, where the functional $J(x(\cdot), u(\cdot))$ is minimized by the the *admissible triple* $\Omega := (x(\cdot), u(\cdot), p(\cdot))$.

3 Methodology

A very simple discretization of (1) and (3) is obtained by means of the Simpsons and Euler numerical schemes, respectively. Recurrence relations generated are used in developing the respective large sparse matrix operators. The augmented Lagrangian functional with relaxation parameter (factor) is then used as an accelerator variant in the formulation of the modified ADMM (M-ADMM) algorithm. The spectral and convergence analyses are carried out to ensure the well-posedness of the algorithm.

3.1 Background and preliminaries

Suppose that the discrete time interval is given as $I_k = [t_k, t_{k+1}]$ by letting $t_k = t_0 + k\delta$ for $k = 0, 1, \ldots, N$. Then the discretization operator, say f_x, maps each discrete point t_k in the discrete time interval $I_k \subseteq \mathbb{R}$ into each discrete (grid) point of the concatenated state vector $x_i^{(k)} \in \mathbb{R}^{nN}$ for all $i = 1, 2, \ldots, n$, while the operator, say f_u, maps the points into each discrete point of the concatenated control vector $u_j^{(k)} \in \mathbb{R}^{m(N+1)}$ for all $j = 1, 2, \ldots, m$. It is then expressed as

$$f_x : [t_0, T] \subseteq \mathbb{R} \longrightarrow x_i^{(k)} \in \bar{x}_i, \tag{8}$$

$$f_u : [t_0, T] \subseteq \mathbb{R} \longrightarrow u_j^{(k)} \in \bar{u}_j, \tag{9}$$

where $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n) \in \mathbb{R}^{nN}$ and $\bar{u} = (\bar{u}_1, \ldots, \bar{u}_m) \in \mathbb{R}^{m(N+1)}$ for $\bar{x}_i = (x_i^{(1)}, \ldots, x_i^{(N)}) \in \mathbb{R}^N$, and $\bar{u}_j = (u_j^{(0)}, \ldots, u_j^{(N)}) \in \mathbb{R}^{N+1}$, respectively. Considering the existence of multiple delay constants $r_j \in \mathbb{R}$ (for $j = 1, 2, \ldots, d$) and $q_l \in \mathbb{R}$ (for $l = 1, 2, \ldots, e$) on the state and control variables, respectively. Then some basic Theorems are expressed below to aid in the process of discretization.

Theorem 1 (Rationality theorem)**.** Given the real numbers r_j, $r_{j+1} > 0$ for $r_j < r_{j+1}$, then there exists a unique real number $\delta < 1$ such that the ratios of the numbers is a rational number $\mathbb{Q}$.

Proof. Let the delay constants be arranged in an increasing manner; that is, $r_j \leq r_{j+1}$ and $q_l \leq q_{l+1}$ for every $j,\ l$. It then implies that for every r_j and q_l, there exist integers $m_j \in \mathbb{Z}^+$ and $q_l \in \mathbb{Z}^+$ such that

$$\frac{r_j}{m_j} = \frac{\delta q_l}{\delta w_l} \in \mathbb{R}, \tag{10}$$

where δ is known as the step-length or interval length. This clearly shows that the delays, for both the state and control variables, are integer multiples of the step-length. □

The proof of this transformation technique was suggested by Guinn [11] to derive first-order necessary conditions for unconstrained OCPs with pure state delays. To further elaborate on the assumptions above and the applications, we then introduce Theorem 2 below.

Theorem 2. Given any interval $[a, b]$, there exists a step-length $h \in \mathbb{R}^+$ such that each sub-interval is a constant multiple of the step-length.

Proof. Let the entire length of the interval $l = b - a$ be partitioned into m sub-intervals $[r_{j-1}, r_j]$ for $j = 1, 2, \ldots, m$. Then there exists $h \in \mathbb{R}$ with $r_j = hv_j$ for all $v_j \in \mathbb{Z}^+$ such that the length of each sub-interval can be written as $l_j = r_j - r_{j-1}$.

$$\frac{l_j}{h} = \frac{(r_j - r_{j-1})}{h} = n_j + \delta_j \qquad\qquad n_j \in \mathbb{Z}^+,\ \delta_j \in [0, 1).$$

Then,

$$\frac{(hv_j - hv_{j-1})}{h} = (v_j - v_{j-1}) = n_j + \delta_j \qquad\qquad n_j \in \mathbb{Z}^+,\ \delta_j \in [0, 1). \tag{11}$$

Since $v_j - v_{j-1} \in \mathbb{Z}^+$ then $n_j + \delta_j \in \mathbb{Z}^+$ such that $\delta_j = 0$ or $\delta_j = 1$ for all $j = 1, 2, \ldots, d$ (by Theorem 1). Since $\delta_j = 1 \notin [0, 1)$, then $\delta_j = 0$ for all $j = 1, 2, \ldots, d$. By slotting $\delta_j = 0$ into (11), we then conclude that the length of each sub-interval $l_j = (r_j - r_{j-1})$ can be expressed as a multiple integer $v_j \in \mathbb{Z}^+$ of h expressed as $r_j = r_{j-1} + hn_j$, where $v_j - v_{j-1} = n_j$ for all $j = 1, 2, \ldots, m$. □

The essence of Theorems 1 and 2 above is to establish the fact that there exists a real number, referred to as *step-length*, $h \in \mathbb{R}$ that divides the entire state and control delay constants with multiples of real positive integers.

Consequent to 1 and 2, the multiple state and control variables are presented as follows:

$$x^{(k-v_j)} = \begin{cases} \phi(t_{k-v_j}), & k - v_j \leq 0;\ k = 0, 1, 2, \ldots, v_j,\ t \in [t_0 - r, t_0],\ \text{(known)} \\ x^{(k-v_j)}, & k - v_j > 0;\ k = (v_j + 1), \ldots, N,\ t \in [t_0, T],\ \text{(unknown)} \\ & x(t_0) = x_0 \text{ given} \end{cases} \tag{12}$$

and

$$u^{(k-w_l)} = \begin{cases} \psi(t_{k-w_l}), & k - w_l < 0;\ k = 0, 1, 2, \ldots, (w_l - 1),\ t \in [t_0 - q, t_0],\ \text{(known)} \\ u^{(k-w_l)}, & k - w_l \geq 0 \text{ for } k = w_l, (w_l + 1), \ldots, N,\ t \in [t_0, T],\ \text{(unknown)} \end{cases} \tag{13}$$

where $r = r_d = \max\{r_j\}_{j=1}^{d}$ and $q = q_e = \max\{q_l\}_{l=1}^{e}$ are the largest delays on the state and control variables, respectively. In other words, the discretized delay state and control variables are represented in the form below:

$$x^{(k-v_j)} = \left(x_1^{(k-v_j)}, x_2^{(k-v_j)}, \ldots, x_n^{(k-v_j)}\right)^T \in \mathbb{R}^n, \quad \text{for any } j, \tag{14}$$

$$u^{(k-w_l)} = \left(u_1^{(k-w_l)}, u_2^{(k-w_l)}, \ldots, u_n^{(k-w_l)}\right)^T \in \mathbb{R}^m, \quad \text{for any } l, \tag{15}$$

while $x(t)$ and $u(t)$ are estimated within their respective delay intervals by the given delay functions $\phi(t)$ and $\psi(t)$, respectively, such that $\phi(t) \simeq \phi(t_{-k})$ and $\psi(t) \simeq \psi(t_{-k})$, where $t_k = t_0 - k\delta$ for $k = 1, 2, \ldots, v_d, \ldots, w_e$ if $v_d \leq w_e$. However, the largest state and control delay vectors for each iterate $k = 0, 1, \ldots$ are represented below:

$$\hat{x} = \left(x^{(1-v_j)}, x^{(2-v_j)}, \ldots, x^{(N-v_j)}\right)^T \in \mathbb{R}^{n(N-v_j)}, \quad \text{for all } j = 1, 2, \ldots, d, \tag{16}$$

$$\hat{u} = \left(u^{(-w_l)}, u^{(1-w_l)}, \ldots, u^{(N-w_l)}\right)^T \in \mathbb{R}^{m(N+1)}, \quad \text{for all } l = 1, 2, \ldots, e. \tag{17}$$

3.2 Discretization of the objective functional

The objective function in (1) is of the form

$$\min J(x, u) = \frac{1}{2} \int_{t_0}^{T} (x^T P x + u^T Q u)\, dt, \tag{18}$$

where $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$, $u = (u_1, u_2, \ldots, u_m)^T \in \mathbb{R}^m, P \in \mathbb{R}^{n \times n}$, and $Q \in \mathbb{R}^{m \times m}$. The full discretization of the objective function (1) using the Simpson integration formula gives

$$\min J(x, u) \simeq \frac{1}{2} \sum_{k=1}^{N-1} F(t_k, x^{(k)}, u^{(k)}) \tag{19}$$

and when expanded, it can be expressed as the quadratic function

$$\min_{\bar{x},\bar{u}} \frac{1}{2}\,\bar{x}^T\bar{P}\bar{x} + \frac{1}{2}\,\bar{u}^T\bar{Q}\bar{u} + R, \tag{20}$$

where $R = \frac{\delta}{6}(x^{(0)})^T P x^{(0)} \in \mathbb{R}$ and the concatenated state and control variables are $\bar{x} = (x_1^{(1)}, x_1^{(2)}, \ldots, x_n^{(N)})^T \in \mathbb{R}^{nN}$ and $\bar{u} = (u_1^{(0)}, u_1^{(1)}, \ldots, u_m^{(N)})^T \in \mathbb{R}^{m(N+1)}$, respectively. However, the block-diagonal coefficient matrices $\bar{P} \in \mathbb{R}^{nN\times nN}$ and $\bar{Q} \in \mathbb{R}^{m(N+1)\times m(N+1)}$ are the block-matrix operators of the objective functional stated below as

$$\bar{P} = \begin{pmatrix} \frac{4\delta}{3}P & 0 & \cdots & \cdots & 0 \\ 0 & \frac{2\delta}{3}P & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{4\delta}{3}P & 0 \\ 0 & \cdots & \cdots & 0 & \frac{\delta}{3}P \end{pmatrix} \quad \text{and} \quad \bar{Q} = \begin{pmatrix} \frac{\delta}{3}Q & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \frac{4\delta}{3}Q & \ddots & \cdots & \cdots & \vdots \\ \vdots & \ddots & \frac{2\delta}{3}Q & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \frac{4\delta}{3}Q & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{\delta}{3}Q \end{pmatrix}$$

The detailed derivation is in Appendix as in Olotu, Dawodu, and Yusuf [19]. However, in determining the properties of the derived matrix operators above, the following definitions are then introduced.

Definition 1 (Sylvester criterion)**.** A square $(n\times n)$ matrix is positive definite if it is real, symmetric with all the eigenvalues $(\lambda_i > 0;\ i = 1, 2, \ldots, n)$ or the leading principal minors $(M_i > 0;\ i = 1, 2, \ldots, n)$ are strictly positive.

Definition 2 (Corollary to Sylvester criterion)**.** If the principal diagonal entries a_{ii} are the only nonzero entries of a square $(n \times n)$ matrix such that $a_{ij} = 0$ for $i \neq j$, then the leading principal minors $(M_i > 0;\ i = 1, 2, \ldots, n)$ are given by $M_i = \prod_{j=1}^{i} a_{jj}$ for $i = 1, 2, \ldots, n$.

Definition 3. Every positive definite matrix is invertible.

Sequel to Definitions 1 and 2, the constructed matrix-operator P is symmetric and positive definite since all the leading principal minors are strictly positive. Consequently, by Definition 3, the matrix P is invertible.

3.3 Discretization of the constraints

The Euler numerical scheme in (21) below is used for the discretization of the multi-delay constraints;

$$x^{(k+1)} = x^{(k)} + \delta f^k, \tag{21}$$

where $f^k = f(t_k, x^{(k)}, u^{(k)}, \hat{x}^{(k)}, \hat{u}^{(k)})$ and expressed in terms of the RHS of (3) as

$$f^k = Ax^{(k)} + Bu^{(k)} + \sum_{j=1}^{d} \alpha_j x^{(k-v_j)} + \sum_{l=1}^{e} \beta_l u^{(k-w_l)}, \quad k = 0, 1, 2, \ldots, N-1, \tag{22}$$

where $\hat{x}^{(k)} = \{x^{(k-v_1)}, x^{(k-v_2)}, \ldots, x^{(k-v_d)}\}$ and $\hat{u}^{(k)} = \{u^{(k-w_1)}, u^{(k-w_2)}, \ldots, u^{(k-w_e)}\}$. Adapting (21) and (22) to (3) of the OCP yields the recurrence relation below:

$$\theta x^{(k)} - x^{(k+1)} + \omega u^{(k)} \leq -\delta \sum_{j=1}^{d} \alpha_j x^{(k-v_j)} - \delta \sum_{l=1}^{e} \beta_l u^{(k-w_l)}, \tag{23}$$

where $\theta = (I_n + A\delta) \in \mathbb{R}^{n\times n}$, $\omega = B\delta \in \mathbb{R}^{n\times m}$, $I_n \in \mathbb{R}^{n\times n}$ (identity) and $k = 0, 1, \ldots, N-1$. Slotting the various values of k into (23) forms the linear inequality below:

$$\bar{A}\bar{x} + \bar{B}\bar{u} \leq \bar{E}\bar{x}^h + \bar{F}\bar{u}^h = C, \tag{24}$$

where $\bar{A}$ is a multi-diagonal matrix with nN rows and nN columns (i.e., $\bar{A} \in \mathbb{R}^{nN\times nN}$); $\bar{B}$ is also a diagonal matrix with nN rows and $m(N+1)$ columns (i.e., $\bar{B} \in \mathbb{R}^{nN\times m(N+1)}$), and $\bar{x}$ is the concatenated vector of the state variable with dimension $nN \times 1$ while $\bar{u}$ is the concatenated row vector of the control variable with dimension $m(N+1) \times 1$. The dimensions of the matrices $\bar{E}$, $\bar{x}^h$, $\bar{F}$ and $\bar{u}^h$ are $nN \times n(v_d+1)$, $n(v_d+1) \times 1$, $nN \times mw_e$ and $mw_e \times 1$, respectively. The respective structures of the concatenated vector-matrices are represented below:

$$\bar{A}\bar{x} = \begin{pmatrix} -I_n & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \theta & -I_n & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \theta & -I_n & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \bar{\alpha_1} & \cdots & 0 & \theta & -I_n & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \bar{\alpha_d} & \cdots & \bar{\alpha_1} & \cdots & 0 & \theta & -I_n & \ddots & \ddots & \vdots \\ 0 & \bar{\alpha_d} & \cdots & \bar{\alpha_1} & \cdots & 0 & \theta & -I_n & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \bar{\alpha_d} & \cdots & \bar{\alpha_1} & \cdots & 0 & \theta & -I_n \end{pmatrix} \begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(v_1+1)} \\ \vdots \\ x^{(v_2-v_1)} \\ \vdots \\ x^{(v_d-v_{d-1})} \\ \vdots \\ x^{(v_d+1)} \\ \vdots \\ x^{(N-1)} \\ x^{(N)} \end{pmatrix},$$

$$\bar{B}\bar{u} = \begin{pmatrix} \omega & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \omega & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \bar{\beta}_1 & \cdots & 0 & \omega & 0 & \cdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \bar{\beta}_e & \cdots & \bar{\beta}_1 & \cdots & 0 & \omega & 0 & \ddots & \vdots \\ 0 & \bar{\beta}_e & \cdots & \bar{\beta}_1 & \cdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \bar{\beta}_e & \cdots & \bar{\beta}_1 & \cdots & 0 & \omega & 0 \end{pmatrix} \begin{pmatrix} u^{(0)} \\ \vdots \\ x^{(v_1+1)} \\ \vdots \\ u^{(w_2-w_1)} \\ \vdots \\ \vdots \\ u^{(w_e+1)} \\ \vdots \\ u^{(N)} \end{pmatrix},$$

$$\bar{E}\bar{x}^h = \begin{pmatrix} -\theta & \cdots & -\bar{\alpha}_1 & \cdots & -\bar{\alpha}_s & \cdots & -\bar{\alpha}_{d-1} & -\bar{\alpha}_d \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & 0 \\ -\bar{\alpha}_1 & \cdots & -\bar{\alpha}_s & \cdots & -\bar{\alpha}_{d-1} & -\bar{\alpha}_d & 0 & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & \vdots \\ -\bar{\alpha}_s & \cdots & -\bar{\alpha}_{d-1} & -\bar{\alpha}_d & \cdots & 0 & \cdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & \vdots \\ -\bar{\alpha}_{d-1} & -\bar{\alpha}_d & \cdots & 0 & \cdots & \cdots & \cdots & \vdots \\ -\bar{\alpha}_d & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & 0 \end{pmatrix} \begin{pmatrix} x^{(0)} \\ x^{(-1)} \\ \vdots \\ x^{(v_2-v_3)} \\ \vdots \\ \vdots \\ \vdots \\ x^{(v_d-v_{d-1})} \\ \vdots \\ x^{(1-v_d)} \\ x^{(-v_d)} \end{pmatrix}$$

and

$$\bar{F}\bar{u}^h = \begin{pmatrix} -\bar{\beta}_1 & \cdots & -\bar{\beta}_2 & \cdots & -\bar{\beta}_s & \cdots & -\bar{\beta}_{e-1} & -\bar{\beta}_e \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & 0 \\ -\bar{\beta}_1 & \cdots & -\bar{\beta}_s & \cdots & -\bar{\beta}_{e-1} & -\bar{\beta}_e & 0 & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & \vdots \\ -\bar{\beta}_s & \cdots & -\bar{\beta}_{e-1} & -\bar{\beta}_e & \cdots & 0 & \cdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & \vdots \\ -\bar{\beta}_{e-1} & -\bar{\beta}_e & \cdots & 0 & \cdots & \cdots & \cdots & \vdots \\ -\bar{\beta}_e & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\ddots & 0 \end{pmatrix} \begin{pmatrix} x^{(0)} \\ u^{(-1)} \\ \vdots \\ u^{(w_2-w_3)} \\ \vdots \\ \vdots \\ \vdots \\ u^{(w_e-w_{e-1})} \\ \vdots \\ u^{(1-w_e)} \\ u^{(-w_e)} \end{pmatrix},$$

where the delay coefficients $\bar{\alpha_k}$ and $\bar{\beta_k}$ are defined below as

$$[\bar{\alpha}_k] = \begin{cases} -\delta\alpha_k, & 1 \leq k \leq d, \\ 0 & \text{elsewhere} \end{cases} \tag{25}$$

and

$$[\bar{\beta}_k] = \begin{cases} -\delta\beta_k, & 1 \leq k \leq e, \\ 0 & \text{elsewhere.} \end{cases} \tag{26}$$

However, the entries $[\bar{a}_{ij}] \in \bar{A}$, $[\bar{b}_{ij}] \in \bar{B}$, $[\bar{e}_{ij}] \in \bar{E}$, and $[\bar{f}_{ij}] \in \bar{F}$ of the various matrix structures are described below as follows:

$$[\bar{a}_{ij}] = \begin{cases} -I_n, \, i = j \quad 1 \leq i \leq N, \\ \theta, \quad j = i - 1 \quad 2 \leq i \leq N, \\ \bar{\alpha}_k, \, j = i - 1 - v_k \quad 2 + v_k \leq i \leq N, \quad k = 1, 2, \ldots, d, \\ 0 \quad \text{elsewhere,} \end{cases} \tag{27}$$

$$[\bar{b}_{ij}] = \begin{cases} \omega, \;\; i = j \quad 1 \leq i \leq N, \\ \bar{\beta}_k, \, j = i - w_k \quad 1 + w_k \leq i \leq N, \quad k = 1, 2, \ldots, e, \\ 0 \quad \text{elsewhere,} \end{cases} \tag{28}$$

$$[\bar{e}_{ij}] = \begin{cases} -\theta, \quad i = j = 1, \\ -\bar{\alpha}_k, \, j = v_k + 1 - i \quad 1 \leq i \leq v_k, \quad k = 1, 2, \ldots, d, \\ 0 \quad \text{elsewhere,} \end{cases} \tag{29}$$

$$[\bar{f}_{ij}] = \begin{cases} -\sigma, \quad i = j = 1, \\ -\bar{\beta}_k, \, j = w_k + 1 - i \quad 1 \leq i \leq w_k, \quad k = 1, 2, \ldots, e, \\ 0 \quad \text{elsewhere.} \end{cases} \tag{30}$$

It is imperative to note that the real symmetric and positive definite properties of the discretized matrices $\bar{P}, \bar{Q}, \bar{A}$, and $\bar{B}$ affect the well-posedness of the algorithm and the selection of the optimal (relaxation and penalty) parameters; hence the bases for the assumption.

3.4 Splitting method

The D-R splitting technique, also known as ADMM, is a special case of the splitting technique and can be traced back to work done on monotone operators and operator splitting methods into the general Hilbert spaces. Eckstein and Ferris [7] showed in turn that DR splitting is also a special case of the

proximal point algorithm applicable to optimal control systems. Suppose there exist two separable closed convex functions (or proximal operators) $f(x)$ and $g(x)$ that are non-expansive and not necessarily symmetric. Then the iterates of the D-R splitting algorithm for $\min f(x) + g(x)$ starting with any ζ^0 and repeating for $k = 0, 1, \ldots$ are stated as

$$\begin{aligned} \zeta^{k+1} &\longleftarrow \zeta^k + \text{prox}_g(2x^k - \zeta^k) - x^k, \\ x^{k+1} &\longleftarrow \text{prox}_f(\zeta^{k+1}), \end{aligned}$$

where x^k converges to the solution of $0 \in \partial f(x) + \partial g(x)$, if the solution exists. The D-R Algorithm can be re-structured by introducing a new variable $w^k = x^k - \zeta^k$ if when substituted into the D-R splitting Algorithm above, it yields its equivalent iterates expressed below as

$$\begin{aligned} u^{k+1} &\longleftarrow \text{prox}_g(x^k + w^k), \\ x^{k+1} &\longleftarrow \text{prox}_f(u^{k+1} - w^k), \\ w^{k+1} &\longleftarrow w^k + x^{k+1} - u^{k+1}, \end{aligned}$$

where $x = \text{prox}_f(u - w)$ satisfies $0 \in \partial f(x) + \partial g(x)$. However, the equivalent D-R Algorithm can be amended by introducing the Nesterov-type accelerated variants or relaxation factor $\alpha \in [0, 2]$ in order to improve the rate of convergence of the iterates. The relaxed iterates is arrived at by replacing u^{k+1} with $\alpha u^{k+1} + (1 - \alpha)x^k$ in the sense of Nesterov [16]. The derived algorithm is stated as

$$\begin{aligned} &\text{Initialize at any } \zeta^0,\ x^0,\ \text{fixed } \alpha \text{ and repeat for } k = 0, 1, \ldots, \\ &u^{k+1} \longleftarrow \text{prox}_g(x^k + w^k), \\ &x^{k+1} \longleftarrow \text{prox}_f(\alpha u^{k+1} + (1 - \alpha)x^k - w^k), \\ &w^{k+1} \longleftarrow w^k + x^{k+1} - (1 - \alpha)x^k - \alpha u^{k+1}, \end{aligned}$$

where it is an over-relaxation for $1 < \alpha < 2$, under-relaxation for $0 < \alpha < 1$ and reduces to its non-relaxed form if $\alpha = 1$. The extension of the splitting method in the computations of the updates of the dual and adjoint variables of a separable convex constrained problem of the form $\min f(x_1) + g(x_2)$ s.t. $h(x_1, x_2) = 0$ requires a sequential minimization of the augmented Lagrangian functional $L_\rho(x_1, x_2, \zeta)$; hence the name alternating direction method of multipliers (ADMM).
The three steps iterates (updates) for the ADMM algorithm are as follows:

$$x^{k+1} \longleftarrow \arg\min_{x_1} \left(f(x_1) + \zeta^T h(x_1, x_2^k) + \frac{\rho}{2} \parallel h(x_1, x_2^k) \parallel_2^2 \right), \tag{31}$$

$$x_2^{k+1} \longleftarrow \arg\min_{x_2} \left(g(x_2) + \zeta^T h(x_1^{k+1}, x_2) + \frac{\rho}{2} \parallel h(x_1^{k+1}, x_2) \parallel_2^2 \right), \tag{32}$$

$$\zeta^{k+1} \longleftarrow \zeta^k + \rho h(x_1^{k+1}, x_2^{k+1}), \tag{33}$$

where $\rho > 0$ is the penalty parameter and $x_1^{k+1} \equiv \alpha x_1^{k+1} - (1-\alpha)x_2^k$ for any accelerator variant $\alpha \in [0, 2]$ puts the algorithm in a relaxed mode. The ADMM is a more general method for handling all kinds of convex optimization problems which includes the ℓ_1- regularization of the for $\min \sum_i f_i(w) + ||w||$, which other traditional methods such as gradient descent, Newton, an so on cannot handle. The ease with which the ADMM can be implemented in parallel for large-scaled data at a faster rate gives it an edge over other algorithms.

3.5 Analysis on bounded control problems

In a general box constrained nonlinear programming problem

$$\min \{f(x) \,|\, x \in \Omega \subseteq \mathbb{R}^n\}, \tag{34}$$

where $\Omega = \{x \in \mathbb{R}^n \,|\, l \leq x \leq u\}$ is a feasible space and a closed convex subset of $\mathbb{R}^n$, $f(x)$ and a strongly convex quadratic function that is twice continuously differentiable in $\mathbb{R}^n$, l and u are the lower and upper bounds, respectively. Carreira-Perpiñán [5] and Combettes and Pesquet [6] extended the theory of proximal operator to a proximal bound-constrained quadratic program (QP) (34) with the Lagrange formulation presented as

$$L_\rho(x, z, \xi) = f(x) + g(z) + \frac{\rho}{2} \parallel x - z + \xi \parallel_2^2 \quad \text{s.t.} \quad x = z, \tag{35}$$

where $z \in \mathbb{R}^n$ is the slack (auxiliary) variable, $\xi = \frac{\lambda}{\rho} \in \mathbb{R}^n$ is the scaled dual (adjoint) vector, $\rho > 0$ is the penalty parameter, and $g(z) = \frac{\mu}{2} \parallel z - v \parallel_2^2$ is the convex term (or indicator function for the nonnegative orthant).
The proximal M-ADMM formulation is then expressed in the form

$$x^{k+1} \longleftarrow \mathbf{prox}_{f,\rho}(z^k - \xi^k) = \arg\min_x \left(f(x) + \frac{\rho}{2} \parallel x - z^k + \xi^k \parallel_2^2 \right), \tag{36}$$

$$z^{k+1} \longleftarrow \mathbf{prox}_{g,\rho}(x^{k+1} - \xi^k) = \arg\min_z \left(\frac{\mu}{2} \parallel z - v \parallel_2^2) + \frac{\rho}{2} \parallel x^{k+1} - z + \xi^k \parallel_2^2 \right)$$

$$\text{s.t.} \quad l \leq z \leq u, \tag{37}$$

$$\xi^{k+1} \longleftarrow \xi^k + x^{k+1} - z^{k+1}. \tag{38}$$

However, there exist three possible cases of the location of the parabola vertex of the objective functional upon which the optimal solution is determined. See sketches below for illustration. Each of the cases indicated in Figures 1, 2, and 3 represents the locations of the parabola vertex z_i^* within the bounds, below the lower bound, and above the upper bound, respectively, while Figure 4 represents the median of the three points. Therefore, the optimal solution defined on the feasible space is the median of each of the ith component $(i = 1, 2, \ldots, N)$ of the three vectors; lower bound l, upper bound

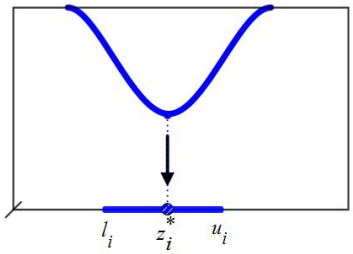


Figure 1: Vertex within bounds: $l_i \leq z_i \leq u_i$

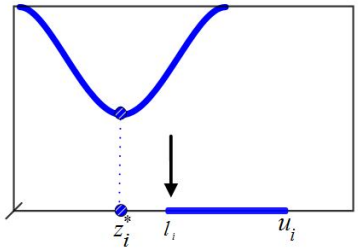


Figure 2: Vertex below lower bound: $z_i \leq l_i$

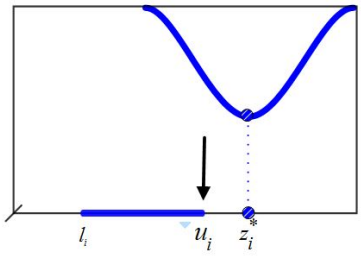


Figure 3: Vertex above upper bound: $u_i \leq z_i$

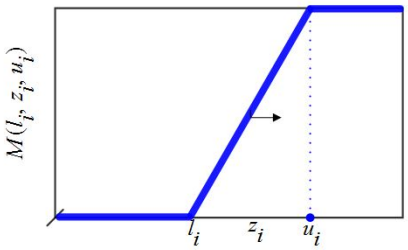


Figure 4: Median points: $M(l_i, z_i,\ u_i)$

u, and the parabola vertex z_i^*. The choice of the next iterate $\bar{z}$ is determined as follows:

$$\bar{z}_i = \begin{cases} M(z_i^*, l_i, u_i) = \text{Min}\big[\text{Max}(z_i^*, l_i), u_i\big] = \text{Min}\big[l_i, u_i\big] = l_i, & z_i^* \leq l_i, \\ M(\bar{l}_i, z_i^*, u_i) = \text{Min}\big[\text{Max}(l_i, z_i^*), u_i\big] = \text{Min}\big[z_i^*, u_i\big] = z_i^*, & \bar{l}_i \leq z_i^* \leq u_i, \\ M(l_i, u_i, z_i^*) = \text{Min}\big[\text{Max}(l_i, u_i), z_i^*\big] = \text{Min}\big[u_i, z_i^*\big] = \bar{u}_i, & z_i^* \geq \bar{u}_i. \end{cases} \tag{39}$$

3.6 M-ADMM algorithm formulation

The original ADMM formulation is modified to accommodate both the state and control variables before imposing the Karush–Khun–Tucker (KKT) optimality conditions. The Gauss–seidel accelerator variant is introduced to improve the convergence properties of the algorithm as illustrated in the convergence analysis below. The re-formulated compact convex quadratic optimization problem is then stated as

$$\text{Min} J(\bar{x},\bar{u}) = \frac{1}{2}\bar{x}^T P\bar{x} + \frac{1}{2}\bar{u}^T Q\bar{u} + R \tag{40}$$

$$\text{s.t. } G(x,u) = \bar{A}\bar{x} + \bar{B}\bar{u} - C \leq 0, \tag{41}$$

$$\bar{\gamma} \leq \bar{u} \leq \bar{\sigma}, \tag{42}$$

$$\bar{\sigma} - \hat{u} \geq 0, \tag{43}$$

$$\bar{-}\gamma + \hat{u} \geq 0, \tag{44}$$

where

$$\bar{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(N)}) \in \mathbb{R}^{nN}, \quad \bar{u} = (u^{(0)}, u^{(1)}, \ldots, u^{(N)}) \in \mathbb{R}^{m(N+1)},$$
$$\bar{A} \in \mathbb{R}^{nN\times nN}, \quad \bar{C} \in \mathbb{R}^{nN}, \quad \bar{B} \in \mathbb{R}^{nN\times m(N+1)},$$
$$\bar{P} \in \mathbb{R}^{nN\times nN}, \text{ and } \bar{Q} \in \mathbb{R}^{m(N+1)\times m(N+1)},$$

while $\bar{\gamma} = (1/\sqrt{N}+1(\gamma_1^{(1)}, \gamma_1^{(2)}, \ldots, \gamma_1^{(N+1)}, \gamma_2^{(1)}, \ldots, \gamma_m^{N+1}) \in \mathbb{R}^{m(N+1)}$ and $\bar{\sigma} = (1/\sqrt{N}+1)(\sigma_1^{(1)}, \sigma_1^{(2)}, \ldots, \sigma_1^{(N+1)}, \sigma_2^{(1)}, \ldots, \sigma_m^{N+1}) \in \mathbb{R}^{m(N+1)}$ for all $\gamma_i \in \gamma$ and $\sigma_i \in \sigma$, respectively. In addition, $\bar{P}$ and $\bar{Q}$ are the real, symmetric, and positive-definite matrix operators.

3.7 Proximal-bound formulation

In formulating the *convex-proximal bound* constrained program for the discretized multi-delay OCP in (40)–(42), the Carreira-Perpiñá approach [5] is applied. Suppose that the control bound (42) is re-formulated into two equations, $\bar{\sigma} - \bar{u} \geq 0$ and $\bar{-}\gamma + \hat{u} \geq 0$ such that

$$\begin{aligned} 0 &\leq \frac{\mu}{2} \parallel \underbrace{(\sigma - \gamma)}_{y} - \underbrace{(v_1 + v_2)}_{v} \parallel_2^2 \\ &\leq \frac{\mu}{2} \parallel \bar{\sigma} - \bar{u} - v_1 \parallel_2^2 + \frac{\mu}{2} \parallel \bar{-}\gamma + \bar{u} - \xi_2 \parallel_2^2 + \frac{\mu}{2} \parallel y - v \parallel_2^2, \end{aligned} \tag{45}$$

where $g(y) = \frac{\mu}{2} \parallel y - v \parallel_2^2$ is a control bound function defined for $\bar{u} \geq y$ and $\bar{\gamma} \leq y \leq \bar{\sigma}$; while v is the slack variable replacing the inequalities and μ is the penalty parameter. Therefore, the associated augmented Lagrangian functional for the quadratic formulation is expressed as

$$\begin{aligned} L_{\rho_1,\rho_2}(\bar{x},\bar{u},z,v,y,\xi_1,\xi_2,\mu) =& \frac{1}{2}\bar{x}^T P\bar{x} + \frac{1}{2}\bar{u}^T Q\bar{u} + R + \frac{\mu}{2} \parallel y - v \parallel_2^2 \\ &+ \frac{\rho_1}{2} \parallel A\bar{x} + B\bar{u} - C + z + \xi_1 \parallel_2^2 \\ &+ \frac{\rho_2}{2} \parallel \bar{u} - y + \xi_2 \parallel_2^2 \\ &\text{s.t.} \quad z, v \geq 0 \text{ and } \bar{\gamma} \leq y \leq \bar{\sigma}, \end{aligned} \tag{46}$$

with the M-ADMM iterations stated as

$$\bar{x}^{k+1} \longleftarrow \arg\min_{\bar{x}} \left\{ \frac{1}{2}\bar{x}^T P\bar{x} + \frac{\rho_1}{2} \parallel A\bar{x} + B\bar{u}^k - C + z^k + \xi_1^k \parallel_2^2 \right\},$$
$$\bar{u}^{k+1} \longleftarrow \arg\min_{\bar{u}} \left\{ \frac{1}{2}\bar{u}^T Q\bar{u} + \frac{\rho_1}{2} \parallel A\bar{x}^{k+1} + B\bar{u} - C + z^k \parallel_2^2 + \frac{\rho_2}{2} \parallel \bar{u} - y^k + \xi_2^k \parallel_2^2 \right\},$$
$$z^{k+1} \longleftarrow \arg\min_{z} \left\{ \frac{\rho_1}{2} \parallel A\bar{x}^{k+1} + B\bar{u}^{k+1} - C + z + \xi_1^k \parallel_2^2 \right\} \text{ s.t. } z \geq 0,$$
$$v^{k+1} \longleftarrow \arg\min_{v} \left\{ \frac{\mu}{2} \parallel y - v \parallel_2^2 \right\} \text{ s.t. } v \geq 0,$$

$$y^{k+1} \longleftarrow \arg\min_{y} \left\{ \frac{\mu}{2} \parallel y - v \parallel_2^2 + \frac{\rho_2}{2} \parallel \bar{u}^{k+1} - y + \xi_2^k \parallel_2^2 \right\} \text{ s.t. } \quad \bar{\gamma} \leq y \leq \bar{\sigma},$$
$$\xi_1^{k+1} = \xi_1^k + \left(A\bar{x}^{k+1} + B\bar{u}^{k+1} - C + z^{k+1} \right),$$
$$\xi_2^{k+1} = \xi_2^k + \left(\bar{u}^{k+1} - y^{k+1} \right),$$

where $\bar{x}$ and $\bar{u}$ are the primal variables, λ is the Lagrange multiplier, $\rho_1,\ \rho_2 > 0$ are the penalty parameters, $\|\cdot\|_2$ is the euclidean (spectral) norm of a vector (matrix) argument, $\xi_i = \lambda_i/\rho_i,\ i = 1, 2$ are the scaled dual vectors, y is the introduced auxiliary (slack) vector, and $l_+(z)$ is the indicator function for the non-negative orthants defined as $l_+(y) = \frac{\mu}{2} \parallel y - v \parallel_2^2$ and ∞ otherwise.

3.8 Derivations of the update formulas

Applying the KKT optimality conditions on the augmented Lagrangian functional (46) for the sequential minimization of the variables requires updating all the critical variables as indicated in the ADMM iterations. Moreover the update formulas of the adjoint variables will be accelerated using the Gauss–Seidel relaxation factor $\alpha \in [0, 2]$ as clearly illustrated in the works of Nesterov [16] and Ghadimi et al. [8]; though this work considers an over-relaxation where $\alpha \in (1, 2]$.

State-update - $\bar{x}$: Setting $\partial_{\bar{x}} L(\bar{x}, \cdot) = 0$ yields

$$\bar{x}^{k+1} = -\rho_1(P + \rho_1 A^T A)^{-1} A^T (B\bar{u}^k - C + z^k + \xi_1^k) \quad (\bar{x} - \text{update}), \quad (47)$$

where $(P + \rho_1 A^T A)^{-1}$ is invertible since P is real, symmetric, and positive definite and $A^T A$ is positive semi-definite sequel to Definitions 1, 2, and 3.

Control-update - $\bar{u}$: Setting $\partial_{\bar{u}} L(\bar{x}^{k+1}, \bar{u}, \cdot^k) = 0$ and replacing $A\bar{x}^{k+1}$ with $h(\bar{x}^{k+1}, \bar{u}^k, \alpha) = \alpha A\bar{x}^{k+1} - (1-\alpha)(B\bar{u}^k - C + z^k)$ to relax the algorithm with the relaxation factor yield

$$\bar{u}^{k+1} = -\rho(\bar{Q} + \rho\bar{B}^T\bar{B})^{-1}\bar{B}^T \left[\alpha(\bar{A}\bar{x}^{k+1} - C + z^k) - (1-\alpha)\bar{B}\bar{u}^k + v^k \right]. \quad (48)$$

Auxiliary (Slack)-update - $\bar{z}$: Setting $\partial_z L(\cdot^{k+1}, z, \cdot^k) = 0$ and relaxing the algorithm by replacing $A\bar{x}^{k+1}$ with $h(\bar{x}^{k+1}, \bar{u}^k, \alpha)$ yields

$$z^* = -\alpha(A\bar{x}^{k+1} + B\bar{u}^{k+1} - C) - (1-\alpha)[B(\bar{u}^{k+1} - \bar{u}^k) - z^k] - \xi_1^k. \quad (49)$$

Therefore,

$$\begin{aligned} z^{k+1} &= \max\left\{0, z^*\right\} \quad (50)\\ &= \max\left\{0, -\alpha(A\bar{x}^{k+1} + B\bar{u}^{k+1} - C) - (1-\alpha)[B(\bar{u}^{k+1} - \bar{u}^k) - z^k] - \xi_1^k\right\}. \end{aligned}$$

Updating the parabola vertex - y : Setting $\partial_y L(\cdot^{k+1}, y, \cdot^k) = 0$ in the derivation of the parabola vertex y in the box constraint yields

$$\partial_y L(\cdot^{k+1}, y, \cdot^k) = \mu(y - v) - \rho_2(\bar{u} - y + \xi_2) = 0 \quad (51)$$

$$y^* = \frac{\mu v^{k+1} + \rho_2(\bar{u}^{k+1} + \xi_2^k)}{(\mu + \rho_2)} \quad \text{s.t.} \quad \bar{\gamma} \le y^* \le \bar{\sigma}, \quad (52)$$

where for each component y_i^* of $y^* = (y_1, y_2, \ldots, y_N)$, the minimum of the upward parabola is defined in the sub-interval $[\bar{\gamma}_i, \bar{\sigma}_i]$ such that the solution is the median $[\bar{\gamma}_i, y_i^*, \bar{\sigma}_i]$ of $\bar{\gamma}_i, \bar{\sigma}_i$ and each component y_i^* of the parabola vertex derived above is specifically defined as

$$y_i^* = \frac{\mu v_i^{k+1} + \rho_2(\bar{u}_i^{k+1} + \xi_{2i}^k)}{(\mu + \rho_2)} \quad \text{s.t.} \quad \bar{\gamma}_i \le y_i^* \le \bar{\sigma}_i \text{ and element-wise for all } i. \quad (53)$$

Therefore,

$$\begin{aligned} y^{k+1} &= \text{Min}\left[\bar{\gamma}, \frac{\mu v^{k+1} + \rho_2(\bar{u}^{k+1} + \xi_2^k)}{(\rho_2 + \mu)}, \bar{\sigma}\right], \quad (54)\\ &= \text{Min}[\text{Max}(\bar{\gamma}, y^*), \bar{\sigma}], \end{aligned}$$

with the steps involving element-wise thresholding operations. Considering the extension of the above concept of proximal bound operator on the OCP, let the state variable $\bar{x} \in \mathbb{R}^N$ belong to a set of admissible state functions X (written $\bar{x} \in X \subseteq \mathbb{R}^\infty$) and let the control variable $\bar{u} \in \mathbb{R}^{(N+1)}$ belong to the set of admissible control functions U (written $\bar{u} \in U \in \mathbb{R}^\infty$). We then assume that $v^{(\bar{\gamma})} = (\bar{x}^*, \bar{\gamma})$ and $v^{(\bar{\sigma})} = (\bar{x}^*, \bar{\sigma})$ are the points on the state-control coordinates $(\bar{x}, \bar{u})$ for which the control is bounded below and above by $\bar{\gamma} \in \mathbb{R}^{(N+1)}$ and $\bar{\sigma} \in \mathbb{R}^{(N+1)}$, respectively. If there exists any point on the ith component of the control vector $(i = 1, 2, \ldots (N+1))$, then $y_i \in \left[\bar{\gamma}_i, \bar{\sigma}_i\right]$ is such that $v^{(y_i)} = (\bar{x}_i^*, y_i)$ is the point on the $(\bar{x}, \bar{u})$ trajectory that prescribes the parabola vertex of the objective function $f(\bar{x}, \bar{u})$ at its minimum. Therefore, the median M of $v^{(\bar{\gamma}_i)}$, $v^{(\bar{\sigma}_i)}$ and the parabola vertex $v^{(y_i)}$ are defined as follows:

$$(\bar{x}_i^*, y_i^*) = \begin{cases} \text{Min}\big[\text{Max}(v^{(\bar{\gamma}_i)}, v^{(y_i)}), v^{(\bar{\sigma}_i)}\big] = v^{(y_i)}, & v^{(y_i)} \in \big[v^{(\bar{\gamma}_i)}, v^{(\bar{\sigma}_i)}\big], \\ \text{Min}\big[\text{Max}(v^{(\bar{u}_i)}, v^{(\bar{\gamma}_i)}), v^{(\bar{\sigma}_i)}\big] = v^{(\bar{\gamma}_i)}, & v^{(y_i)} \leq v^{(\bar{\gamma}_i)}, \\ \text{Min}\big[\text{Max}(v^{(\bar{\gamma}_i)}, v^{(\bar{\sigma}_i)}), v^{(y_i)}\big] = v^{(\bar{\sigma}_i)}, & v^{(y_i)} \geq v^{(\bar{\sigma}_i)}, \end{cases} \tag{55}$$

where $\bar{x} = \bar{x}_i^*$ and $\bar{u} = y_i^*$ are the optimal state and control variables, respectively, within the feasible space. Therefore, at $\bar{x} = \bar{x}^{(k+1)}$, the $\bar{u}$-update is defined as

$$(\bar{x}^{k+1}, y^{k+1}) = M\left[v^{(\bar{\gamma}_i)}, v^{(\bar{\sigma}_i)}, v^{(y_i)}\right], \text{ element-wise for all } i \in \{1, 2, \ldots, (N+1)\}, \tag{56}$$

where $v^{(y_i)} = (\bar{x}_i^*, y_i^*) \equiv (\bar{x}_i^*, y_i^{k+1})$ such that

$$y_i^{k+1} = \frac{\mu v_i^{k+1} + \rho_2(\bar{u}_i^{k+1} + \xi_{2i}^k)}{(\rho_2 + \mu)}, \quad \bar{x}_i^* = \bar{x}_i^{(k+1)}, \quad u_i = y_i, \text{ for all } i \in \{1, 2, \ldots, (N+1)\}. \tag{57}$$

For further illustration on the analysis of the proximal bound and its effects on the objective functional value, we then consider the figures below.
Suppose that $(\bar{x}^*, y^*)$ is the vertex parabola for which $f(\bar{x}, \bar{u})$ is at minimum. Then

$$F(\bar{x}^*, y^*) \leq F(\bar{x}^*, \bar{u}), \text{ s.t. } \bar{x}^* \in X, \text{ for all } \bar{u} \in U \quad \text{and} \quad y^* \in \big[\bar{\gamma}, \bar{\sigma}\big].$$

Assume that there exists a bound $\big[\bar{\gamma}, \bar{\sigma}\big] \subseteq U$ on the control variable, simply called bounded (box) constraint, for which the objective functional value $F(\bar{x}, \bar{u})$ is at minimum; then in the process of the algorithm, the three expected cases are stated as

$$F(\bar{x}^*, y^*) = \begin{cases} F(\bar{x}^*, y^*), & \text{for all } \bar{u} \in \big[\bar{\gamma}, \bar{\sigma}\big] \subseteq U & \text{Case I} \\ F(\bar{x}^*, \bar{\gamma}), & \text{for all } \bar{u} \leq \bar{\gamma} \subseteq U & \text{Case II} \\ F(\bar{x}^*, \bar{\sigma}), & \text{for all } \bar{u} \geq \bar{\sigma} \subseteq U & \text{Case III} \end{cases} \tag{58}$$

Case I: $\bar{\gamma} \leq \bar{u} \leq \bar{\sigma}$
In Figure 5, the vertex point $(\bar{x}^*, \bar{u}^v)$ that falls within the control bound is same as the optimal state-control trajectories $(\bar{x}^*, \bar{u}^*)$ such that $F(\bar{x}, \bar{u}^*) = F(\bar{x}, \bar{u}^v)$. The functional Ω within the region of feasibility is then defined as

$$\Omega^* = \{F(\bar{x}^*, \bar{u}^v) \,|\, \bar{x}^* \in X \text{ and } \bar{u} \in [\bar{\gamma}, \bar{\sigma}]\}. \tag{59}$$

Case II: $\bar{u} \leq \bar{\gamma}$
In Figure 6, the vertex point $(\bar{x}^*, \bar{u}^v)$ that falls below the lower control bound is same as the optimal state-control trajectories $(\bar{x}^*, \bar{\gamma})$ such that $F(\bar{x}, \bar{u}^*) = F(\bar{x}, \bar{\gamma})$. The functional Ω within the region of feasibility is then defined as

$$\Omega^* = \{F(\bar{x}^*, \bar{\gamma}) \,|\, \bar{x}^* \in X \text{ and } \bar{u} \leq \bar{\gamma} \quad \text{for all } \bar{u} \in U\} \tag{60}$$

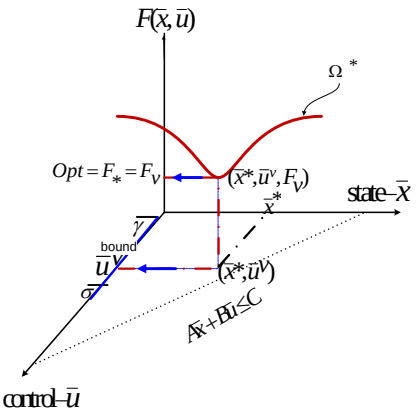


Figure 5: Control within bounds: $\bar{\gamma} \leq \bar{u}^v \leq \bar{\sigma}$

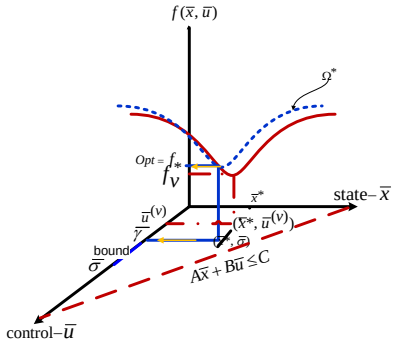


Figure 6: control below lower bound: $\bar{u}^v \leq \bar{\gamma}$

Case III: $\bar{u} \geq \bar{\sigma}$
In Figure 7, the vertex point $(\bar{x}^*, \bar{u}^v)$ falls above the upper control bound with optimal state-control trajectories given as $(\bar{x}^*, \bar{\sigma})$ such that $F(\bar{x}, \bar{u}^*) = F(\bar{x}, \bar{\sigma})$. The functional Ω within the region of feasibility is then defined as

$$\Omega^* = \{F(\bar{x}^*, \bar{\gamma}) \,|\, \bar{x}^* \in X \text{ and } \bar{u} \geq \bar{\sigma} \quad \text{for all } \bar{u} \in U\}. \tag{61}$$

Updating the scaled-dual variables - $\xi_1,\ \xi_2$: It yields

$$\xi_1^{k+1} = \xi_1^k + (A\bar{x}^{k+1} + B\bar{u}^{k+1} - C + z^{k+1}), \tag{62}$$

and

$$\xi_2^{k+1} = \xi_2^k + \bar{u}^{k+1} - y^{k+1}. \tag{63}$$

3.9 Primal-dual convergence

The convergence of the M-ADMM for the bounded control in terms of its primal-dual feasibility requires deriving the primal-dual residuals and their behaviors for large iterations.

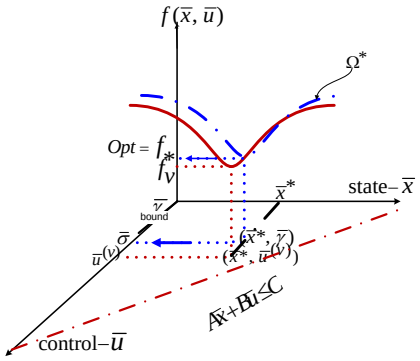


Figure 7: Control above upper bound: $\bar{\sigma} \leq \bar{u}^v$

Theorem 3. Given the bounded control problem (40)–(42) with the multipliers for the linear and bounded control constraints given as λ_1 and λ_2, respectively, then there exists a dual residual $s^{k+1} = \rho_1 B^T\left[z^{k+1} - z^k\right] - \rho_2\left[u^{k+1} - y^k\right]$ that converges to zero for given penalty parameters ρ_1 and ρ_2 for the linear and control bounds, respectively.

Proof. Given the objective function $p^k = f(x^k) + g(u^k)$, linear inequality constraint $Ax + Bu \leq C$ and $y = \sigma - \gamma$, the associated Lagrangian with slack z is stated as

$$\begin{aligned} L(\bar{x},\bar{u},z,\lambda_1,\lambda_2:\rho_1,\rho_2) = f(\bar{x}) + g(\bar{u}) + \lambda_1^T(A\bar{x} + B\bar{u} - C + z) \\ + \frac{\rho_1}{2} \parallel A\bar{x} + B\bar{u} - C + z \parallel_2^2 + \frac{\mu}{2} \parallel y - v \parallel_2^2 \\ + \lambda_2^T(\bar{u} - y) + \frac{\rho_2}{2} \parallel \bar{u} - y \parallel_2^2 \end{aligned}$$

Applying the optimality conditions KKT to the Lagrangian function above yields

$$\partial g(\bar{u}) + B^T[\lambda_1^k + \rho_1(A\bar{x}^{k+1} + B\bar{u} - C + z^k)] + \lambda_2^k + \rho_2(\bar{u} - y^k) = 0,$$
$$\partial g(\bar{u}^{k+1}) + B^T\big[\lambda_1^k + \rho_1 \underbrace{(A\bar{x}^{k+1} + B\bar{u}^{k+1} - C + z^{k+1})}_{r_1^{k+1}}\big] + \lambda_2^k$$
$$+ \rho_2 \underbrace{(\bar{u}^{k+1} - y^{k+1})}_{r_2^{k+1}} - \rho_1 B^T z^{k+1} + \rho_1 B^T z^k + \rho_2 y^{k+1} - \rho_2 y^k = 0,$$
$$\partial g(\bar{u}^{k+1}) + B^T\big[\underbrace{\lambda_1^k + \rho_1 r_1^{k+1}}_{\lambda_1^{k+1}}\big] + \underbrace{\lambda_2^k + \rho_2 r_2^{k+1}}_{\lambda_2^{k+1}}$$
$$= \underbrace{\rho_1 B^T\big[z^{k+1} - z^k\big] - \rho_2\big[y^{k+1} - y^k\big]}_{s^{k+1}},$$

where the **primal residuals** are expressed as

$$r_1^{k+1} = A\bar{x}^{k+1} + B\bar{u}^{k+1} - C + z^{k+1}, \tag{64}$$
$$r_2^{k+1} = \bar{u}^{k+1} - y^{k+1}. \tag{65}$$

If $x^{k+1} = x^*$ minimizes the convex function, then

$$\partial g(\bar{u}^*) + B^T\lambda_1^* + \lambda_2^* = 0 \tag{66}$$

and setting $u^{k+1} = y^{k+1}$ completes the proof by expressing the **dual residual** as

$$s^{k+1} = \rho_1 B^T\big[z^{k+1} - z^k\big] - \rho_2\big[u^{k+1} - y^k\big]. \tag{67}$$

□

Let the state-control variables be concatenated as $\hat{w} = (\bar{x}, \bar{u})$ with dimension $nN + m(N+1) \times 1$, $\hat{A} = \big[\bar{A}, \bar{B}\big]$ with dimension $nN \times (nN + mN + m)$, and $\hat{P} = \left[\begin{array}{c|c} [c|c]\bar{P} & \hat{0} \\ \hline \hat{0}^T & \bar{Q} \end{array}\right]$ with dimension $(nN + mN + m) \times (nN + mN + m)$. This yields a re-formulated augmented problem of the form

$$\text{Min}\frac{1}{2}\hat{w}^T\hat{P}\hat{w} + R \quad \text{s.t.} \quad \hat{A}\hat{w} \leq C, \tag{68}$$

which results into Theorem 4, as an extension of Ghadimi et al. [8].

Theorem 4. Given a convex quadratic programming (QP) problem $\min \frac{1}{2}\hat{w}^T\hat{P}\hat{w} + q^T\hat{w}$ such that $\hat{A}\hat{w} \leq b$, where $\hat{P} \in \mathbb{R}^{n\times n}$, $\hat{w} \in \mathbb{R}^n$, $q \in \mathbb{R}^n$, $\hat{A} \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$, then the optimal step-size for the QP is

$$\bar{\rho}_{CAL} = \left[\sqrt{\lambda_{\min}\left(\hat{A}\hat{P}^{-1}\hat{A}^T\right)\lambda_{\max}\left(\hat{A}\hat{P}^{-1}\hat{A}^T\right)}\right]^{-1} \tag{69}$$

and the convergence factor is

$$\xi_R^* = \frac{\lambda_{\max}\left(\hat{A}\hat{P}^{-1}\hat{A}^T\right) - \sqrt{\lambda_{\min}\left(\hat{A}\hat{P}^{-1}\hat{A}^T\right)\lambda_{\max}\left(\hat{A}\hat{P}^{-1}\hat{A}^T\right)}}{\lambda_{\max}\left(\hat{A}\hat{P}^{-1}\hat{A}^T\right) + \sqrt{\lambda_{\min}\left(\hat{A}\hat{P}^{-1}\hat{A}^T\right)\lambda_{\max}\left(\hat{A}\hat{P}^{-1}\hat{A}^T\right)}}. \tag{70}$$

For $\alpha \in (1, 2], 0 < \xi_R < 1$, $\hat{P}$ is symmetric and positive definite and $\lambda_{\min}(\hat{A}\hat{P}^{-1}\hat{A}^T)$ and $\lambda_{\max}(\hat{A}\hat{P}^{-1}\hat{A}^T)$ are the minimum and maximum eigenvalues of the matrix $\hat{A}\hat{P}^{-1}\hat{A}^T$, respectively.

3.10 The Error dynamics of the relaxed Algorithm

Let us consider the augmented convex OCP in (68) with ℓ_2-regularized estimation, with the regularization parameters $\delta > 0$, as stated below:

$$\text{Min}\frac{1}{2}\hat{w}^T\hat{P}\hat{w} + R + \frac{\delta}{2}||z||_2^2 \quad \text{s.t.} \quad \hat{A}\hat{w} + z - C = 0. \tag{71}$$

The resulting relaxed ADMM iterations take the form

$$\hat{w}^{k+1} = -\left(\hat{P} + \rho\hat{A}^T\hat{A}\right)^{-1}\hat{A}^T\left(\lambda^k + \rho(z^k - C)\right) \tag{72}$$

$$z^{k+1} = \frac{-\left[\lambda^k + \rho\left(\alpha(\hat{A}\hat{w}^{k+1} - C) - (1-\alpha)z^k\right)\right]}{\delta + \rho} \tag{73}$$

$$\lambda^{k+1} = \lambda^k + \rho\left(\alpha(\hat{A}\hat{w}^{k+1} + z^{k+1} - C) + (1-\alpha)(z^{k+1} - z^k)\right). \tag{74}$$

Considering the convergence of the relaxed iterations, (72)–(74), it is imperative to note that the relaxed iterations return the standard iterations of the convex control problem by setting $\alpha = 1$ and $\delta = 0$ with the augmented variable $\hat{w}^{k+1} = [\hat{x}^{k+1}\ \hat{u}^{k+1}]^T$. In characterizing the convergence of the relaxed iterations, we then analyze the *error dynamics* of the ADMM to know how the errors associated with $\hat{w}^k$ or z^k vanish. Making λ^k subject of formula in (73) and inserting into (74) yield $\lambda^{k+1} = -\delta z^{k+1}$. Inserting the w-update into the z-update and using the fact that $\lambda^k = -\delta z^k$, we arrive at

$$z^{k+1} = \underbrace{\frac{1}{\delta+\rho}\left[\delta I + \rho\left((1-\alpha)I + \alpha(\rho-\delta)\hat{A}(\hat{P} + \rho\hat{A}^T\hat{A})^{-1}\hat{A}^T\right)\right]}_{E} z^k$$
$$-\frac{\alpha\rho}{\delta+\rho}\left[\rho\hat{A}(\hat{P} + \rho\hat{A}^T\hat{A})^{-1}\hat{A}^T - I\right]C. \tag{75}$$

For $z = z^*$ a fixed-point of (75), that is,

$$z^* = Ez^* - \frac{\alpha\rho}{\delta+\rho}\left[\rho\hat{A}(\hat{P}+\rho\hat{A}^T\hat{A})^{-1}\hat{A}^T - I\right]C,$$

then the dual error $e^{k+1} = z^{k+1} - z^*$ evolves as $e^{k+1} = Ee^k$ and $z^{k+1} - z^* = E(z^k - z^*)$ such that the $z-$ update of the relaxed iterations above converge if and only if the spectral radius of the error matrix E in the above linear iterations is less than one. The eigenvalues of E can then be written as

$$\xi_{\hat{R}}(\alpha,\rho,\lambda_i(\hat{P})) = 1 - \frac{\alpha\rho(\lambda_i(\hat{P})+\delta)}{(\rho+\lambda_i(\hat{P}))(\rho+\delta)}, \tag{76}$$

for $\rho,\ \delta$ and $\lambda_i(\hat{P}) \in \mathbb{R}_+$ (i.e., positive) provided $\alpha \in \left(1,\ \frac{2(\rho+\lambda_i(\hat{P}))(\rho+\delta)}{\rho(\lambda_i(\hat{P})+\delta)}\right)$. To ascertain the upper bounds ρ and α that guarantee an improvement with strictly smaller convergence factor compared to that of the classical ADMM iterations, we take the derivatives $\xi^*_{\hat{R}}$ with respect to the parameters ρ and α given by (ρ^*,α^*) $=\underset{\rho,\alpha}{\operatorname{argmin}}\max_i|\xi_{\hat{R}}(\alpha,\rho,\lambda_i(\hat{P}))|$ while that for the lower bound of $\xi_{\hat{R}}$ is $\xi^*_{\hat{R}} = \max_i|\xi_{\hat{R}}(\rho^*,\alpha^*,\lambda_i(\hat{P}))|$. This yields the jointly optimal step-size and convergence factor expressed as $(\rho^*,\xi^*_{\hat{R}})$, where the detail proof of the convergence analysis can be seen in [8]. We then conclude that the over-relaxed iterations are guaranteed to converge faster for all $\alpha \in (1,2]$. The developed M-ADMM algorithm is stated below; see Appendix for flowchart.

3.11 Algorithm: M-ADMM for proximal bound-constrained program

Input parameters and operators $\rho,\ \alpha,\ A,\ B,\ C,\ \epsilon^{\text{prim}},\ \epsilon^{\text{dual}}$
Initialize $\bar{x}^0,\ \bar{u}^0,\ z^0,\ \xi_1^0 = \lambda_1^0/\rho_1,\ \xi_2^0 = \lambda_1^0/\rho_2$
Set $P \succeq 0,\ Q \succeq 0$ (symmetric and positive definite)
For $k = 0,1,2,\ldots,$ **do**
| Compute $\bar{x}^{k+1}$ using (47)
| Compute $\bar{u}^{k+1}$ using (48)
| Compute $\bar{z}^{k+1}$ using (50)
| Compute $\bar{y}^{k+1}$ using (54) and set $\bar{u}_1^{k+1} = y_i^{k+1}$, element-wise for all i
| Update ξ_1^{k+1} and ξ_2^{k+1} using (62) and (63), respectively.
| Compute $||r^{k+1}||_2^2$ and $||s^{k+1}||_2^2$ using (64) and (67), respectively.
| **If** $||r^{k+1}||_2^2 \le \epsilon^{\text{prim}}$ and $||s^{k+1}||_2^2 \le \epsilon^{\text{dual}}$
| | **stop** and return (output)
| **else**
| | **repeat** process
| **end (If)**

end (for)
Return $\bar{x}^{k+1},\ \bar{u}^{k+1},\ z^{k+1},\ J^*,\ \xi_1^{k+1},\ \xi_2^{k+1}$

3.12 Computational Complexity

Iteration complexity: Each step of the M-ADMM updates (iterations) is $O(N)$ with large computational efforts required. However, catching the Cholesky factorization of the matrices $(P + p_1 A^T A)$ and $(Q + \rho_1 B^T B + \rho_2 I)$ in the form RR^T (with R the upper triangular) will make the computation more effective, provided P and Q are real symmetric and positive definite. If P or Q is dense, then the computation of the Cholesky factor is a one-time cost of $O\left(\frac{1}{3}N^3\right)$ while the linear system is $O(N^2)$ by solving the two triangular systems. However, if P or Q is large and (sufficiently) sparse, then computing the Cholesky factor and solving the linear system are both possibly $O(N)$. The solution can be obtained in few iterations of the M-ADMM algorithm when the CGM or L-BFGS iterative solver is deployed with a warm-start to carry out faster convergence of the updates. In practice, the warm-start in which the values of $z = v$ and $\xi_1 = \xi_2 = 0$ are usually adopted.
Termination criteria: The earlier derived stopping criteria, in (64)–(67), can be deployed with a tolerance range of 10^{-k} (for $k = 3, \ldots, 6$). Moreover, the speed at which the M-ADMM converges depends significantly on the quadratic penalty parameters by Boyd et al. in [3]. However, the primal $\epsilon^{Prim} > 0$ and dual $\epsilon^{Dual} > 0$ tolerances are selected as the termination (stopping) criteria for the convergence of the M-ADMM, and their choices depend on the relative ϵ^{rel} and absolute ϵ^{abs} criteria on account that the ℓ_2 norms are in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. The selected primal and dual tolerances are so small such that the algorithm converges (terminates) whenever $||r^{k+1}||_2 \leq \epsilon^{Prim}$ and $||s^{k+1}||_2 \leq \epsilon^{Dual}$. Usually in literature, the values, $\epsilon^{rel} = 10^{-3}$ and $\epsilon^{abs} = 10^{-3}$, are used as reasonable stopping criteria for the ADMM algorithms and can be reduced to improve the accuracy. Stated below are basic computations in literature as in [3].

$$\epsilon^{Prim} = \sqrt{n}\epsilon^{abs} + \epsilon^{rel}.\max\{||Ax^{k+1}||_2, ||Bu^{k+1}||_2, ||z^{k+1}||_2, ||C||_2\} \quad (77)$$

$$\epsilon^{Dual} = \sqrt{m}\epsilon^{abs} + \epsilon^{rel}.||\rho u||_2 \quad (78)$$

4 Numerical experiments

In this section, the algorithm was implemented with numerical examples on a *Core i3* CPU. The quasi-Newton solver is interfaced with the M-ADMM in obtaining the optimizer. The convergence factor of ADMM for varying

tolerances and penalty parameters of the linear constraint is later illustrated. These examples demonstrate that the M-ADMM method converges for an over-relaxed factor $\alpha^* \in [1.8, 2.0]$ for faster convergence.

Example 1. Consider the one-dimensional convex multi-delay bounded OCP:

$$\begin{aligned}
\text{Minimize } J(x(t),u(t)) &= \frac{1}{2}\int_0^1 [2x^2(t)+u^2(t)]dt \\
\text{subject to} \qquad \dot{x}(t) &\leq -2x(t)+3u(t)+2x(t-0.1)+x(t-0.2) \\
&\quad -x(t-0.5)+u(t-0.1)-2u(t-0.3) \\
x(0) &= 1, \qquad t\in[0,1], \\
x(t) &= 2t^2+1, \qquad t\in[-0.5,0], \\
u(t) &= 3t+2, \qquad t\in[-0.3,0], \\
0 \leq u(t) &\leq 2.
\end{aligned}$$

The discretized positive definite matrices $\bar{P}(\succ 0)$ and $\bar{Q}(\succ 0)$ for the objective functional, using $\delta = 0.1$ for purpose of illustration, are, respectively, given as

$$\bar{P} = \begin{pmatrix} 0.0889 & 0 & \cdots & \cdots & 0 \\ 0 & 0.0444 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0.0889 & 0 \\ 0 & \cdots & \cdots & 0 & 0.0222 \end{pmatrix} \text{ and } \bar{Q} = \begin{pmatrix} 0.0444 & 0 & \cdots & \cdots & 0 \\ 0 & 0.0222 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0.0222 & 0 \\ 0 & \cdots & \cdots & 0 & 0.0111 \end{pmatrix}.$$

While the constraint coefficients are as given in subsection 3.3 using the optimal relaxation parameter $\alpha^* = 2.0$. Table 1 demonstrates the various effects of changing relaxation parameters (factors) on the convergence and results of the M-ADMM algorithm measured by the iterative cycles. The optimal relaxation factor is observed to be $\alpha^* = 1.90$.
Figures 8 and 9 show the effects of the varying penalty parameters ρ_2 on the rate of convergence of the M-ADMM algorithm for increasing relaxation factors α. Clearly, the lines overlap as the algorithm progresses. The optimum values were also obtained at $\alpha^* = 1.90$, as shown in the simulation, indicating an increase in the rate of convergence. Table 2 shows the results when the algorithm was ran for various values of the step-length δ and tolerances for fixed values of ρ_1, ρ_2, ρ_3, and μ.

Figure 10 illustrates primal and dual convergences of the M-ADMM algorithm using the optimal over-relaxation factor $\alpha = 2.00$ while Figure 11 demonstrates the responses of the state-control variables and the objective values for the chosen tolerance (Tol.=10^{-3}). The objective values $J^* = 0.3824$

Tolerances	10^{-3}	10^{-4}	10^{-5}	10^{-6}
No. of iterations	(k)	(k)	(k)	(k)
$\alpha = 1.00$	66	128	193	259
$\alpha = 1.20$	55	107	160	216
$\alpha = 1.40$	47	98	138	189
$\alpha = 1.60$	42	80	122	170
$\alpha = 1.80$	37	71	109	155
$\alpha^* = 2.00$	34	69	130	204

Table 1: Effects of varying relaxation factors on fixed penalty parameters ($\mu = 0.3,\ \rho_1 = 0.1,\ \rho_2 = 0.2,\ \rho_3 = 0.3 : \delta = 0.1$)

Step-lengths	$\delta = 0.10$		$\delta = 0.05$		$\delta = 0.01$	
Tolerances	10^{-3}	10^{-4}	10^{-3}	10^{-4}	10^{-3}	10^{-4}
$\|\|x\|\|$	2.3788	2.3919	2.8830	2.9223	3.2970	3.6359
$\|\|u\|\|$	0.8937	0.8933	0.9271	0.9263	0.9812	0.9796
$\|\|J*\|\|$	0.3827	0.3900	0.3968	0.3982	0.3623	0.3730
$\xi^*_{\bar{R}}$	0.5827	0.5827	0.0528	0.0528	0.0471	0.0471
$\bar{\rho}_{CAL}$	0.0557	0.0557	0.7379	0.7379	0.9383	0.9383
k	11	16	14	25	34	69
Time (secs)	0.0504	0.05623	0.0886	00305	0.3778	0.2057

Table 2: Summary of results at optimum relaxation parameter ($\mu = 0.3,\ \rho_1 = 0.1,\ \rho_2 = 0.2,\ \rho_3 = 0.3,\ \alpha^* = 2.0$)

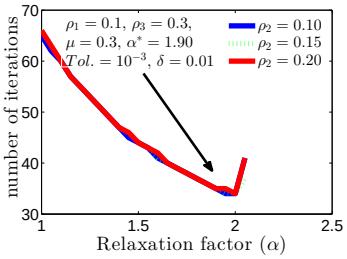


Figure 8: Effects of varying α, ρ on conv. as $||r^k||_2^2 \leq \epsilon^{dual}$ and $||s^k||_2^2 \leq \epsilon^{primal}$

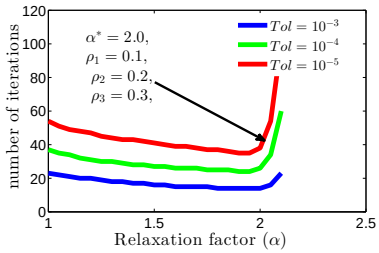


Figure 9: Effects of varying α, Tol. on conv. as $||r^k||_2^2 \leq \epsilon^{dual}$ and $||s^k||_2^2 \leq \epsilon^{primal}$

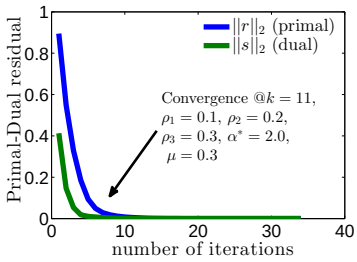


Figure 10: Primal-Dual Convergence: $||r^k||_2 \to r^*$, $||d^k||_2^2 \to 0$

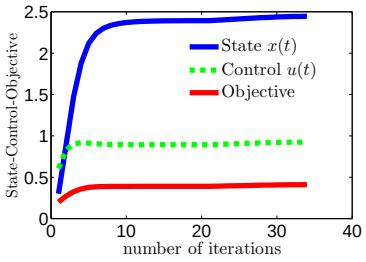


Figure 11: Optimal trajectories: State, control, and objective

were arrived at with the respective optimal state and control values 2.3788 and $0.8937 \in [0, 2]$, respectively.

Example 2. Considering the generalized multi-delay OCP with bounded control:

$$\text{MinJ}(x,u) = \int_0^{0.5} \left[2x_1^2 + x_1x_2 + x_2^2 + u_1^2 + u_1u_2 + u_2^2\right]dt \tag{79}$$

$$\begin{aligned}
\text{s.t.} \quad \dot{x_1}(t) \leq & 2x_1(t) + x_2(t) + u_1(t) + 3u_2(t) + x_1(t-0.1) + x_2(t-0.1) \\
& + 2x_1(t-0.2) + x_2(t-0.2) - x_1(t-0.3) + 2u_1(t-0.1) \\
& + u_2(t-0.1) + u_1(t-0.3) + 2u_2(t-0.2), \\
\dot{x_2}(t) \leq & x_1(t) - u_1(t) + 2u_2(t) - x_1(t-0.1) + x_1(t-0.2) \\
& + 2x_2(t-0.2) - x_1(t-0.3) + u_2(t-0.1) + u_1(t-0.2) \\
& + 3u_2(t-0.2), \qquad t \in [0, 0.5] \qquad (80) \\
x(0) = & (1,1), \qquad (81) \\
x(t) = & (2t+1, t^2+1), \qquad -0.3 \leq t \leq 0, \qquad (82) \\
u(t) = & (2, 2+t), \qquad -0.2 \leq t \leq 0, \qquad (83) \\
& (1,1) \leq u(t) \leq (2,3), \qquad (84)
\end{aligned}$$

where $x(t) \in \mathbb{R}^2$ and $u(t) \in \mathbb{R}^2$.
The problem is re-formulated into the format in (1)–(7) to make it amenable to the M-ADMM algorithm. The coefficient matrices are defined below.

$$P = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix},\ Q = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},\ A = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 1 & 3 \\ -1 & 2 \end{bmatrix},\ \gamma = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ \sigma = \begin{bmatrix} 2 \\ 3 \end{bmatrix},$$

$$\alpha_1 = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix},\ \alpha_2 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},\ \alpha_3 = \begin{bmatrix} -1 & 0 \\ -1 & 0 \end{bmatrix},\ \beta_1 = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix},\ \text{and } \beta_2 = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}.$$

The coefficient matrices, P and Q, of the quadratic functional are symmetric, invertible (non-singular), and positive definite (i.e., $P \succ 0$ and $Q \succ 0$) since their respective eigenvalues ($\lambda_1 = 1.585,\ \lambda_2 = 4.4142$) and ($\lambda_1 = 1,\ \lambda_2 = 3$) are all positive. This ensures that the operators are well-posed for the algorithm. The choice of the step-length $\delta = 0.1$ was made with $N = 5$ such that the coefficient matrices P and Q of the state and control variables for the continuous-time problem yield the block matrices $\bar{P} \in \mathbb{R}^{10\times 10}$ and $\bar{Q} \in \mathbb{R}^{12\times 12}$, respectively. The structures of their block matrix operators, are as defined in (3.2). However, the recurrence relation from the discretization of the constraint, as derived in (23), yields

$$\theta x^{(k)} - x^{(k+1)} + \omega u^{(k)} = -\delta \sum_{j=1}^{3} \alpha_j x^{(k-v_j)} - \delta \sum_{l=1}^{2} \beta_l u^{(k-w_l)}, \tag{85}$$

where $v = r_3/\delta = 3$, $w = q_2/\delta = 2$ and the various associated matrices are defined below:

$$\theta = I_n + A\delta = \begin{bmatrix} 1.2000 & 0.1000 \\ 0.1000 & 1.0000 \end{bmatrix},\ \omega = B\delta = \begin{bmatrix} 0.1000 & 0.3000 \\ -0.1000 & 0.2000 \end{bmatrix},$$

with the formulated block-matrices $\bar{A}$, $\bar{B}$, and C are as defined in subsection 3.3. Table 3 shows the results in the first phase of the algorithm. The M-ADMM was ran for various values of the relaxation parameter α evenly spaced between 1.0 and 2.0 with a step-size of 0.05 for a fixed value of the penalty parameters ($\rho_1 = 0.1,\ \rho_2 = 0.2,\ \rho_3 = 0.3,\ \alpha^* = 2.0$) and tolerance ($Tol = 10^{-3}$) to ascertain the optimum relaxation factor (α^*) using the number of iterations as the measure of convergence.

Table 4 also demonstrates the effects of some of the selected relaxation parameters (factors) on the optimum values of the optimization problem as the tolerances decrease geometrically for fixed values of the penalty parameters. Figures 12 and 13 show the rate of convergence of M-ADMM Algorithm using the primal and dual residuals for increasing values of the iteration. Clearly, both lines converge to zero as the algorithm progresses for the varying values of α and ρ_2 as shown in Figure 14, where the average elapsed time per/cycle falls within a second when run on an Intel computer (*Inspiron 15 Core i3*). The optimum values were obtained at $\alpha^* = 2.0$, as shown in the number of

Step-lengths	$\delta = 0.10$		$\delta = 0.05$		$\delta = 0.01$	
Tolerances	10^{-3}	10^{-4}	10^{-3}	10^{-4}	10^{-3}	10^{-4}
$\|\|x\|\|$	14.5367	14.1757	0.138099	0.135782	0.138099	0.135782
$\|\|u\|\|$	2.7139	2.7203	0.15	144	0.22	164
$\|\|J*\|\|$	18.3905	18.6315	0.16	140	0.22	164
ξ_R^*	0.6910	0.6910	0.16	140	0.22	164
$\bar{\rho}_{CAL}$	0.1612	0.1612	0.16	140	0.22	164
k	117	138	0.16	140	0.22	164
Time (secs)	0.2678	0.3852	0.16	140	0.22	164

Table 3: Summary of results at optimum relaxation parameter ($\rho_1 = 0.1,\ \rho_2 = 0.2,\ \rho_3 = 0.3, \alpha^* = 2.0$)

Tolerances	10^{-3}	10^{-4}	10^{-5}	10^{-6}
No. of iterations	(k)	(k)	(k)	(k)
$\alpha = 1.00$	239	467	703	940
$\alpha = 1.20$	199	389	585	786
$\alpha = 1.40$	175	345	523	700
$\alpha = 1.60$	155	307	464	622
$\alpha = 1.80$	140	277	419	562
$\alpha^* = 2.00$	127	253	383	514

Table 4: Effects of varying relaxation factors on fixed penalty parameters ($\rho_1 = 0.1,\ \rho_2 = 0.2,\ \rho_3 = 0.3 : \delta = 0.1$)

iterations per cycle, indicating a decrease in the rate of convergence. The M-ADMM ran for various values of the step-length δ geometrically spaced between 0.010 and 0.10 with a constant factor of 0.5 using the tolerances of 10^{-3} and 10^{-4}. Figure 14 further illustrates the faster rate of convergence

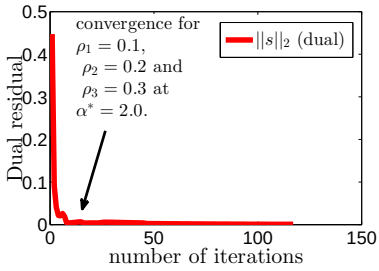


Figure 12: Dual convergence: $||d^k||_2^2 \to 0$, μ and $\alpha^* = 2.0$

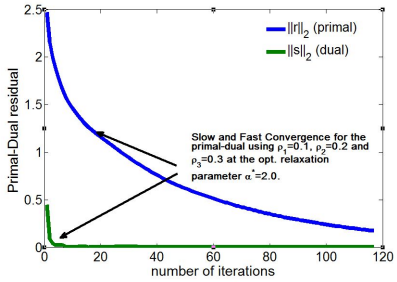


Figure 13: Primal-dual convergence: $||r^k||_2 \to r^*,\ ||d^k||_2^2 \to 0$

of the M-ADMM with the optimal over-relaxation factor $\alpha = 2.00$ compared to the non-relaxed ADMM for which $\alpha^* = 1$. However, Figure 15 demon-

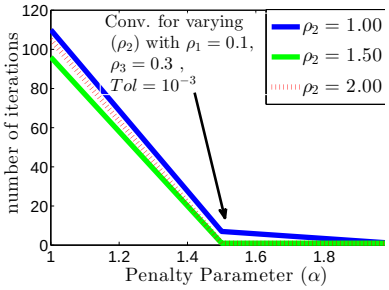


Figure 14: Convergence for varying α, ρ_2: ($\rho_1 = 0.1,\ \rho_3 = 0.3,\ \alpha^* = 2.0$)

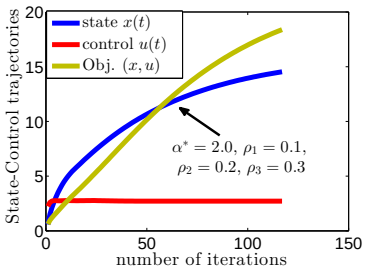


Figure 15: Optimal trajectories: State, control, and objective

strates the responses of the state-control variables and the objective values to increasing iterations at chosen tolerance (Tol.=10^{-3}), relaxation parameters ($\rho_1 = 0.1$, $\rho_2 = 0.2$, $\rho_3 = 0.3$) and optimal penalty parameter $\alpha^* = 2.0$. In addition, the objective values are observed to increase with increasing values of the state variables until the optimal objective value $J^* = 18.3905$ was arrived at with the respective optimum state and control trajectories stated as

$$x^* = \begin{bmatrix} 4.1416 \\ 13.9343 \end{bmatrix} \text{ and } u^* = \begin{bmatrix} 1.0000 \\ 2.5230 \end{bmatrix} \text{ while } \begin{bmatrix} ||x^*|| \\ ||u^*|| \end{bmatrix} = \begin{bmatrix} 14.5367 \\ 2.7139 \end{bmatrix}$$

and $||(1,\,2)^T|| \leq ||u^*|| \leq ||(2,\,3)^T||$. Applying the formulas in (69) and (70) yields the results for the calculated penalty parameter and convergence factor as $\bar{\rho}^* = 0.1612$ and $xi^*_{\bar{R}} = 0.6910$, respectively, where $\bar{P} \in \mathbb{R}^{(4N+2)\times(4N+2)}$ and $\bar{A} = [A, B] \in \mathbb{R}^{2N\times(4N+2)}$.

5 Conclusion

The ADMM for strictly convex QP with a particular focus on the QPs arising from discretized multi-delay OCP with bounded control was studied. A convergence proof and a method of analysis of the problem were proposed that allows the selection of the optimal relaxation parameter for the acceleration of the algorithm for varying penalty parameters of the augmented Lagrangian functional; while other parameters remain fixed. A major drawback of this approach is that, the penalty parameters for both the linear and bound constraints cannot be derived simultaneously. In the future, we will extend the computational framework to include the effects of both varying parameters in the computational complexity of the algorithm. This research was limited to OCP with linear constraints of non-fractional order. However, in subsequent work, the scope of the work will be extended to convex constrained OCPs

with coupled and/or nonlinear differential constraints with both integer and (or) fractional order.

Acknowledgments

I am grateful to Prof. Montaz Ali of *The University of The Witwatersrand, SA* for introducing me to the ADMM and for his helpful comments in improving the presentation of this article.

References

1. Bashier, E.B.M. and Patidar, K.C. *Optimal control of an epidemiological model with multiple time delays,* Appl. Math. Comput. 292 (2017), 47–56.

2. Boley, D. *Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs,* SIAM J. Optim. 23(4) (2013), 2183–2207.

3. Boyd, S., Parikh, N., Chu, E. and Eckstein, J. *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn. 3(1) (2011), 11–22.

4. Boyd, S. and Vandenberghe, L. *Convex Optimization*, Cambridge University Press, NY, 2004.

5. Carreira-Perpiñán, M.A. *An ADMM algorithm for solving a proximal bound-constrained quadratic program*, arXive:1412.8493[math.OC], 2014.

6. Combettes, P. and Pesquet, J. *Proximal splitting methods in signal processing*, Fixed-point algorithms for inverse problems in science and engineering, 185–212, Springer Optim. Appl., 49, Springer, New York, 2011.

7. Eckstein, J. and Ferris, M. *Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control*, INFORMS J. Comput. 10(2) (1998), 218–235.

8. Ghadimi, E., Teixeira, A., Shames, I. and Johansson, M. *Optimal Parameter Selection for the Alternating Direction Method of Multipliers (ADMM): Quadratic Problems,* Linnaeus center, Elect. Eng., Royal Inst. of Technology, Sidney, 2014.

9. Gollman, L., Kern, D. and Maurer, H. *Optimal control problems with delays in state and control variables subject to mixed control-state constraints,* Optimal Control Appl. Methods 30(4) (2009), 341–365.

10. Gollman, L. and Maurer, H. *Theory and Applications of Optimal Control Problems with Multiple Delays*, J. Ind. Manag. Optim. 10 (2014), 413–441.

11. Guinn, T. *Reduction of delayed optimal control problems to nondelayed problems*, J. Optim. Theory Appl. 18 (1976), 371–377.

12. He, B., Hong, X. and Yuan, X. *On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM*, J. Sci. Comput. 66 (2016), 1204–1217.

13. He, B. and Yuan, X. *Block-wise alternating direction method of multipliers for multiple-block convex programming and beyond*, SSMAI J. Comput. Math. 691 (2015), 145–174.

14. Laarabi, H., Abta, A. and Hattaf, K. *Optimal control of a delayed SIRS epidemic model with vaccination and treatment,* Acta Biotheor, 63 (2015), 87–97.

15. Moreau, J.J. *Fonctions convexes duales et points proximaux dans un espace hilbertien*, (French) C. R. Acad. Sci. Paris 255 (1962), 2897–2899.

16. Nesterov, Y.E. *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), 543–547.

17. Olotu, O. and Dawodu, K.A. *On the discretized algorithm for optimal proportional control constrained by delay differential equation,* Journal of Mathematical Theory and Modelling by IISTE, 3(8) (2013), 157–169.

18. Olotu, O. and Dawodu, K.A. *Numerical solution to one-dimensional multi-delay system using the modified alternating direction method of multipliers*, Journal of the Nigerian Association of Mathematical Physics, Nigeria. 50 (2019), 107–116.

19. Olotu, O., Dawodu, K.A. and Yusuf, T.T. *Algorithm for solving multi-delay optimal control problems using modified alternating direction method of multipliers*, J. Appl. Computat. Math. 8 (3) (2019), 1000445.

20. Pasquale, L.D. and Gerardo, T. *Quadratic programming with bound constraints*, Encyclopedia of Optimization, USA, 2009.

21. Rihan, F.A. and Anwar, M.N. *Qualitative analysis of delayed SIR epidemic model with a saturated incidence rate,* Int. J. Differ. Equ. 2012, Art. ID 408637, 13 pp.

22. Rockafellar, R.T. *Monotone operators and the proximal point algorithm*, SIAM J. Control and Optim., 14 (5), (1976), 877–898.

23. Sun, D., Toh, K.C. and Yang, L. *A convergent 3-block semi-proximal alternating direction method of multipliers for conic programming with 4-type of constraints*, SIAM J. Optim., 25(2) (2015) 882–915.

24. Voglis, C. and Lagaris, I.E. *Boxcqp: An algorithm for bound constrained convex quadratic problems* 1st IC-SCCE, Athens, 2004.

25. Wu, D. and Shang, Y. *Complete solutions to general box-constrained global optimization problems*, J. Appl. Math. 2011, Art. ID 478608, 17 pp.

26. Xinmin, L., Andrew, H.B., Zachary, G. and Rodney, D. *Constrained quadratic optimization for radiation treatment planning,* American Control Conference (ACC), Boston, USA, 2016.

27. Xue, Y. and Duan, X. *Dynamic analysis of an SIR epidemic model with nonlinear incidence rate and double delays*, Intl. J. Inf. Syst. Sci. Computer Inf. 7 (2011), 92–102.

28. Yao, Y., Zhu, X., Dong, H., Wu, S., Wu, H., Carol Tong, L. and Zhou, X. *ADMM-based problem decomposition scheme for vehicle routing problem with time windows*, Transp. Res. B: Meth. 129, (2019), 156–174.

Appendix

A. Derivation of $\bar{P}$ and $\bar{Q}$

By Simpson's rule, if for each $k = 1, 2, \ldots, \frac{N}{2} - 1$, $f(t_{2k})$ appears in the term corresponding to the interval $[t_{2k-2}, t_{2k}]$ and $f(t_{2k-1})$ appears in the term corresponding to the interval $[t_{2k}, t_{2k+2}]$ for $k = 1, 2, \ldots, \frac{N}{2}$, then the approximation of the integral in (18), given that $F(t, x(t), u(t)) = x^T P x + u^T Q u$, can be expressed as

$$I := \frac{1}{2}\int_{t_0}^{T} F(t, x(t), u(t))\, dt \simeq \frac{1}{2}\sum_{k=1}^{N-1} F(t_k, x^{(k)}, u^{(k)}) \tag{86}$$
$$\simeq \frac{1}{2}\left[\frac{\delta}{3}\Big(f^{(0)} + 2\sum_{k=1}^{\frac{N}{2}-1} f^{(2k)} + 4\sum_{k=1}^{\frac{N}{2}} f^{(2k-1)} + f^{(N)}\Big)\right],$$

where

$$f^{(0)} = (x^{(0)})^T P x^{(0)} + (u^{(0)})^T Q u^{(0)}, \tag{87}$$
$$f^{(2k)} = (x^{(2k)})^T P x^{(2k)} + (u^{(2k)})^T Q u^{(2k)}, \tag{88}$$
$$f^{(2k-1)} = (x^{(2k-1)})^T P x^{(2k-1)} + (u^{(2k-1)})^T Q u^{(2k-1)}, \text{ and} \tag{89}$$
$$f^{(N)} = (x^{(N)})^T P x^{(N)} + (u^{(N)})^T Q u^{(N)}. \tag{90}$$

Substituting equations (87)–(90) into (86) yields

$$I \approx \frac{\delta}{6}(x^{(0)})^T P x^{(0)} + \frac{1}{2}\Big[\frac{2\delta}{3}\sum_{k=1}^{\frac{N}{2}-1}(x^{(2k)})^T P x^{(2k)} + \frac{4\delta}{3}\sum_{k=1}^{\frac{N}{2}}(x^{(2k-1)})^T P x^{(2k-1)}$$
$$+ \frac{\delta}{3}(x^{(N)})^T P x^{(N)}\Big] + \frac{1}{2}\Big[\frac{\delta}{3}(u^{(0)})^T P u^{(0)} + \frac{2\delta}{3}\sum_{k=1}^{\frac{N}{2}-1}(u^{(2k)})^T P u^{(2k)}$$
$$+ \frac{4\delta}{3}\sum_{k=1}^{\frac{N}{2}}(u^{(2k-1)})^T P u^{(2k-1)} + \frac{\delta}{3}(u^{(N)})^T P u^{(N)}\Big]. \tag{91}$$

Expanding (91) for various values of k gives

$$I \approx \frac{\delta}{6}(x^{(0)})^T P x^{(0)} + (x_1^{(1)}, x_1^{(2)}, \ldots, x_n^{(N)}) \begin{pmatrix} \frac{4\delta}{3}P & 0 & \cdots & \cdots & 0 \\ 0 & \frac{2\delta}{3}P & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{4\delta}{3}P & 0 \\ 0 & \cdots & \cdots & 0 & \frac{\delta}{3}P \end{pmatrix} \begin{pmatrix} x_1^{(1)} \\ x_1^{(2)} \\ \vdots \\ \vdots \\ x_n^{(N)}) \end{pmatrix}$$

$$+ \ (u_1^{(0)}, u_1^{(1)}, \cdots, u_m^{(N-1)}, u_m^{(N)}) \begin{pmatrix} \frac{\delta}{3}Q & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \frac{4\delta}{3}Q & \ddots & \cdots & \cdots & \vdots \\ \vdots & \ddots & \frac{2\delta}{3}Q & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \frac{4\delta}{3}Q & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & \frac{\delta}{3}Q \end{pmatrix} \begin{pmatrix} u_1^{(0)} \\ u_1^{(1)} \\ \vdots \\ \vdots \\ u_m^{(N-1)}) \\ u_m^{(N)}) \end{pmatrix}$$

expressed as

$$I \approx \min_{\bar{x},\bar{u}} \ \frac{1}{2}\,\bar{x}^T \bar{P} \bar{x} + \frac{1}{2}\,\bar{u}^T \bar{Q} \bar{u} + R, \tag{92}$$

where any elements $\bar{p}_{ij} \in \bar{P}$ and $\bar{q}_{ij} \in \bar{Q}$ are described as

$$[\bar{p}_{ij}] = \begin{cases} \frac{4\delta}{3}P, & i = j(\text{odd}) & 1 \le i \le N-1, \\ \frac{2\delta}{3}P, & i = j(\text{even}) & 2 \le i \le N, \\ \frac{2\delta}{3}P, & i = j, \ i = N, & \\ 0 & \text{elsewhere} & \end{cases}$$

and

$$[\bar{q}_{ij}] = \begin{cases} \frac{\delta}{3}Q, & i = j, \ i = 1, N, & \\ \frac{4\delta}{3}Q, & i = j(\text{even}) & 2 \le i \le N, \\ \frac{2\delta}{3}Q, & i = j(\text{odd}) & 3 \le i \le N, \\ 0 & \text{elsewhere.} & \end{cases}$$

B. Flowchart for the M-ADMM Algorithm

The diagram below (Figure 16) is the flowchart demonstrating the procedure of the M-ADMM algorithm in handling the discretized optimal control (or constrained optimization) problem with bounded (boxed) control. The left-side of the flowchart is the M-ADMM subroutine, which is the outer loop of Algorithm. However, the right-side of the flowchart is the optimization solver, which in this case is the "quasi-Newton solver" of the low-memory BFGS type.

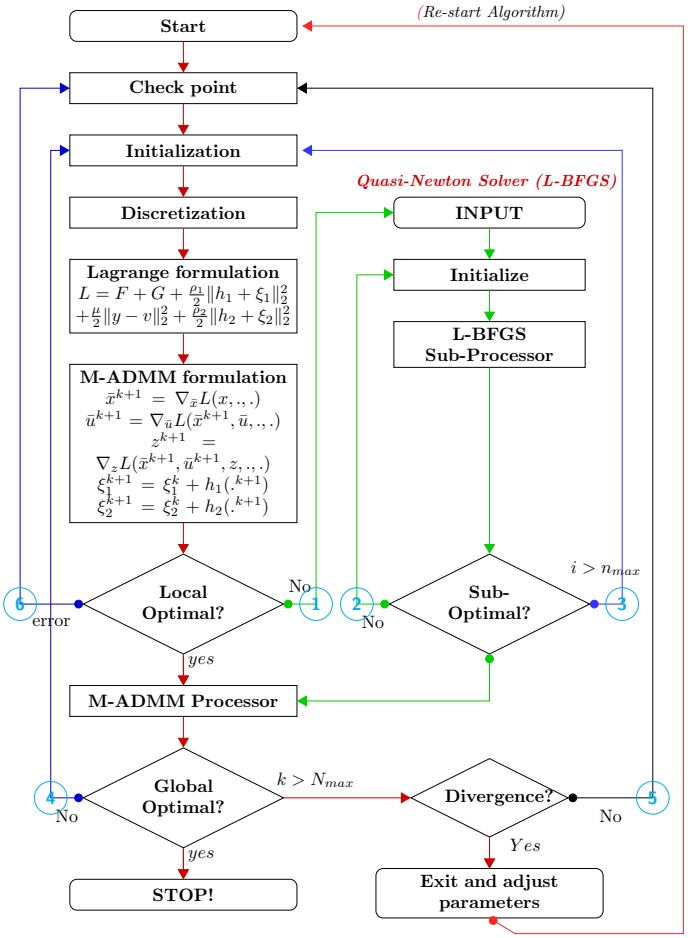


Figure 16: Flowchart for M-ADMM with bounded control

How to cite this article

Dawodu, K. Modified ADMM algorithm for solving proximal bound formulation of multi-delay optimal control problem with bounded control. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 1-36. doi: 10.22067/ijnao.2021.69185.1020.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 37-53
DOI:10.22067/ijnao.2021.69416.1021
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Numerical solution of a system of Volterra integral equations in application to the avian human influenza epidemic model

R. Katani

Abstract

We propose an efficient multistage method for solving a system of linear and nonlinear Volterra integral equations of the second kind. This numerical method is based on the Gauss–Legendre quadrature rule that obtains several values of unknown function at each step, and it will be shown that the order of the convergence is $O(M^{-4})$, where M is the number of the nodes in the time discretization. The method has also the advantages of simplicity of application, less computational time, and useful performance for large intervals. In order to show the efficiency of the method, numerical results for the avian human influenza epidemic model is obtained that is comparable with the fourth-order Runge–Kutta method.

AMS subject classifications (2020): 45D05; 65R20.

Keywords: The avian human influenza epidemic model; system of Volterra integral equations; Gauss–Legendre quadrature rule; convergence order.

1 Introduction

Numerical methods for solving a variety of system of Volterra integral equations (SVIE) are offered in many publications. Here readers may refer to [1, 3, 7, 10] and others.
In this article, we introduce a multistage numerical method for solving SVIE, and then the introduced method will be used to the avian human influenza epidemic model, which recently has been described by Iwami, Takeuchi, and Liu [6]. This mathematical model is proposed to interpret the spread of the

Received 15 March 2021; revised 1 August 2021; accepted 1 August 2021
Roghayeh Katani
Faculty of Sciences, Yasouj University, Yasouj-Iran. email: katani@yu.ac.ir

avian influenza from the bird world to the human world. The avian influenza has caused the death of millions of birds (almost 100% death) and it has become a disease of great importance both for animal and human health. Fortunately, there is still no evidence that the avian influenza virus can be transmitted among humans [12], but this virus is unstable and lacks of gnomic proofreading mechanism. Therefore the small errors, which occur when the copies themselves, go undetected and uncorrected. Since the specifics of mutations and evolution of the influenza viruses cannot be predicted, it is hardly possible to know if or when a virus such as the avian influenza, might acquire the properties needed to spread easily and sustainably among humans [5]. However, experts warn about an occurrence of the so-called "mutant the avian influenza", which can be easily transmitted among humans with potentially devastating consequence. The US Congressional Budget Office has formally modeled the likely consequences of pandemic influenza and estimates that up to 2 million of the US population might die, with up to 40% of all workers ill for as long as three or more weeks [11]. Thus, it is necessary and urgent to investigate the transmission process of the avian influenza and to take efficient measures to control the spread of the avian influenza.

2 Preliminaries

In this section, we introduce notations, definitions, and preliminaries facts, which are used throughout this article. Consider SVIE

$$Y(t) = G(t) + \int_0^t K(t, s, Y^t(s))ds, \qquad 0 \le s \le t \le T, \tag{1}$$

where $Y(t) = [y^{(1)}(t), y^{(2)}(t), \ldots, y^{(n)}(t)]^t$ is the desired function, $G(t) = [g^{(1)}(t), g^{(2)}(t), \ldots, g^{(n)}(t)]^t$, and

$$K(t, s, Y^t(s)) = \begin{bmatrix} k_1(t, s, y^{(1)}(t), y^{(2)}(t), \ldots, y^{(n)}(t)) \\ k_2(t, s, y^{(1)}(t), y^{(2)}(t), \ldots, y^{(n)}(t)) \\ \vdots \\ k_n(t, s, y^{(1)}(t), y^{(2)}(t), \ldots, y^{(n)}(t)) \end{bmatrix},$$

are the source functions (A^t is the transposed matrix of A). Let system (1) is uniquely solvable. Necessary and sufficient conditions for the existence and uniqueness of solution (1) can be found in [9]. Therefore we assume the following conditions:
(I) The source function $G(t)$ is continuous (i.e., each component is continuous).
(II) The kernel $K(t, s, U)$ is a continuous function for $0 \le s \le t \le T$ and $0 \le \| U \| < \infty$.

(III) The kernel K satisfies the Lipschitz condition

$$\| K(t,s,U) - K(T,s,V) \| \leq L \| U - V \|,$$

where the norm is defined as $\| U \| = \max_{0 \leq i \leq n} |u_i(t)|$.

3 Numerical procedure

Let $I_h = \{t_m = mh, m = 0, 1, \ldots, M\}$ with $t_M = Mh = T$ be a given uniform mesh on $[0, T]$. For given real numbers c_j with $0 = c_0 < c_1 < \cdots < c_p = 1$, we choose the set of mesh points as $\Pi_h := \{t_{m,j} = t_m + c_j h, m = 0, 1, \ldots, M - 1, j = 0, 1, \ldots, p\}$, where c_j, $j = 1, 2, \ldots, p-1$, are the roots of the $(p-1)$th Legendre polynomial. The Legendre polynomials $P_{p-1}(x)$ are orthogonal with respect to the weight functions $w(x) = 1$ over the interval $[-1, 1]$, that is,

$$\int_{-1}^{1} P_m(x) P_n(x) dx = \delta_{nm},$$

where δ_{mn} is the Kronecker delta (see, for example, [8, 4] for Legendre polynomials and [13] for funding roots of polynomials).

For simplicity, let $p = 3$ ($c_1 = \frac{3-\sqrt{3}}{6}$, $c_2 = \frac{3+\sqrt{3}}{6}$) and let $n = 2$ (system dimensional). Then we will have a multistep method with at least four order of convergence, which one can generalize it by increasing the Legendre polynomial degree.
Assume that $y^{(i)}_{m,j} \approx y^{(i)}(t_{m,j})$, $m = 0, 1, \ldots, M-1$, $j = 0, 1, 2, 3$ and that $Y_{m,j} \approx Y(t_{m,j}) = [y^{(1)}(t_{m,j}), y^{(2)}(t_{m,j})]^t$. Thus system (1) can be rewrite as

$$\begin{aligned} y^{(1)}(t_{m,j}) =& g^{(1)}(t_{m,j}) + \int_0^{t_m} k_1(t_{m,j}, s, y^{(1)}(s), y^{(2)}(s)) ds \\ &+ \int_{t_m}^{t_{m,j}} k_1(t_{m,j}, s, y^{(1)}(s), y^{(2)}(s)) ds, \end{aligned} \tag{2}$$

$$\begin{aligned} y^{(2)}(t_{m,j}) =& g^{(2)}(t_{m,j}) + \int_0^{t_m} k_2(t_{m,j}, s, y^{(1)}(s), y^{(2)}(s)) ds \\ &+ \int_{t_m}^{t_{m,j}} k_2(t_{m,j}, s, y^{(1)}(s), y^{(2)}(s)) ds, \end{aligned} \tag{3}$$

$$m = 0, 1, \ldots, M-1, \ j = 1, 2, 3.$$

Note that $Y_{0,0} = [y^{(1)}_{0,0}, y^{(2)}_{0,0}]^t = [g^{(1)}(0), g^{(2)}(0)]^t$, and suppose that approximations $Y_{0,1}, Y_{0,2}, Y_{1,0}, \ldots, Y_{n,0}$, have been calculated in the previous steps. Therefore the first integral in relations (2) and (3) can be estimated by using the Gauss–Legendre quadrature rule,

$$\int_0^{t_m} k_1(t_{m,j},s,y^{(1)}(s),y^{(2)}(s))ds = \sum_{i=0}^{m-1}\int_{t_i}^{t_{i+1}} k_1(t_{m,j},s,y^{(1)}(s),y^{(2)}(s))ds$$
$$\approx \frac{h}{2}\sum_{i=0}^{m-1}\left[k_1(t_{m,j},t_{i,1},y_{i,1}^{(1)},y_{i,1}^{(2)}) + k_1(t_{m,j},t_{i,2},y_{i,2}^{(1)},y_{i,2}^{(2)})\right] =: A_1(m,j),$$
$$\int_0^{t_m} k_2(t_{m,j},s,y^{(1)}(s),y^{(2)}(s))ds = \sum_{i=0}^{m-1}\int_{t_i}^{t_{i+1}} k_2(t_{m,j},s,y^{(1)}(s),y^{(2)}(s))ds$$
$$\approx \frac{h}{2}\sum_{i=0}^{m-1}\left[k_2(t_{m,j},t_{i,1},y_{i,1}^{(1)},y_{i,1}^{(2)}) + k_2(t_{m,j},t_{i,2},y_{i,2}^{(1)},y_{i,2}^{(2)})\right] =: A_2(m,j). \tag{4}$$

If $j=3$, then the second integral in relations (2) and (3) can be similarly approximated as

$$\int_{t_m}^{t_{m,3}} k_1(t_{m,3},s,y^{(1)}(s),y^{(2)}(s))ds$$
$$\approx \frac{h}{2}\left[k_1(t_{m,3},t_{m,1},y_{m,1}^{(1)},y_{m,1}^{(2)}) + k_1(t_{m,3},t_{m,2},y_{m,2}^{(1)},y_{m,2}^{(2)})\right],$$
$$\int_{t_m}^{t_{m,3}} k_2(t_{m,3},s,y^{(1)}(s),y^{(2)}(s))ds$$
$$\approx \frac{h}{2}\left[k_2(t_{m,3},t_{m,1},y_{m,1}^{(1)},y_{m,1}^{(2)}) + k_2(t_{m,3},t_{m,2},y_{m,2}^{(1)},y_{m,2}^{(2)})\right], \tag{5}$$

and for $j=1$, by using this quadrature rule, we obtain

$$\int_{t_m}^{t_{m,1}} k_1(t_{m,1},s,y^{(1)}(s),y^{(2)}(s))ds$$
$$\approx \frac{c_1 h}{2}\left[k_1(t_{m,1},t_m+\frac{h}{6},y^{(1)}(t_m+\frac{h}{6}),y^{(2)}(t_m+\frac{h}{6}))\right.$$
$$\left.+k_1(t_{m,1},t_m+\frac{2-\sqrt{3}}{6}h,y^{(1)}(t_m+\frac{2-\sqrt{3}}{6}h),y^{(2)}(t_m+\frac{2-\sqrt{3}}{6}h))\right],$$
$$\int_{t_m}^{t_{m,1}} k_2(t_{m,1},s,y^{(1)}(s),y^{(2)}(s))ds$$
$$\approx \frac{c_1 h}{2}\left[k_2(t_{m,1},t_m+\frac{h}{6},y^{(1)}(t_m+\frac{h}{6}),y^{(2)}(t_m+\frac{h}{6}))\right.$$
$$\left.+k_2(t_{m,1},t_m+\frac{2-\sqrt{3}}{6}h,y^{(1)}(t_m+\frac{2-\sqrt{3}}{6}h),y^{(2)}(t_m+\frac{2-\sqrt{3}}{6}h))\right]. \tag{6}$$

Finally, for $j=2$, we obtain

$$\int_{t_m}^{t_{m,2}} k_1(t_{m,2},s,y^{(1)}(s),y^{(2)}(s))ds$$
$$\approx \frac{c_2h}{2}\left[k_1(t_{m,2},t_m+\frac{h}{6},y^{(1)}(t_m+\frac{h}{6}),y^{(2)}(t_m+\frac{h}{6}))\right.$$
$$\left.+k_1(t_{m,2},t_m+\frac{2+\sqrt{3}}{6}h,y^{(1)}(t_m+\frac{2+\sqrt{3}}{6}h),y^{(2)}(t_m+\frac{2+\sqrt{3}}{6}h))\right],$$
$$\int_{t_m}^{t_{m,2}} k_2(t_{m,2},s,y^{(1)}(s),y^{(2)}(s))ds$$
$$\approx \frac{c_2h}{2}\left[k_2(t_{m,2},t_m+\frac{h}{6},y^{(1)}(t_m+\frac{h}{6}),y^{(2)}(t_m+\frac{h}{6}))\right.$$
$$\left.+k_2(t_{m,2},t_m+\frac{2+\sqrt{3}}{6}h,y^{(1)}(t_m+\frac{2+\sqrt{3}}{6}h),y^{(2)}(t_m+\frac{2+\sqrt{3}}{6}h))\right]. \tag{7}$$

In relations (6) and (7), $t_m+\frac{2\pm\sqrt{3}}{6}h$ and $t_m+\frac{h}{6}$ do not belong to the mesh points Π_h. Then we will have a problem to computing $y^{(i)}(t_m+\frac{2\pm\sqrt{3}}{6}h)$ and $y^{(i)}(t_m+\frac{h}{6})$, $i=1,2$. In order to overcome to this difficulty, we use the Lagrange interpolation

$$y^{(i)}(x)\approx \mathcal{P}_i(x)=\sum_{i'=0}^{3} L_{i'}(x)y^{(i)}_{m,i'}, \qquad L_{i'}(x)=\prod_{\substack{j'=0\\ j'\neq i'}}^{3}\frac{x-t_{m,j'}}{c_{i'}-c_{j'}},$$

for $x=t_m+\frac{2\pm\sqrt{3}}{6}h$ or $x=t_m+\frac{h}{6}$. Substituting these approximations into (2), leads to the system of algebraic equations

$$y^{(i)}_{m,1}=g^{(i)}(t_{m,1})+A_i(m,1)+\frac{c_1h}{2}\left[k_i(t_{m,1},t_m+\frac{h}{6},\mathcal{P}_1(t_m+\frac{h}{6}),\mathcal{P}_2(t_m+\frac{h}{6}))\right.$$
$$\left.+k_i(t_{m,1},t_m+\frac{2-\sqrt{3}}{6}h,\mathcal{P}_1(t_m+\frac{2-\sqrt{3}}{6}h),\mathcal{P}_2(t_m+\frac{2-\sqrt{3}}{6}h))\right],$$
$$y^{(i)}_{m,2}=g^{(i)}(t_{m,2})+A_i(m,2)+\frac{c_2h}{2}\left[k_i(t_{m,2},t_m+\frac{h}{6},\mathcal{P}_1(t_m+\frac{h}{6}),\mathcal{P}_2(t_m+\frac{h}{6}))\right.$$
$$\left.+k_i(t_{m,2},t_m+\frac{2+\sqrt{3}}{6}h,\mathcal{P}_1(t_m+\frac{2+\sqrt{3}}{6}h),\mathcal{P}_2(t_m+\frac{2+\sqrt{3}}{6}h))\right],$$
$$y^{(i)}_{m,3}=g^{(i)}(t_{m,3})+A_i(m,3)$$
$$+\frac{h}{2}\left[k_i(t_{m,3},t_{m,1},y^{(1)}_{m,1},y^{(2)}_{m,1})+k_i(t_{m,3},t_{m,2},y^{(1)}_{m,2},y^{(2)}_{m,2})\right],$$
$$i=1,2. \tag{8}$$

This system can be solved by using an iterative method such as Newton–Raphson method or by using a suitable software package such as Maple or MATLAB

4 Convergence analysis

Theorem 1. The approximation method given by system (8) is convergent, and its order of convergence is at least 4 for the functions k_i and $g^{(i)}$ $(i = 1, 2)$ with at least fourth-order continuous derivatives.

Proof. Define $E_{m,j} = Y_{m,j} - Y(t_{m,j})$, where $Y(t_{m,j}), Y_{m,j} \in R^2$ denote, respectively, the exact and approximate solutions of (1) at the point $t = t_{m,j}$. Subtracting (6) from (2) and (3) (for $j = 1$) leads to

$$\begin{aligned}
E_{m,1} =& Y_{m,1} - Y(t_{m,1}) \\
=& \frac{h}{2}\sum_{i=0}^{m-1}\left[K(t_{m,1}, t_{i,1}, Y_{i,1}^t) + K(t_{m,1}, t_{i,2}, Y_{i,2}^t)\right] \\
&+ \frac{c_1 h}{2}\Big[K(t_{m,1}, t_m + \frac{h}{6}, \mathcal{P}(t_m + \frac{h}{6})) \\
&+ K(t_{m,1}, t_m + \frac{2-\sqrt{3}}{6}h, \mathcal{P}(t_m + \frac{2-\sqrt{3}}{6}h))\Big] \\
&- \int_0^{t_m} K(t_{m,1}, s, Y^t(s))ds - \int_{t_m}^{t_{m,1}} K(t_{m,1}, s, Y^t(s))ds,
\end{aligned}$$

where $\mathcal{P}(x) = [\mathcal{P}_1(x), \mathcal{P}_2(x)]$. By adding and diminishing the terms

$$\frac{h}{2}\sum_{i=0}^{m-1}\left[K(t_{m,1}, t_{i,1}, Y^t(t_{i,1})) + K(t_{m,1}, t_{i,2}, Y^t(t_{i,2}))\right],$$

and

$$\frac{c_1 h}{2}\left[K(t_{m,1}, t_m + \frac{h}{6}, Y^t(t_m + \frac{h}{6})) + K(t_{m,1}, t_m + \frac{2-\sqrt{3}}{6}h, Y^t(t_m + \frac{2-\sqrt{3}}{6}h))\right],$$

we can write

$$\begin{aligned}
||E_{m,1}|| \leq& \frac{h}{2}\sum_{i=0}^{m-1}\left[||K(t_{m,1}, t_{i,1}, Y_{i,1}^t) - K(t_{m,1}, t_{i,1}, Y^t(t_{i,1}))|| \right.\\
&\left. + ||K(t_{m,1}, t_{i,2}, Y_{i,2}^t) - K(t_{m,1}, t_{i,2}, Y^t(t_{i,2}))||\right] \\
&+ \frac{c_1 h}{2}\Big[||K(t_{m,1}, t_m + \frac{h}{6}, \mathcal{P}(t_m + \frac{h}{6})) - K(t_{m,1}, t_m + \frac{h}{6}, Y^t(t_m + \frac{h}{6}))||
\end{aligned}$$

$$+||K(t_{m,1},t_m+\frac{2-\sqrt{3}}{6}h,\mathcal{P}(t_m+\frac{2-\sqrt{3}}{6}h))$$
$$-K(t_{m,1},t_m+\frac{2-\sqrt{3}}{6}h,Y^t(t_m+\frac{2-\sqrt{3}}{6}h))||\Big]$$
$$+(m+1)||e(G)||.$$

In the previous relation, $e(G)$ is the upper bound of the Gauss–Legendre integration error, that is,

$$||\frac{h}{2}\sum_{i=0}^{m-1}\left[K(t_{m,1},t_{i,1},Y^t(t_{i,1}))+K(t_{m,1},t_{i,2},Y^t(t_{i,2}))\right]-\sum_{i=0}^{m-1}\int_{t_i}^{t_{i+1}}K(t_{m,1},s,Y^t(s))ds||$$
$$+||\frac{c_1h}{2}\left(K(t_{m,1},t_m+\frac{h}{6},Y^t(t_m+\frac{h}{6}))+K(t_{m,1},t_m+\frac{2-\sqrt{3}}{6}h,Y^t(t_m+\frac{2-\sqrt{3}}{6}h))\right)$$
$$-\int_{t_m}^{t_{m,1}}K(t_{m,1},s,Y^t(s))ds||\leq m||e(G)||+||e(G)||.$$

If $k_i\in C^4([0,1])$ $(i=1,2)$, then $e(G)=\frac{1}{4320}h^4k^{(4)}(t,\widetilde{t},Y^t(\widetilde{t}))$ where $0<\widetilde{t}<1$; see [2].
In the following, by using the Lipschitz condition for the kernel K, we have

$$||E_{m,1}||\leq\frac{hL}{2}\sum_{i=0}^{m-1}(||E_{i,1}||+||E_{i,2}||)+\frac{c_1hL}{2}\Big[||\mathcal{P}(t_m+\frac{h}{6}))-Y^t(t_m+\frac{h}{6}))||$$
$$+||\mathcal{P}(t_m+\frac{2-\sqrt{3}}{6}h))-Y^t(t_m+\frac{2-\sqrt{3}}{6}h))||\Big]+(m+1)||e(G)||.$$

Let $I(t)$ be the Lagrange interpolation error, that is, $I(t)=Y^t(t)-\mathcal{P}(t)$. Thus

$$||E_{m,1}||\leq\frac{hL}{2}\sum_{i=0}^{m-1}(||E_{i,1}||+||E_{i,2}||)$$
$$+\frac{c_1hL}{2}\left[||I(t_m+\frac{h}{6}))||+||I(t_m+\frac{2-\sqrt{3}}{6}h))||\right]+(m+1)||e(G)||.$$

Without loss of generality, assume $\max_{j=1,2,3}||E_{m,j}||=||E_{m,1}||$. Then it is easy to see that

$$||E_{m,1}||\leq hL\sum_{i=0}^{m-1}||E_{i,1}||+c_1hL||I||+(m+1)||e(G)||,$$

where I is the maximum error of the Lagrange interpolation. Finally, from the Gronwall inequality [2], we conclude that

$$||E_{m,1}|| \leq (c_1 hL||I|| + M||e(G)||) \exp(Lt_m).$$

This yields $||E_{m,1}|| \to 0$ as $h \to 0$ and for the functions k_1, k_2, and g with at least fourth-order continuous derivatives, we have $e(G) = O(h^4)$ and $I = O(h^4)$. So we expect the error generally to be at least $O(h^4)$ as the numerical results confirm it. □

5 Application

5.1 Description of the model

In this part, the introduced numerical method is applied to interpret the avian human influenza epidemic model. Iwami, Takeuchi, and Liu [6] proposed the following mathematical model to interpret the spread of the avian human influenza epidemic

$$\begin{aligned} X' &= \eta - bX - \omega XY, \\ Y' &= \omega XY - (b+m)Y, \\ S' &= \lambda - \mu S - (\beta_1 Y - \beta_2 H)S, \\ B' &= \beta_1 SY - (\mu + d_1 + \epsilon)B, \\ H' &= \beta_2 SH + \epsilon B - (\mu + \alpha + \gamma)H, \\ R' &= \gamma H - \mu R. \end{aligned} \tag{9}$$

All birds and humans in the effective population are divided into six main groups, respectively, including susceptible birds (X), birds infected with the avian influenza (Y), susceptible humans (S), humans infected with the avian influenza (B), humans infected with mutant the avian influenza (H), and humans recovered from mutant the avian influenza (R). It is assumed that all birds infected with the avian influenza are dead or remain infected and can never recover. The parameter η and λ are the birth rates for birds and humans, respectively. The natural death rate is assumed to be b and μ for birds and humans, respectively. Furthermore, m and d are the death rates infected by wild the avian influenza, respectively, for birds and humans, and α is the additional death rate induced by mutant the avian influenza. The parameters ω and β_1 are the rate at which the avian influenza is contracted from an average infected bird, β_2 is the transmission rate of mutant the avian influenza in humans, γ is the recovery rate, and ϵ is the mutation rate.

In [6], it was investigated mathematical properties and qualitative analysis of the above model, also was defined the basic reproductive number for infected birth with the avian influenza ("r_0, which is the number of newly infected birds that are produced from anyone infected bird when all birds are susceptible") and the basic reproductive number for infected humans with

mutant the avian influenza (R_0) as

$$r_0 = \frac{\eta\omega}{b(b+m)}, \qquad R_0 = \frac{\lambda\beta_2}{\mu(\mu+\alpha+\gamma)}.$$

Finally, using some theorems, global analysis to spread of the avian influenza and mutant influenza in the human world is obtained as follows:
"If $r_0 \leq 1$ and $R_0 < 1$, then the avian influenza and mutant the avian influenza cannot spread in the human world. On the other hand, if $R_0 > 1$, then mutant the avian influenza spreads in the human world. Moreover, if $r_0 > 1$, then both the avian influenza and mutant the avian influenza spread in the human world."
It must be pointed that all constants are positive in the system (9) and that ϵ is sufficiently small. Moreover, b is sufficiently larger than μ ($b \gg \mu$), α is less than d, and d is less than m ($m > d > \alpha$) because of the differences of the virulence [6].

5.2 Solution procedure

One can transfer the system (9) to the SVIE

$$\begin{aligned}
X(t) &= X(0) + \eta t - \int_0^t (b + \omega Y(s))X(s)ds,\\
Y(t) &= Y(0) + \int_0^t (\omega X(s) - b - m)Y(s)ds,\\
S(t) &= S(0) + \lambda t - \int_0^t (\mu + \beta_1 Y(s) + \beta_2 H(s))S(s)ds,\\
B(t) &= B(0) + \int_0^t (\beta_1 S(s)Y(s) - (\mu + d + \epsilon)B(s))ds,\\
H(t) &= H(0) + \int_0^t (\beta_2 S(s)H(s) + \epsilon B(s) - (\mu + \alpha + \gamma)H(s))ds,\\
R(t) &= R(0) + \int_0^t (\gamma H(s) - \mu R(s))ds. \qquad (10)
\end{aligned}$$

In this way, we can apply the multistage method for this SVIE. In order to do it, set $t = t_{n,i}$. Therefore

$$\begin{aligned}
X(t_{n,i}) =& X(0) + \eta t_{n,i} - \int_0^{t_n} (b + \omega Y(s))X(s)ds - \int_{t_n}^{t_{n,i}} (b + wY(s))X(s)ds,\\
Y(t_{n,i}) =& Y(0) + \int_0^{t_n} (\omega X(s) - b - m)Y(s)ds + \int_{t_n}^{t_{n,i}} (wX(s) - b - m)Y(s)ds,
\end{aligned}$$

$$
\begin{aligned}
S(t_{n,i}) =& S(0) + \lambda t_{n,i} - \int_0^{t_n} (\mu + \beta_1 Y(s) + \beta_2 H(s)) S(s) ds \\
&- \int_{t_n}^{t_{n,i}} (\mu + \beta_1 Y(s) + \beta_2 H(s)) S(s) ds, \\
B(t_{n,i}) =& B(0) + \int_0^{t_n} (\beta_1 S(s) Y(s) - (\mu + d + \epsilon) B(s)) ds \\
&+ \int_{t_n}^{t_{n,i}} (\beta_1 S(s) Y(s) - (\mu + d + \epsilon) B(s)) ds, \\
H(t_{n,i}) =& H(0) + \int_0^{t_n} (\beta_2 S(s) H(s) + \epsilon B(s) - (\mu + \alpha + \gamma) H(s)) ds \\
&+ \int_{t_n}^{t_{n,i}} (\beta_2 S(s) H(s) + \epsilon B(s) - (\mu + \alpha + \gamma) H(s)) ds, \qquad (11) \\
R(t_{n,i}) =& R(0) + \int_0^{t_n} (\gamma H(s) - \mu R(s)) ds + \int_{t_n}^{t_{n,i}} (\gamma H(s) - \mu R(s)) ds, \\
& n = 0, 1, \ldots, N-1, \ i = 1, 2, 3.
\end{aligned}
$$

Assume that $X_{n,i}, Y_{n,i}, \ldots, R_{n,i}$ are approximation for exact solutions $X(t)$, $Y(t), \ldots, R(t)$ in the point $t_{n,i}$. Similar to the previous section, we can approximate the integrals in the SIVE (11). Use two points Gauss–Legendre for integration on the interval $[0, t_n]$. Thus

$$
\begin{aligned}
&\int_0^{t_n} (b + \omega Y(s)) X(s) ds \approx \frac{h}{2} \sum_{j=0}^{n-1} [(b + \omega Y_{j,1}) X_{j,1} + (b + \omega Y_{j,2}) X_{j,2}] =: A_1, \\
&\int_0^{t_n} (\omega X(s) - b - m) Y(s) ds \\
&\quad \approx \frac{h}{2} \sum_{j=0}^{n-1} [(\omega X_{j,1} - b - m) Y_{j,1} + (\omega X_{j,2} - b - m) Y_{j,2}] =: A_2, \\
&\vdots \\
&\int_0^{t_n} (\gamma H(s) - \mu R(s)) ds \approx \frac{h}{2} \sum_{j=0}^{n-1} [\gamma H_{j,1} - \mu R_{j,1} + \gamma H_{j,2} - \mu R_{j,2}] =: A_6.
\end{aligned}
$$

Continuing the same process for other integrals leads to the system of algebraic equations

$$
\begin{aligned}
X_{n,1} =& X_0 + \eta t_{n,1} - A_1 - \frac{h c_1}{2} \left[(b + \omega \mathcal{P}_2(t_n + \frac{2 - \sqrt{3}}{6} h)) \mathcal{P}_1(t_n + \frac{2 - \sqrt{3}}{6} h) \right. \\
&\left. + (b + \omega \mathcal{P}_2(t_n + \frac{h}{6})) \mathcal{P}_1(t_n + \frac{h}{6}) \right],
\end{aligned}
$$

$$
\begin{aligned}
X_{n,2} =& X_0 + \eta t_{n,2} - A_1 \\
&- \frac{hc_2}{2}\left[(b+\omega\mathcal{P}_2(t_n+\frac{2+\sqrt{3}}{6}h))\mathcal{P}_1(t_n+\frac{2+\sqrt{3}}{6}h)\right. \\
&\qquad \left.+(b+\omega\mathcal{P}_2(t_n+\frac{h}{6}))\mathcal{P}_1(t_n+\frac{h}{6})\right], \\
X_{n,3} =& X_0 + \eta t_{n,3} - A_1 \\
&- \frac{h}{2}\left[(b+\omega Y_{n,1})X_{n,1} + (b+\omega Y_{n,2})X_{n,2}\right], \\
Y_{n,1} =& Y_0 + A_2 \\
&+ \frac{hc_1}{2}\left[(\omega\mathcal{P}_1(t_n+\frac{2-\sqrt{3}}{6}h) - b - m)\mathcal{P}_2(t_n+\frac{2-\sqrt{3}}{6}h)\right. \\
&\qquad \left.+(\omega\mathcal{P}_1(t_n+\frac{h}{6}) - b - m)\mathcal{P}_1(t_n+\frac{h}{6})\right], \\
Y_{n,2} =& Y_0 + A_2 \\
&+ \frac{hc_2}{2}\left[(\omega\mathcal{P}_1(t_n+\frac{2+\sqrt{3}}{6}h) - b - m)\mathcal{P}_2(t_n+\frac{2+\sqrt{3}}{6}h)\right. \\
&\qquad \left.+(\omega\mathcal{P}_1(t_n+\frac{h}{6}) - b - m)\mathcal{P}_1(t_n+\frac{h}{6})\right], \\
Y_{n,3} =& Y_0 + A_2 \\
&- \frac{h}{2}\left[(\omega X_{n,1} - b - m)Y_{n,1} + (\omega X_{n,2})Y_{n,2}\right], \\
&\vdots \\
R_{n,1} =& R_0 + A_6 \\
&+ \frac{hc_1}{2}\left[\gamma\mathcal{P}_5(t_n+\frac{2-\sqrt{3}}{6}h) - \mu\mathcal{P}_6(t_n+\frac{2-\sqrt{3}}{6}h)\right. \\
&\qquad \left.+\gamma\mathcal{P}_5(t_n+\frac{h}{6}) - \mu\mathcal{P}_6(t_n+\frac{h}{6})\right], \\
R_{n,2} =& R_0 + A_6 \\
&+ \frac{hc_2}{2}\left[(\gamma\mathcal{P}_5(t_n+\frac{2+\sqrt{3}}{6}h) - \mu\mathcal{P}_6(t_n+\frac{2+\sqrt{3}}{6}h)\right. \\
&\qquad \left.+\gamma\mathcal{P}_5(t_n+\frac{h}{6}) - \mu\mathcal{P}_6(t_n+\frac{h}{6})\right],
\end{aligned}
$$

$$R_{n,3} = R_0 + A_6 - \frac{h}{2}\left[\gamma H_{n,1} - \mu R_{n,1} + \gamma H_{n,2} - \mu R_{n,2}\right], \qquad n = 0, 1, \ldots, N-1,$$

where

$$X(t_n + hx) \approx \mathcal{P}_1(t_n + hx) = \sum_{j=0}^{3} L_j(t_n + hx) X_{n,j},$$

$$Y(t_n + hx) \approx \mathcal{P}_2(t_n + hx) = \sum_{j=0}^{3} L_j(t_n + hx) Y_{n,j},$$

$$\vdots$$

$$R(t_n + hx) \approx \mathcal{P}_6(t_n + hx) = \sum_{j=0}^{3} L_j(t_n + hx) R_{n,j},$$

for the Lagrange polynomial $L_j(t_n + hx) = \prod_{\substack{m=0 \\ m \neq j}}^{3} \frac{x - c_m}{c_j - c_m}$ and $x = \frac{2 \pm \sqrt{3}}{6}$ or $x = \frac{1}{6}$.
In each step $(n = 0, 1, \ldots, N-1)$, three values of unknowns functions can be obtained by solving this system.

6 Numerical results and discussion

In this section, we present some numerical results to investigate the spread of the avian influenza. The initial value of system (10) is fixed at $X(0) = 10$, $Y(0) = 2$, $S(0) = 100$, $B(0) = 0$, $H(0) = 0$, and $R(0) = 0$. These amounts are chosen from the region $\Omega := \{(X, Y, S, B, H, R) : X > 0, Y > 0, S > 0, B = 0, H = 0, R = 0\}$, which denotes that there do not exist the infected humans with the avian influenza and mutant the avian influenza.
Similar to [6], the parameters are fixed at $b = 5$, $m = 5$, $\lambda = 3$, $\mu = 0.015$, $\beta_1 = 0.2$, $\epsilon = 10^{-3}$, $d = 1$, $\alpha = 0.06$, and $\gamma = 0.01$. We will provide numerical results of infected birds (Y) with the avian influenza, infected humans (B) with the avian influenza, and infected humans (H) with mutant the avian influenza by choosing different constants η, ω, and β_2.
The numerical experiments are carried out in Maple, and nonlinear systems arising from the nonlinear integral equations are solved by Maple routine $fsolve$.

Example 1. In system (10), let $\eta = 26.5$, $\omega = 2$ and $\beta_2 = 0.003$. Then

$$r_0 = \frac{\eta\omega}{b(b+m)} = 1.06, \qquad R_0 = \frac{\beta_2\lambda}{\mu(\mu+\alpha+\gamma)} \approx 7.1.$$

Figures 1 and 2:(a) describe the pandemic effect on the birds and the human world. Figure 1:(a) describes an endemic in the infected birds just after the occurrence of the avian influenza. According to the reported global analysis from [6], Figure 1:(b) shows an initially pandemic of the avian influenza and afterward pandemic level would be decreased, and Figure 2:(a) interprets that the infected humans with mutant the avian influenza outbreak and eventually keep the relatively high level of the size. It suggests that the mutant the avian influenza pandemic will occur if we do not take efficient measures to control the spread of the avian influenza. Moreover, CPU times of the computations are shown in the figures.

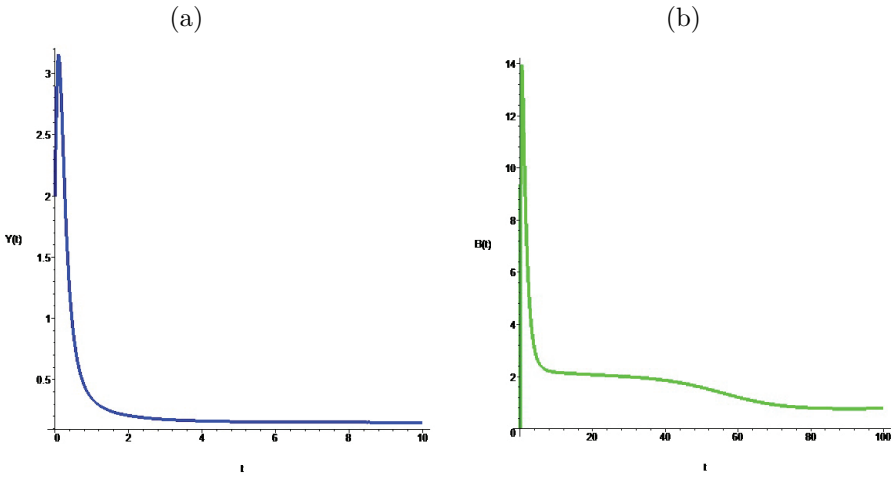


Figure 1: (a) The bird rate infected with avian, $T = 10, M = 100$, CPU time=19.5s. (b) The human rate infected with the avian influenza, $T = 100, M = 1000$, CPU time=408s. $r_0 = 1.06, R_0 \approx 7.1$.

Example 2. The parameters are the same as the previous example except for $\beta_2 = 0.0015$ (the transmission rate of mutant the avian influenza is reduced by half). Therefore the basic reproduction number for the avian influenza in the bird and human world, respectively, is equal to $r_0 = \frac{\eta\omega}{b(b+m)} = 1.06$, $R_0 = \frac{\beta_2\lambda}{\mu(\mu+\alpha+\gamma)} \approx 3.5$.

Figures 2:(b) and 3:(a), respectively, show the endemic process in the bird world after the morbidity of the avian influenza and the occurrence of the pandemic of the avian influenza in the human world. Figure 3:(b) shows the human rate infected with mutant the avian influenza. Since R_0 is relatively small, we do not have mutant the avian influenza pandemic and in this case, we have only the avian influenza pandemic.

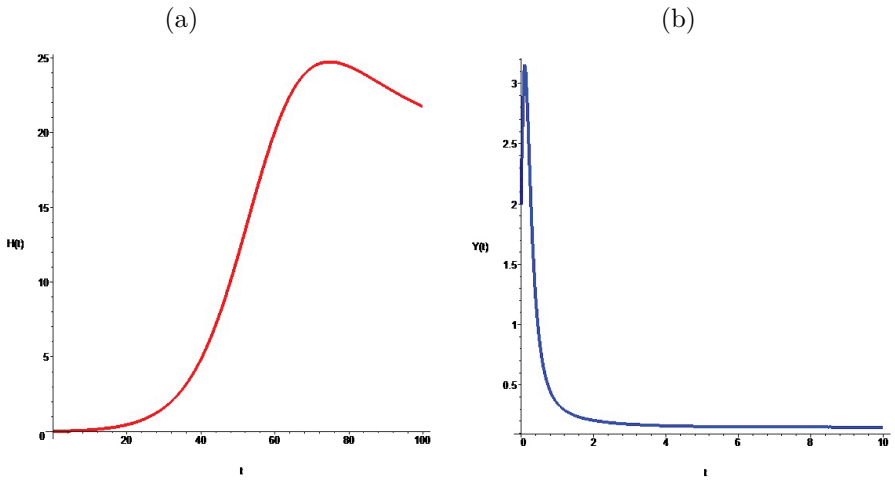


Figure 2: (a) The rate infected with mutant the avian influenza, $T = 100, M = 1000$, CPU time=408s, $r_0 = 1.06, R_0 \approx 7.1$. (b) The bird rate infected with the avian influenza, $T = 10, M = 100$, CPU time=19.3s, $r_0 = 1.06, R_0 \approx 3.5$.

Example 3. Let $\eta = 30$ and $\omega = 1.5$; then $r_0 = 0.9$ and $R_0 \approx 7.1$.

The numerical results for this example are shown in Figures 4 and 5. In these figures, there are two pandemic for the avian influenza and mutant the avian influenza even, when all the infected birds and humans with the avian influenza were extinct.

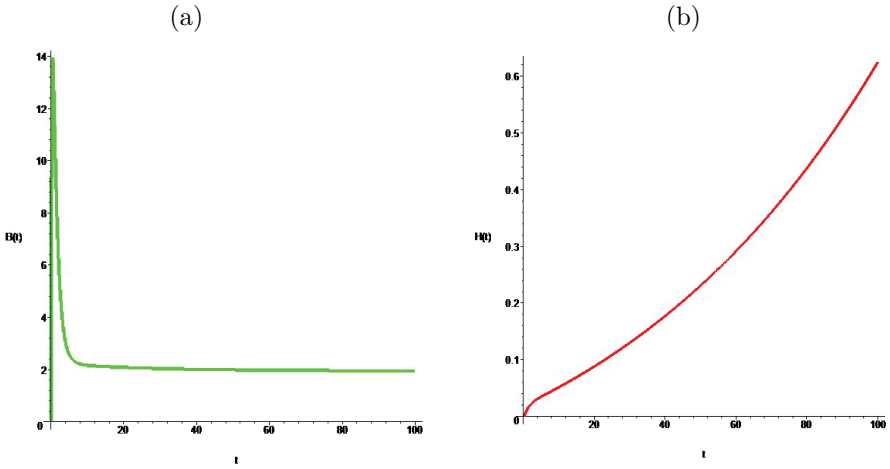


Figure 3: (a) The human rate infected with the avian influenza, (b) The rate infected with mutant the avian influenza. $T = 100, M = 1000$, CPU time=398s. $r_0 = 1.06, R_0 \approx 3.5$.

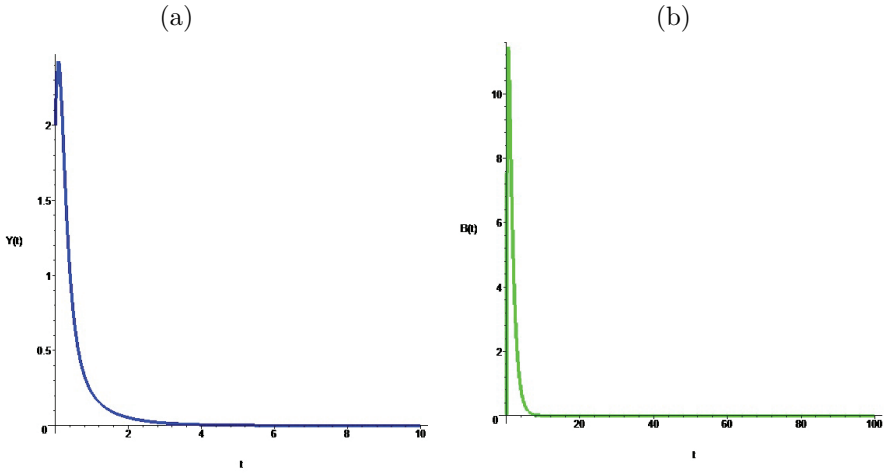


Figure 4: (a) The bird rate infected with the avian influenza, $T = 10, M = 100$, CPU time=18.2s. (b) The human rate infected with the avian influenza, $T = 100, M = 1000$, CPU time=409s. $r_0 = 0.9, R_0 \approx 7.1$.

From these numerical results similar to [6], we conclude the following result:

- Elimination policy of the infected birds is not necessarily useful, because the speed of spread of disease in the bird world is fast and their lifetime is shorter than humans.

- For $r_0 < 1$, the infected humans extinct, but there are some infected humans for $r_0 > 1$ (see Figures 1:(a), 3:(a), 4:(b)).

- If R_0 is not large enough, then the outbreak of mutant influenza will not occur (see Figures 2:(a), 3:(b), 5).

The numerical results for Examples 1–3 are comparable with the fourth-order Runge–Kutta method [6] in the whole intervals. Moreover, the presented method, in contrast to Runge–Kutta methods, did not need to computation a lot of derivations, and the computation time for this method is short. Of course, we can increase the convergence order of the method. Also, numerical results are obtained for each time interval that is one of the other advantages of the introduced method.

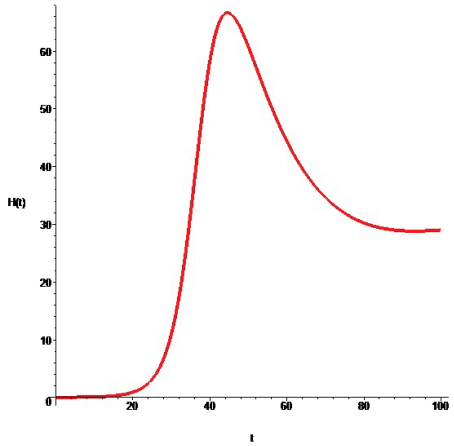


Figure 5: The rate infected with mutant the avian influenza, $T = 100, M = 1000$, CPU time=409s, $r_0 = 0.9, R_0 \approx 7.1$.

References

1. Agarwal, R.P., O'Regan, D., Tisdell, C. and Wong, P.J.Y. *Constant-sign solutions of a system of Volterra integral equations,* Comput. Math. Appl. 54 (1) (2007) 58–75.

2. Brunner, H. and van der Houwen, P.J. *The numerical solution of Volterra equations*, Amsterdam etc., North-Holland 1986.

3. Caliò, F., Garralda-Guillem, A.I., Marchetti, E. and Ruiz Galán, M. *Numerical approaches for systems of Volterra-Fredholm integral equations,* Appl. Math. Comput. 225 (2013) 811–821.

4. Gautschi, W. *Orthogonal polynomial, computation and approximation*, Oxford science publications 2004.

5. Iwami, S., Takeuchi, Y., Korobeinikov, A. and Liu, X. *Prevention of the avian influenza epidemic: What policy should we choose?*, J. Theor. Biol. 252 (4) (2008) 732–741.

6. Iwami, S., Takeuchi, Y. and Liu, X. *The avian-human influenza epidemic model*, Math. Biosci. 207 (1) (2007) 1–25.

7. Katani, R. and Shahmorad, S. *A block by block method with Romberg quadrature for the system of Urysohn type Volterra integral equations,* Comput. Appl. Math. 31 (1) (2012) 1–13.

8. Koekoek, R., Lesky, P. and Swarttouw, R. *Hypergeometric orthogonal polynomials and their q-analogues,* Springer 2010.

9. Linz, P. *Analytical and numerical methods for Volterra equations*, SIAM Philadelphia 1985.

10. Maleknejad, K. and Ostadi, A. *Numerical solution of system of Volterra integral equations with weakly singular kernels and its convergence analysis*, Appl. Numer. Math. 115 (2017) 82–98.

11. Poland, G.A., Jacobson, R.M. and Targonski, P.V. *The avian and pandemic influenza: an overview*, Vaccine 25 (2007) 3057–3061.

12. Shinya, K., Ebina, M., Yamada, S., Ono, M., Kasai, N. and Kawaoka, Y., *The avian flu: Influenza virus receptors in the human airway,* Nature 440 (2006) 435–436.

13. Trefethen, L.N. *Approximation theory and approximation practice* (Applied Mathematics), society for industrial and applied mathematics 2013.

How to cite this article
Katani, R. Numerical solution of a system of Volterra integral equations in application to avian-human influenza epidemic model. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 37-53. doi: 10.22067/ijnao.2021.69416.1021

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 55-72
DOI:10.22067/ijnao.2021.70731.1038
https://ijnao.um.ac.ir/

Research Article

Asymptotic and numerical methods for solving singularly perturbed differential difference equations with mixed shifts

S. Priyadarshana, S.R. Sahu and J. Mohapatra*

Abstract

This article deals with an efficient approximation method named successive complementary expansion method (SCEM) for solving singularly perturbed differential-difference equations with mixed shifts. It is compared with the method of matched asymptotic expansion (MMAE) and the parameter uniform upwind finite difference scheme for solving such a model. The comparison shows, unlike the MMAE, the SCEM method requires no matching procedure. It requires less computation when compared to the upwind finite difference scheme on the Shishkin mesh. The error analysis is carried out to prove the robustness of the method. Some numerical experiments are provided, which show the effectiveness of the proposed method.

AMS subject classifications (2020): 65L10; 65L11.

Keywords: Singular perturbation; Mixed shifts; Asymptotic expansion; MMAE; SCEM; Upwind scheme.

---

*Corresponding author
Received 31 May 2021; revised 21 July 2021; accepted 3 August 2021
Sushree Priyadarshana
Department of Mathematics, National Institute of Technology Rourkela-769008, Odisha, India. e-mail: 519ma2009@nitrkl.ac.in

Subal Ranjan Sahu
Department of Mathematics, The ICFAI University Tripura-799210, Tripura, India. e-mail: subalranjan1@gmail.com

Jugal Mohapatra
Department of Mathematics, National Institute of Technology Rourkela-769008, Odisha, India. e-mail: jugal@nitrkl.ac.in

1 Introduction

It is noteworthy to observe the wide use of differential equations involving small parameters in a variety of disciplines from physics, chemistry, economics to biology. A differential equation, in general, is said to be singularly perturbed if its higher-order derivative gets multiplied with a small parameter ε $(0 < \varepsilon \ll 1)$ known as the perturbation parameter. In many mathematical models, these types of equations contain some small delay and small advance terms, which make them singularly perturbed differential-difference equations (SPDDE). Due to the presence of the perturbation parameter, these problems exhibit boundary or interior layers. The development in solutions to these problems both analytically and numerically is not a settled direction in mathematics. The massive use of these equations in many fields of science and engineering leads many researchers to solve such problems more efficiently. These equations usually arise in control theory [12], generation of action potential in nerve cells by random synaptic inputs in dendrites, modeling of activation of a neuron, and many more [25, 27].

Over the last few decades, many researches have been done on singularly perturbed differential equations. One may refer to [1, 6, 5, 13, 14, 24] and the references therein for different kinds of numerical methods. To mention a few, Kadalbajoo and Sharma [10, 9] surveyed various asymptotic and numerical methods for the solution of such problems. Lange and Miura [15, 16] gave an asymptotic approach to solve SPDDE along with the turning point problems. Sirisha, Phaneendra, and Reddy [26] used the idea of domain decomposition and introduced the mixed finite difference method. Using post-processing and grid distribution techniques, Mohapatra and Natesan [19, 20, 21] formulated uniformly convergent numerical methods. Duressa and Reddy [8] constructed a domain decomposition method and used a terminal boundary point to the domain for decomposing it into two regions. Melesse, Tiruneh, and Derese [18] used an initial value method for solving these problems. A collocation method was proposed to solve these boundary value problems using a modified B-spline basis function by Arora and Kaur [2]. Rao and Chakravarthy [23] used an exponentially fitted tridiagonal finite difference method. Recently, Cengizci, Natesan, and Atay [4] used an asymptotic numerical hybrid method for the singularly perturbed system of two-point reaction-diffusion boundary-value problems. Mushahary, Sahu, and Mohapatra [22] used a finite difference scheme for solving SPDDE.

Our main objective in this paper is to prove a successive complementary expansion method (SCEM) as an efficient alternative to the method of matched asymptotic expansion (MMAE) for solving SPDDE. This method, in general, applies to all the singularly perturbed problems that can be approximated by the MMAE. The main idea of SCEM is to use a correction term for the approximation of the inner layer to complement the solution for the entire domain. The quality of the approximation is improved by iteratively using corrections to the new terms for the inner layer. The ability to

provide a uniformly valid approximation without any matching procedure is the main advantage of using SCEM over MMAE.

In this work, some SPDDEs are taken into consideration, which exhibit boundary layer behavior at the left-end of the interval. These problems contain mixed shifts, that is, both the delay and the advance terms. The complete description of the problem is given in section 2. At first, the model is approximated asymptotically by the MMAE, and then we have used an efficient asymptotic method as an alternative of widely used MMAE, that is, the SCEM. This method was first introduced by Cousteix and Mauss in [7, 17]. We have also approximated these problems by an upwind finite difference scheme on the Shishkin mesh. The complete working principle of the parameter uniform numerical scheme, MMAE, and SCEM, along with the use of SCEM over MMAE, has been described in section 3. In the support of the above-said methods, some numerical examples with their results are discussed in section 4. The complete discussion with the conclusion is given in the final section.

2 Description of the problem

Consider the following SPDDE with mixed shifts terms:

$$\begin{cases} \varepsilon y''(x) + a(x)y'(x) + b(x)y(x-\delta) \\ \quad +c(x)y(x) + d(x)y(x+\eta) = f(x), & 0 < x < 1, \\ y(x) = \phi(x), & on \ -\delta \le x \le 0, \\ y(x) = \gamma(x), & on \ 1 \le x \le 1+\eta, \end{cases} \tag{1}$$

where $a(x), b(x), c(x), d(x), f(x), \phi(x), \gamma(x)$ are bounded and continuously differentiable functions on $(0, 1)$. Here, ε is the singular perturbation parameter with $0 < \varepsilon \ll 1$. The parameters δ and η are the delay and advance terms, respectively, with $0 < \delta = o(\varepsilon); 0 < \eta = o(\varepsilon)$. Now for the terms of (1) containing the delay and the advance operators, we shall use the Taylor's series approximation in the neighborhood of the point x, that is,

$$\begin{cases} y(x-\delta) \approx y(x) - \delta y'(x), \\ y(x+\eta) \approx y(x) + \eta y'(x). \end{cases} \tag{2}$$

The application of (2) converts (1) to the following form:

$$\begin{cases} \varepsilon y''(x) + \alpha(x)y'(x) + \beta(x)y(x) = f(x), & 0 < x < 1, \\ y(0) = \phi(0), \qquad y(1) = \gamma(1), \end{cases} \tag{3}$$

where $\alpha(x) = a(x) - \delta b(x) + \eta d(x)$ and $\beta(x) = b(x) + c(x) + d(x)$. Assume $\alpha(x) \geq \lambda > 0$ and $\beta(x) > 0$ for some constant λ. Since the parameters δ and η are of $o(\varepsilon)$, so the use of Taylor's series approximation and the conversion of (1) to (3) is valid.

3 Description of the methods

3.1 Asymptotic approximation

Any two functions $\psi(x, \varepsilon)$ and $\xi(x, \varepsilon)$ are said to be asymptotically identical to $\rho(\varepsilon)$ if their difference is less than $\rho(\varepsilon)$ in the domain Ω,

$$\psi(x, \varepsilon) - \xi(x, \varepsilon) = o(\rho(\varepsilon)). \tag{4}$$

Here, $\rho(\varepsilon)$ is an order function and $\xi(x, \varepsilon)$ is the asymptotic approximation of $\psi(x, \varepsilon)$. The general form of asymptotic expansion is

$$\psi(x, \varepsilon) = \sum_{i=0}^{n} \delta_i(\varepsilon)\psi_i(x, \varepsilon), \tag{5}$$

where $\delta_i(\varepsilon)$, $i = 0, 1, \ldots$, is an asymptotic sequence and $\psi_i(x, \varepsilon)$ is of $o(\rho(\varepsilon))$. If $\delta_i(\varepsilon)$ satisfies $\delta_{i+1}(\varepsilon) = o(\delta_i(\varepsilon))$, then the expansion is called generalized asymptotic expansion. If the expansion is of the form

$$\psi(x, \varepsilon) = E_0\psi = \sum_{i=0}^{n} \delta_i^{(0)}(\varepsilon)\psi_i^{(0)}(x, \varepsilon), \tag{6}$$

then it is said to be regular asymptotic expansion. Here, E_0 is an outer expansion operator. So, we conclude $\psi(x, \varepsilon) - E_0\psi(x, \varepsilon) = o(\rho(\varepsilon))$.

3.2 Method of matched asymptotic expansion (MMAE)

In many cases, the functions are not regular in the domain of consideration Ω. So, (6) is only valid upon one finite part of the region Ω. Let that restricted region be named $\Omega_0 \subset \Omega$. Now, the other part of the region (inner region), that is, Ω_1 is located near the boundary layer where the solution undergoes a rapid change to satisfy the boundary conditions. For the inner region, we need to introduce a new variable(stretching variable or boundary layer variable) $\overline{x} = \dfrac{x - x_0}{\xi(\varepsilon)}$. Here, x_0 is the point where the rapid change of the solution occurs and $\xi(\varepsilon)$ is the order of the thickness of the boundary layer.

Now, similarly as in (6), we can construct an asymptotic expansion for the function in Ω_1,

$$\psi(x,\varepsilon) = E_1\psi = \sum_{i=0}^{n} \delta_i^{(1)}(\varepsilon)\psi_i^{(1)}(x,\varepsilon). \tag{7}$$

Both the expansion operators E_0 and E_1 have the same order, that is, $o(\rho(\varepsilon))$. So, we have $\psi(x,\varepsilon) - E_1\psi(\overline{x},\varepsilon) = o(\rho(\varepsilon))$. The valid approximation of the solution thus can be written as

$$\psi_{mmae}(x,\varepsilon) = E_0\psi(x,\varepsilon) + E_1\psi(\overline{x},\varepsilon) - E_0E_1\psi(x,\varepsilon). \tag{8}$$

In MMAE, we calculate two different results separately with two different variables for both the inner and outer layers. Then, to obtain the uniformly valid approximation for the entire domain, we need to subtract the region where both the approximations overlap. For this purpose, we need to develop a matching procedure by the limiting process [3].

3.3 Successive complementary expansion method (SCEM)

As the functions that we consider are not regular in the entire domain Ω, after calculating the solution for the outer region Ω_0 through the expansion E_0 as in (6), we need to extend the solution through a complementary function for the entire domain. This complementary function depends upon the variable $\overline{x}$. The general form of the asymptotic expansion in SCEM can be formulated as

$$\psi_{scem}(x,\overline{x},\varepsilon) = \sum_{i=0}^{n} \delta_i(\varepsilon)(\psi_i^{out}(x,\varepsilon) + \psi_i^{in}(\overline{x},\varepsilon)). \tag{9}$$

Here, $\psi_i^{out}(x,\varepsilon)$ is the same as the outer region solution that has been obtained by MMAE before. The outer solution depends upon x but not upon ε but the complementary function depends both upon x and ε. As in the MMAE method, SCEM does not require any matching procedure.

3.4 Finite difference scheme

In this section, we first construct a piecewise uniform Shishkin mesh. The Shishkin mesh is distinguished from other piecewise uniform meshes in terms of its choice of the transition parameter τ defined as $\tau = \min\left(\frac{1}{2}, \frac{\varepsilon}{\alpha}\ln N\right)$.

Now, we divide each of the intervals $[0,\tau]$ and $[\tau,1]$ into $N/2$ equal sub-intervals. The theory of Shishkin mesh demands as $N\to\infty$ the number of sub-intervals in each interval should be bounded below.

$$\Omega^N=\begin{cases} x(i)=h_i, & i\le N/2,\\ x(i)=H_i, & i>N/2,\end{cases} \tag{10}$$

The coarse part of Shishkin mesh has spacing $H=\dfrac{2(1-\tau)}{N}$ and the fine part has spacing $h=\dfrac{2\tau}{N}$. Now, let us denote $h_i=x_i-x_{i-1}$. The forward and central difference operators are denoted as

$$D^+=\frac{Y_{i+1}-Y_i}{h_{i+1}},\quad D^+D^-=\frac{2}{h_i+h_{i+1}}\left[\frac{Y_{i+1}-Y_i}{h_{i+1}}-\frac{Y_i-Y_{i-1}}{h_i}\right].$$

Now (3) takes the form

$$\begin{aligned}\varepsilon D^+D^-Y_i+\alpha(x)D^+Y_i-\beta(x)Y_i=f(x),\\ y(0)=Y(0),\qquad y(1)=Y(1),\end{aligned} \tag{11}$$

for any mesh function $Y(x_i)=Y_i$. So, for the domain, in (10), we get the following system of equations:

$$P_i^-Y_{i-1}-P_i^0Y_i+P_i^+Y_{i+1}=P_i^N, \tag{12}$$

where

$$\begin{aligned}P_i^-&=\frac{2\varepsilon}{h_i(h_i+h_{i+1})},\\ P_i^0&=\frac{2\varepsilon}{h_i(h_i+h_{i+1})}+\frac{2\varepsilon}{h_{i+1}(h_i+h_{i+1})}+\frac{\alpha(x)}{h_{i+1}}+\beta(x_i),\\ P_i^-&=\frac{2\varepsilon}{h_{i+1}(h_i+h_{i+1})}+\frac{\alpha(x)}{h_{i+1}}.\end{aligned} \tag{13}$$

The matrix associated here is an M-matrix. For detailed analysis of the proposed scheme one can refer [11].

4 Results and discussion

Even though much work has been done by the MMAE method, there is no specific theory that can describe the method in a proper way. So, we need to discuss the working principle of both MMAE and SCEM by some illustrative examples. The proposed SCEM is found efficient in solving examples of the type (1), which possess the left-end boundary layer. The approximated exact

solution of SPDDE of the form (1) with constant coefficients is given by

$$y(x) = C_1 e^{M_1 x} + C_2 e^{M_2 x}, \tag{14}$$

where

$$C_1 = \frac{1 - e^{M2}}{e^{M1} - e^{M2}}, \qquad C_2 = 1 - C_1,$$

$$M_1 = \frac{-\alpha(x) + \sqrt{\alpha(x)^2 - 4\beta(x)}}{2\varepsilon}, \quad M_2 = \frac{-\alpha(x) - \sqrt{\alpha(x)^2 - 4\beta(x)}}{2\varepsilon}.$$

Example 1. Consider the following problem with a delay term:

$$\begin{cases} \varepsilon y''(x) + y'(x) + 2y(x-\delta) - 3y(x) = 0, \\ y(x) = 1, \quad -\delta \le x \le 0, \quad y(1) = 1. \end{cases}$$

The solution of this problem possesses a rapid change near the point $x = 0$ when $\varepsilon \to 0^+$. This region is said to be the boundary layer or the inner layer. So, this possesses a left-end boundary layer. The other region, which is far away from the boundary layer, is said to be the outer region. In case of the outer region, there is no record of any unusual behavior of the solution. We will solve this problem employing MMAE and SCEM and the proposed numerical scheme. For the given example using Taylor's series expansion of the form (2) in the neighborhood of x, we have

$$\varepsilon^* y''(x) + y'(x) - \frac{1}{1-2\delta} y(x) = 0, \qquad where \;\; \varepsilon^* = \frac{\varepsilon}{1-2\delta}. \tag{15}$$

Now, the equation has converted to a second order singularly perturbed problem, which can be solved by both MMAE and SCEM.

4.1 For MMAE

In MMAE, for the outer region solution, which is far away from $x = 0$, we take $x = 1$. Now, the asymptotic approximation for the outer region can be written as

$$y(x) \approx y_0(x) + \varepsilon^* y_1(x) + (\varepsilon^*)^2 y_2(x) + \cdots . \tag{16}$$

Substituting (16) in (15), we reach at

$$\varepsilon^*[y_0''(x) + \varepsilon^* y_1''(x) + (\varepsilon^*)^2 y_2''(x) + \cdots] + [y_0'(x) + \varepsilon^* y_1'(x) + \tag{17}$$
$$(\varepsilon^*)^2 y_2'(x) + \cdots] - \left(\frac{1}{1-2\delta}\right)[y_0(x) + \varepsilon^* y_1(x) + (\varepsilon^*)^2 y_2(x) + \cdots] = 0.$$

For $\varepsilon = 0$, model (17) is reduced to

$$y_0'(x) - \left(\frac{1}{1-2\delta}\right) y_0(x) = 0. \tag{18}$$

The solution of (18) is easily found to be $y_0(x) = A \exp\left(\frac{x}{1-2\delta}\right)$. Here, $A \in \mathbf{R}$ and imposing the outer boundary layer condition (at $x = 1$) we get our required outer solution as

$$y_0(x) = \exp\left(\frac{x-1}{1-2\delta}\right). \tag{19}$$

To obtain the solution for the inner layer, we need to introduce the stretching variable $\overline{x} = \frac{x}{\varepsilon}$. The solution obtained in the inner region that depends upon the variable $\overline{x}$, is denoted by $Y(\overline{x})$. By the use of this stretching variable, we are able to stretch the thin layer near the boundary. Now by applying the chain rule, we get

$$\frac{d}{dx} = \frac{1}{\varepsilon}\frac{d}{d\overline{x}}. \tag{20}$$

Using (20) in (15), we have

$$(\varepsilon^*)^{-1}\frac{d^2Y(\overline{x})}{d\overline{x}^2} + (\varepsilon^*)^{-1}\frac{dY(\overline{x})}{d\overline{x}} - \frac{1}{(1-2\delta)}Y(\overline{x}) = 0. \tag{21}$$

Now, multiplying (21) with ε^*, we have

$$\frac{d^2Y(\overline{x})}{d\overline{x}^2} + \frac{dY(\overline{x})}{d\overline{x}} - \frac{\varepsilon^*}{(1-2\delta)}Y(\overline{x}) = 0. \tag{22}$$

Equation (22) is a regularly perturbed differential equation, and to obtain the solution for the inner region, we need to use the asymptotic approximation of the form

$$Y(\overline{x}) \approx Y_0(\overline{x}) + \varepsilon^* Y_1(\overline{x}) + (\varepsilon^*)^2 Y_2(\overline{x}) + \cdots. \tag{23}$$

We are only doing the expansion for the first term. So, we can make our calculations for $\varepsilon = 0$. Now, we have

$$Y_0''(\overline{x}) + Y_0'(\overline{x}) = 0. \tag{24}$$

On solving (24) and using the boundary condition for the inner layer (at $x = 0$), we arrive at

$$Y_0(\overline{x}) = 1 + C(e^{-\overline{x}} - 1). \tag{25}$$

Clearly, there are no other conditions, so that we can obtain the value of the constant C. Here is the situation where the matching procedure of MMAE works. We have already found out two different solutions, that is, (19) and (25) for two different regions, but, it is obvious that they belong to the same

approximation. Followed by this matching idea [3], we conclude

$$\lim_{x\to 0} y_0(x) = \lim_{\overline{x}\to\infty} Y_0(\overline{x}). \tag{26}$$

Thus, we get $C = 1 - \exp\left(\frac{1}{2\delta - 1}\right)$. So, the solution of our inner region becomes

$$Y_0(\overline{x}) = \exp\left(-\overline{x}\right) - \exp\left(\frac{1}{2\delta-1} - \overline{x}\right) + \exp\left(\frac{1}{2\delta-1}\right). \tag{27}$$

As we have calculated the solutions for both the outer and inner regions, now we can find out the composite solution to our equation. For this, we need to add the solutions of both the regions and subtract the common limit between them. As per (8), we have

$$\begin{aligned} y &\approx y_0(x) + Y_0(\overline{x}) - y_0(0^+), \\ y &\approx y_0(x) + Y_0(\overline{x}) - Y_0(\infty). \end{aligned} \tag{28}$$

Using (28), we obtain the composite MMAE solution as

$$y_{mmae} \approx \exp\left(\frac{x-1}{1-2\delta}\right) + \exp\left(\frac{-x}{\varepsilon}\right) - \exp\left(\frac{1}{2\delta-1}\right) - \frac{x}{\varepsilon}. \tag{29}$$

4.2 For SCEM

The uniformly valid composite expansion in general form of SCEM can be written as

$$y_{scem}(x, \overline{x}, \varepsilon) = \sum_{i=0}^{n} \delta_i(\varepsilon)[y_i(x) + Y_i(\overline{x})]. \tag{30}$$

As we are only interested in solving our problem for $x = 0$, so we get

$$y_{0scem}(x, \overline{x}, \varepsilon) = [y_0(x) + Y_0(\overline{x})]. \tag{31}$$

Here, $y_0(x)$ is the same outer region solution that has been obtained for MMAE in (20). The outer region solution remains the same as the solution here showing no unusual behavior in accordance with the boundary conditions. Using (20) in (31), we get

$$y_{scem} = \exp\left(\frac{x-1}{1-2\delta}\right) + Y_0(\overline{x}). \tag{32}$$

Substituting value of y_{scem} in (15), we have

$$\varepsilon^* y''_{0scem}(x,\overline{x},\varepsilon) + y'_{0scem}(x,\overline{x},\varepsilon) - \frac{1}{1-2\delta} y_{0scem}(x,\overline{x},\varepsilon) = 0. \tag{33}$$

This follows

$$\varepsilon^* \frac{d^2}{dx^2}\left[\exp\left(\frac{x-1}{1-2\delta}\right) + Y_0(\overline{x})\right] + \frac{d}{dx}\left[\exp\left(\frac{x-1}{1-2\delta}\right) + Y_0(\overline{x})\right] - \frac{1}{1-2\delta}\left[\exp\left(\frac{x-1}{1-2\delta}\right) + Y_0(\overline{x})\right] = 0. \tag{34}$$

Using the chain rule for the solution of $Y_0(\overline{x})$ in terms of x, we have

$$\varepsilon^*\left[\left(\frac{1}{1-2\delta}\right)^2 \exp\left(\frac{x-1}{1-2\delta}\right) + \frac{1}{(\varepsilon^*)^2} Y_0''(\overline{x})\right] + \left[\left(\frac{1}{1-2\delta}\right) \exp\left(\frac{x-1}{1-2\delta}\right) + \frac{1}{\varepsilon^*} Y_0'(\overline{x})\right] - \frac{1}{1-2\delta}\left[\exp\left(\frac{x-1}{1-2\delta}\right) + Y_0(\overline{x})\right] = 0. \tag{35}$$

Multiplying ε^* in (35), the resulting equation will be

$$\varepsilon^{*2}\left[\frac{1}{1-2\delta}\right]^2 \exp\left(\frac{x-1}{1-2\delta}\right) + Y_0''(\overline{x}) + \varepsilon^*\left[\frac{1}{1-2\delta}\right] \exp\left(\frac{x-1}{1-2\delta}\right) + Y_0'(\overline{x}) - \varepsilon^*\left[\frac{1}{1-2\delta}\right]\left[\exp\left(\frac{x-1}{1-2\delta}\right) + Y_0(\overline{x})\right] = 0, \tag{36}$$

which is regularly perturbed ordinary differential equation, and for $\varepsilon^* = 0$, it will be reduced to

$$Y_0''(\overline{x}) + Y_0'(\overline{x}) = 0. \tag{37}$$

Now, the solution of the above equation is given by

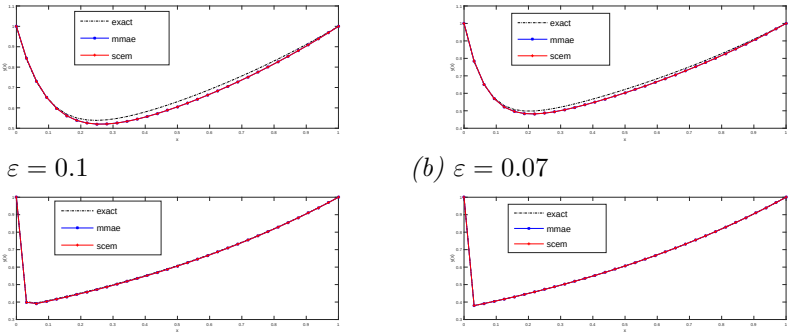
$$Y_0(\overline{x}) = C_1 + C_2 e^{-\overline{x}}. \tag{38}$$

For a uniformly valid SCEM solution, y_{0scem} must satisfy the following boundary conditions:

$$x = 0 \Rightarrow \overline{x} = \frac{x}{\varepsilon} = 0 \Rightarrow y_{0scem}(0,0,\varepsilon) = y_0(0,\varepsilon) + Y_0(0,0,\varepsilon) = 1 \Rightarrow Y_0(0,0,\varepsilon) = 1 - y_0(0,\varepsilon), \tag{39}$$

$$x = 1 \Rightarrow \overline{x} = \frac{x}{\varepsilon} = 1 \Rightarrow y_{0scem}(1,\frac{1}{\varepsilon},\varepsilon) = y_0(1,\varepsilon) + Y_0(1,\frac{1}{\varepsilon},\varepsilon) = 1 \Rightarrow Y_0(1,\frac{1}{\varepsilon},\varepsilon) = 1 - y_0(1,\varepsilon).$$

Using the boundary conditions in (38), we get



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.07$

(c) $\varepsilon = 0.009$ (d) $\varepsilon = 0.0004$

Figure 1: Comparison of solutions with different values of ε for Example 1.

$$C_2 = \frac{\exp\left(\frac{-1}{1-2\delta}\right) - 1}{\exp\left(\frac{-1}{\varepsilon^*}\right) - 1}, \qquad C_1 = -C_2 \exp\left(\frac{-1}{\varepsilon^*}\right). \tag{40}$$

Now, using (40) and (39) in (32), we get the composite SCEM solution as

$$\begin{aligned} y_{scem} = &\exp\left(\frac{x-1}{1-2\delta}\right) - \left[\frac{\exp(\frac{-1}{1-2\delta}) - 1}{\exp(\frac{-1}{\varepsilon^*}) - 1}\right] \exp\left(\frac{-1}{\varepsilon^*}\right) \\ &+ \left[\frac{\exp(\frac{-1}{1-2\delta}) - 1}{\exp(\frac{-1}{\varepsilon^*}) - 1}\right] \exp\left(\frac{-x}{\varepsilon^*}\right). \end{aligned} \tag{41}$$

After getting the approximation to the exact solution by both SCEM and MMAE, it is quite obvious that SCEM is more adaptable. The immense advantage of SCEM is that it gives the uniformly valid approximation to the exact solution without any matching procedure. The error in comparison with the approximated exact solution for both SCEM and MMAE is given in Table 1 for different values of ε. The result is proved to be valid for any value of N. Here for the computational purposes, the nodal points and the values of η and δ are kept fixed to $N = 32$ and $\eta = 0.1\varepsilon = \delta$.

Example 2. Consider the following problem having an advance term:

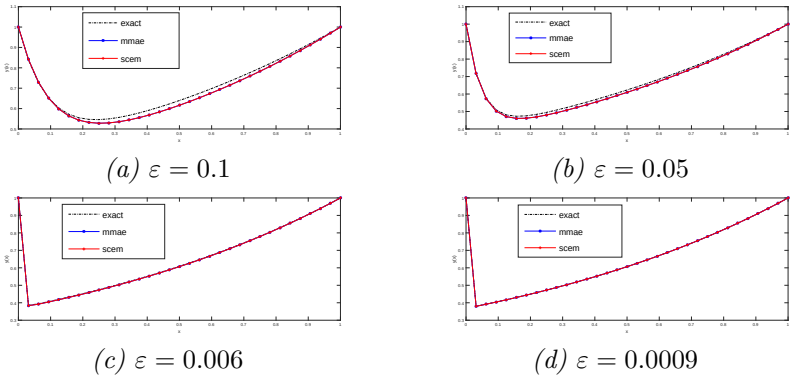
$$\begin{cases} \varepsilon y''(x) + y'(x) - 3y(x) + 2y(x+\eta) = 0, \\ y(0) = 1, \qquad y(x) = 1, 1 \le x \le 1+\eta. \end{cases}$$

This problem has a rapid change near $x = 0$. Now using Taylor's Series expansion in the neighborhood of x, the above problem reduces to

$$\varepsilon^* y''(x) + y'(x) - \frac{1}{1+2\eta} y(x) = 0, \qquad where\ \varepsilon^* = \frac{\varepsilon}{1+2\eta}. \tag{42}$$

ε	$L^\infty error\, in\, MMAE$	$L^\infty error\, in\, SCEM$	Error in Upwind
0.0001	0.00003676	0.00003676	0.04450943
0.0005	0.00018374	0.00018374	0.04428418
0.0010	0.00036728	0.00036728	0.04401059
0.0050	0.00182533	0.00182533	0.04209958
0.0100	0.00361593	0.00361593	0.04025378
0.0500	0.01598307	0.01598307	0.03472166
0.1000	0.02552850	0.02556347	0.03481582
0.3000	0.02853413	0.03217078	0.03427254
0.4000	0.06644741	0.03539967	0.03541439
0.6000	0.15664433	0.04561360	0.03723168
0.8000	0.24352992	0.05726515	0.03825982
1.0000	0.32059406	0.06915099	0.03877738

Table 1: Numerical results with different values of ε for Example 1.



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.05$

(c) $\varepsilon = 0.006$ (d) $\varepsilon = 0.0009$

Figure 2: Comparison of solutions with different values of ε for Example 2.

The one-term SCEM and MMAE approximations for (42) are given as follows:

$$y_{mmae} = \left[1 - \exp\left(\frac{-1}{1+2\eta}\right)\right] \exp\left(\frac{-x}{\varepsilon^*}\right) + \exp\left(\frac{x-1}{1+2\eta}\right),$$

$$y_{scem} = -\left[\frac{\exp(\frac{-1}{1+2\eta}) - 1}{\exp(\frac{-1}{\varepsilon^*}) - 1}\right] \exp\left(\frac{-1}{\varepsilon^*}\right) + \left[\frac{\exp(\frac{-1}{1+2\eta}) - 1}{\exp(\frac{-1}{\varepsilon^*}) - 1}\right] \exp\left(\frac{-x}{\varepsilon^*}\right) + \exp\left(\frac{x-1}{1+2\eta}\right).$$

Example 3. Consider the following problem having mixed shifts:

ε	$L^\infty error\, in\, MMAE$	$L^\infty error\, in\, SCEM$	Error in Upwind
0.0001	0.00003676	0.00003676	0.04448748
0.0005	0.00018367	0.00018367	0.04417452
0.0010	0.00036699	0.00036699	0.04379147
0.0050	0.00181786	0.00181786	0.04101098
0.0100	0.00358632	0.00358632	0.03808839
0.0500	0.01533655	0.01533655	0.02519028
0.1000	0.02377291	0.02379587	0.01688285
0.3000	0.01783685	0.02658575	0.01728136
0.4000	0.04058107	0.02660582	0.02334674
0.6000	0.09131635	0.02829255	0.03160623
0.8000	0.13551419	0.03009785	0.03671486
1.0000	0.17029574	0.03133850	0.04028387

Table 2: Numerical results with different values of ε for Example 2.

$$\begin{cases} \varepsilon y''(x) + y'(x) - 2y(x-\delta) - 5y(x) + y(x+\eta) = 0, \\ y(x) = 1, \quad -\delta \le x \le 0 \quad y(x) = 1, \quad 1 \le x \le 1+\eta. \end{cases}$$

This problem has a rapid change near $x = 0$. By the use of Taylor's Series expansion for both the delay and shift terms as in (2), we have

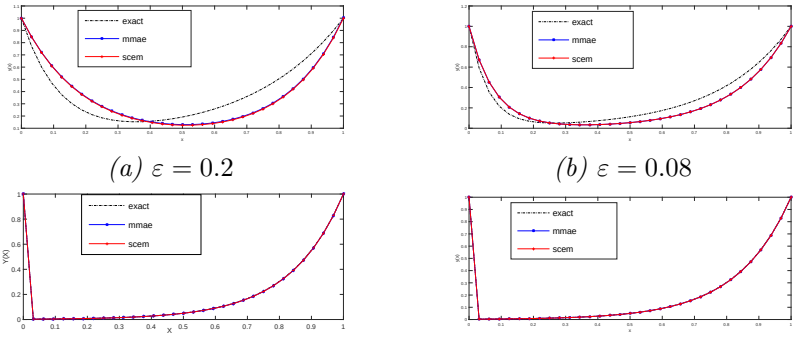
$$\varepsilon^* y''(x) + y'(x) - \left[\frac{6}{1+2\delta+\eta}\right] y(x) = 0, \qquad where\ \varepsilon^* = \frac{\varepsilon}{1+2\delta+\eta}.$$

The one-term SCEM and MMAE approximations are given by

$$y_{mmae} = \left[1 - \exp\left(\frac{-6}{1+2\delta+\eta}\right)\right] \exp\left(\frac{-x}{\varepsilon^*}\right) + \exp\left(\frac{6(x-1)}{1+2\delta+\eta}\right),$$

$$y_{scem} = -\left[\frac{\exp(\frac{-6}{1+2\delta+\eta}) - 1}{\exp(\frac{-1}{\varepsilon^*}) - 1}\right] \exp\left(\frac{-1}{\varepsilon^*}\right) + \left[\frac{\exp(\frac{-6}{1+2\delta+\eta}) - 1}{\exp(\frac{-1}{\varepsilon^*}) - 1}\right] \exp\left(\frac{-x}{\varepsilon^*}\right) + \exp\left[\frac{6(x-1)}{1+2\delta+\eta}\right].$$

The comparison of errors of this example for both the proposed methods and the method prescribed in [10] for $N = 8$ and $N = 256$ are given in Tables 4 and 5, respectively. These comparison results confirm the efficiency of the proposed method.



(a) $\varepsilon = 0.2$ (b) $\varepsilon = 0.08$

(c) $\varepsilon = 0.001$ (d) $\varepsilon = 0.0007$

Figure 3: *Comparison of solutions with different values of ε for Example 3.*

ε	$L^\infty error\ in\ MMAE$	$L^\infty error\ in\ SCEM$	Error in Upwind
0.0001	0.00022006	0.00022006	0.05991807
0.0005	0.00109602	0.00109602	0.05951004
0.0010	0.00218145	0.00218145	0.05900561
0.0050	0.01050670	0.01050670	0.05517982
0.0100	0.02012147	0.02012147	0.05086176
0.0500	0.07770075	0.07770075	0.03679054
0.1000	0.12342561	0.12345914	0.03402622
0.3000	0.16404633	0.18962499	0.01593674
0.4000	0.16021248	0.19825547	0.01255964
0.6000	0.15178975	0.20931461	0.00894834
0.8000	0.21056755	0.21768491	0.00694484
1.0000	0.26983417	0.22335443	0.00562079

Table 3: Numerical results with different values of ε for Example 3.

ε	$L^\infty error\ in\ MMAE$	$L^\infty error\ in\ SCEM$	Error in [10]
10^{-1}	0.10596702	0.10597709	0.12011566
10^{-2}	0.01882187	0.01882187	0.18727108
10^{-3}	0.00209815	0.00209815	0.20429729
10^{-4}	0.00021228	0.00021228	0.20614146
10^{-5}	0.00002125	0.00002125	0.20632746
10^{-6}	0.00000212	0.00000212	0.20634608

Table 4: Comparison of maximum error with $N = 8$ for Example 3.

ε	$L^\infty error\, in\, MMAE$	$L^\infty error\, in\, SCEM$	Error in [10]
10^{-1}	0.10685019	0.10685019	0.00775036
10^{-2}	0.01976044	0.01976044	0.00799076
10^{-3}	0.00218116	0.00218116	0.00963304
10^{-4}	0.00022045	0.00022045	0.00984236
10^{-5}	0.00002206	0.00002206	0.00986365
10^{-6}	0.00000220	0.00000220	0.00986578

Table 5: Comparison of maximum error with $N = 256$ for Example 3.

5 Conclusion

In this work, the well-known MMAE and a parameter uniform numerical scheme, that is, the upwind scheme in Shishkin mesh were compared with the proposed relatively new method, that is, SCEM, for solving singularly perturbed differential-difference equation with mixed shifts. It was observed that both MMAE and SCEM provide highly accurate approximations to the exact solution for comparatively small values ε. Similarly, comparison results confirmed that the numerical scheme, which gives better results for higher values of N, at the same time, MMAE and SCEM give higher-order accuracy for comparatively smaller values of N. It is observed that we do not require any matching procedure in SCEM as required for MMAE, and the boundary conditions are satisfied exactly for all values of the perturbation parameters. These properties make SCEM more flexible and a better alternative while solving singularly perturbed differential equations.

Acknowledgment

The first author expresses her sincere gratitude to DST, Govt. of India for providing INSPIRE fellowship (IF 180938).

References

1. Amiraliyev, G.M., Amiraliyeva, I.G. and Kudu, M. *A numerical treatment for singularly perturbed differential equations with integral boundary condition*, Appl. Math. Comput. 185(1) (2007), 574–582.

2. Arora, G. and Kaur, M. *Numerical simulation of singularly perturbed differential equation with small shift*, AIP Conf. Proc. 1860 (1) (2017), 020047.

3. Bush, A.W. *Perturbation methods for engineers and scientists*, CRC Press, London, 1992.

4. Cengizci, S., Natesan, S. and Atay, M.T. *An asymptotic-numerical hybrid method for singularly perturbed system of two-point reaction-diffusion boundary-value problems*, Turk. J. Math. 43, (2019), 460–472.

5. Chakravarthy, P.P. and Gupta, T. *Numerical solution of a weakly coupled system of singularly perturbed delay differential equations via cubic spline in tension*, Natl. Acad. Sci. Lett. 43(3), (2020), 259–262.

6. Chakravarthy, P.P. and Kumar, K. *A novel method for singularly perturbed delay differential equations of reaction-diffusion type*, Differ. Equ. Dyn. Syst. (2017), 1–12.

7. Cousteix, J. and Mauss, J. *Asymptotic analysis and boundary layers*, Springer Science and Business Media, (2007).

8. Duressa, G.F. and Reddy, Y.N. *Domain decomposition method for singularly perturbed differential difference equations with layer behavior*, Int. J. Eng. Appl. Sci. 7 (1), (2015), 86–102.

9. Kadalbajoo, M.K. and Sharma, K.K. *Numerical analysis of singularly perturbed delay differential equations with layer behavior*, APPL. MATH. Comput. 157(1), (2004), 11–28.

10. Kadalbajoo, M.K. and Sharma, K.K. *Numerical treatment of a mathematical model arising from a model of neuronal variability*, J. MATH. ANAL. APPL. 307(2), (2005), 606–627.

11. Kellogg, R.B. and Tsan, A. *Analysis of some difference approximations for a singular perturbation problem without turning points*, Math. Comput. 32(144), (1978), 1025–1039.

12. Kokotovic, P., Khali, H.K. and O'reilly, J. *Singular perturbation methods in control: analysis and design*, SIAM Books, 25, (1999).

13. Kudu, M., Amirali, I. and Amiraliyev, G.M. *A layer analysis of parameterized singularly perturbed boundary value problems*, Int. J. Appl. Math, 29(4), (2016), 439–449.

14. Kudu, M., Amirali, I. and Amiraliyev, G.M. *Uniform numerical approximation for parameter dependent singularly perturbed problem with integral boundary condition*, Miskolc Mathematical Notes, 19(1), (2018), 337–353, DOI: 10.18514/MMN.2018.2455

15. Lange, C.G. and Miura, R.M. *Singular perturbation analysis of boundary value problems for differential-difference equations. V. small shifts with layer behavior*, SIAM J. Appl. Math. 54(1), (1994), 249–272.

16. Lange, C.G. and Miura, R.M. *Singular perturbation analysis of boundary-value problems for differential-difference equations. VI. small shifts with rapid oscillations*, SIAM J. Appl. Math. 54(1), (1994), 273–283.

17. Mauss, J. and Cousteix, J. *Uniformly valid approximation for singular perturbation problems and matching principle*, Comptes Rendus Mecanique, 330(10), (2002), 697–702.

18. Melesse, W.G.,Tiruneh, A.A. and Derese, G.A. *Solving linear second-order singularly perturbed differential difference equations via initial value method*, Int. J. Differ. Equ. (2019).

19. Mohapatra, J. and Natesan, S. *Uniformly convergent second order numerical method for a singularly perturbed delay differential equation using a Shishkin mesh*, Neural, Parallel and Sci. Comput. 16(3), (2008), 353–370.

20. Mohapatra, J. and Natesan, S. *Uniformly convergent numerical method for singularly perturbed differential-difference equation using grid equidistribution*, Int. J. Numer. Method Biomed. Eng. 27(9), (2011), 1427–1445.

21. Mohapatra, J. and Natesan, S. *Parameter-uniform numerical methods for singularly perturbed mixed boundary value problems using grid equidistribution*, J. Appl. Math. Comput. 37(1), (2011), 247–265.

22. Mushahary, P., Sahu, S.R. and Mohapatra, J. *A parameter uniform numerical scheme for singularly perturbed differential-difference equations with mixed shifts*, J. Appl. Comput. Mech. 6(2), (2020), 344–356.

23. Rao, R.N. and Chakravarthy, P.P. *An exponentially fitted tridiagonal finite difference method for singularly perturbed differential-difference equations with small shift*, Ain Shams Eng. J. 5(4), (2014), 1351–1360.

24. Reddy, N.R. and Mohapatra, J. *An efficient numerical method for singularly perturbed two point boundary value problems exhibiting boundary layers*, Natl. Acad. Sci. Lett. 38(4), (2015), 355–359.

25. Saberi-Nadjafi, J. and Ghassabzade, F.A. *The numerical solution of the singularly perturbed differential-difference equations based on the meshless method*, Int. J. Appl. Math. Res. 3(2), (2014), 116.

26. Sirisha, L., Phaneendra, K. and Reddy, Y.N. *Mixed finite difference method for singularly perturbed differential difference equations with mixed shifts via domain decomposition*, Ain Shams Eng. J. 9(4), (2018), 647–654.

27. Stein, R.B. *Some models of neuronal variability*, Biophys. J. 7(1), (1967), 37–68.

How to cite this article
Priyadarshana, S., Sahu, S., Mohapatra, J. Asymptotic and numerical methods for solving singularly perturbed differential difference equations with mixed shifts. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 55-72. doi:10.22067/ijnao.2021.70731.1038

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 73-109
DOI:10.22067/ijnao.2021.70478.1036
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Numerical approximation for inverse problem of the Ostrovsky–Burgers equation

F. Ghanadian, R. Pourgholi, and S.H. Tabasi*

Abstract

This article considers a nonlinear inverse problem of the Ostrovsky–Burgers equation by using noisy data. Two B-Splines with different levels, the quintic B-spline and septic B-spline, are used to study this problem. For both B-splines, the stability and convergence analysis are calculated, and results show that an excellent estimation of the unknown functions of the nonlinear inverse problem.

AMS subject classifications (2020): 35Q53; 35B35, 68W25, 35R30, 65M70.

Keywords: Ostrovsky equation; Quintic B-spline collocation; Septic B-spline collocation; Convergence analysis; Stability analysis; Noisy data.

1 Introduction

In this paper, we consider the Ostrovsky–Burgers equation

*Corresponding author
Received 18 May 2021; revised 3 August 2021; accepted 8 August 2021
Fatemeh Ghanadian
School of Mathematics and Computer Science, Damghan University, Damghan 36715-364, Iran. e-mail: ghanadian85@gmail.com

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

Seyed Hashem Tabasi
School of Mathematics and Computer Science, Damghan University, Damghan 36715-364, Iran. e-mail: tabasi@du.ac.ir

$$(u_t + bu_{xxx} + uu_x - au_{xx})_x = \gamma u + f(x,t). \tag{1}$$

Equation (1) was appeared in modeling internal waves in the ocean or surface waves in a shallow channel with an uneven bottom under the effects of the interfacial friction (see [11, Chapter 1] and [16, 22]). In (1), the positive constants γ and a are the rotation and friction coefficients, respectively. The function f denotes the external force and b is the dispersion coefficient, which its sign is related to the type of dispersion. Ignoring the dissipation term u_{xx}, (1) leads to the Ostrovsky equation

$$(u_t + bu_{xxx} + uu_x)_x = \gamma u, \tag{2}$$

which was derived by Ostrovsky in 1978 [18] to model weakly nonlinear surface and internal waves in a rotating ocean; see also [9, 10]. It was also demonstrated in [17] that the nonlinear oblique magneto-acoustic waves in a rotating plasma can be described by (2). A model of the propagation of long internal waves in a deep rotating fluid can be found in [7]. If one considers the limit of no high-frequency dispersion $b = 0$, the resulting equation is called the Ostrovsky–Hunter equation [4]. When $\gamma = 0$, (1) turns into the Korteweg-de Vries-Burgers equation

$$u_t + bu_{xxx} + uu_x - au_{xx} = f(x,t), \tag{3}$$

which is the dissipated version of the KdV equation ($a = 0$) modeling the propagation of weakly nonlinear dispersive long waves in some contexts [19]. It is worth noting that despite the similarity of structures of (2) and the KdV equation, the Ostrovsky equation, unlike the KdV equation, is nonintegrable by the method of the inverse scattering transform [10].

In the present paper, we study numerically equation (1) in the domain $(x,t) \in [0,1] \times [0,T]$ with the final time T, the initial condition

$$u(x,0) = p(x), \qquad x \in [0,1] \tag{4}$$

and boundary conditions

$$u(0,t) = f_1(t), \quad u_x(0,t) = f_2(t), \quad t \in [0,T], \tag{5}$$

$$u(1,t) = g_1(t), \quad u_x(1,t) = g_2(t), \quad u_{xx}(1,t) = g_3(t), \quad t \in [0,T], \tag{6}$$

where $p(x)$, $g_1(t)$, $g_2(t)$, $g_3(t)$, and $f(x,t)$ are continuous known functions, while $f_1(t)$, $f_2(t)$, and the wave amplitude $u(x,t)$ are unknown, which remain to be determined. There are different methods to obtain numerical solutions of the dissipative KdV-type equations.

There are different methods to obtain numerical solutions of the dissipative KdV-type equations. Bhatta [2] found numerical solutions of (3) using the modified Bernstein polynomials (B-polynomials). In [14], soliton solutions of the KdVB equation were found by using two numerical methods,

finite difference and a semi-analytic method with the Adomian decomposition. A spectral method based on Lagrange polynomials was used in [24] to approximate solutions of boundary-value problems associated with (3). The classical radial basis functions (RBFs) collocation (Kansa) method for the numerical solution of (3) was formulated in [12]. A spectral collocation method based on differentiated Chebyshev polynomials was elaborated in [15] to obtain numerical solutions of (3). Chen and Wu [5] presented the MQ quasi-interpolation method to find the numerical solutions of (3). The local discontinuous Galerkin method was tested in [23] to study the Kuramoto–Sivashinsky equation.

Here, we consider two numerical methods to find the solutions of (1), the collocation method based on septic B-spline basis functions and quintic B-spline basis functions.

It is known that the use of B-splines has many different features and is effective in numerical works. One of the most important features is that the conditions on the continuity of functions are built-in and have smooth interpolation functions. On the other hand, as the support of each B-spline is embedded only on a few sub-intervals, the resulting matrix related to the discretized equation will be tightly banded.

Moreover, if one combines with collocation, then the solution procedure will be clear and short. In the present work, we provide the combination of the finite difference in t and the quintic B-spline collocation method in x, and employ the Tikhonov regularization method to solve the associated ill-conditioned system, which gives an efficient explicit solution that has high accuracy with the minimal computational effort for the inverse problem (1) and (4)–(6).

This paper is arranged as follows. In Section 2, a description of the septic B-splines collocation method, the uniform convergence, and the stability are explained. The quintic B-splines collocation method and its convergence and the stability are explained in Section 3. The numerical results are presented in Section 4, and finally, Section 5 completes this paper with some concluding remarks.

2 Septic B-spline collection method

To apply the Septic B-spline collection method, region of the solution of the problem is restrained over $0 \leq x \leq 1$. Space interval $[0, 1]$ is separated into uniformly sized finite elements of length h by the knots x_m like that $0 = x_0 < x_1 < \cdots < x_N = 1$. Lengths of these finite elements are $h = \frac{1-0}{N} = (x_{m+1} - x_m)$ for $m = 0, 1, 2, \ldots, N-1$.
Problem (1)–(6) will be solved with the over-specified conditions

$$u(\kappa, t) = h_1(t), \quad u_x(\kappa, t) = h_2(t), \quad u_{xx}(\kappa, t) = h_3(t), \tag{7}$$

where $t \in [0,T]$ and $0 < \kappa < 1$ is a fixed number. A spline is a function that is piecewise-defined by polynomial functions and possesses a high degree of smoothness at the places, where polynomial pieces connect. A B-spline is a spline function that has minimal support to the given degree, smoothness, and domain partition. We define the septic B-spline $B_j(x)$ for $j = -3(1)N+3$ by the following relations:

$$B_j(x) = h^{-7} \begin{cases} (x-x_{j-4})^7, & x \in [x_{j-4}, x_{j-3}), \\ (x-x_{j-4})^7 - 8(x-x_{j-3})^7, & x \in [x_{j-3}, x_{j-2}), \\ (x-x_{j-4})^7 - 8(x-x_{j-3})^7 & \\ +28(x-x_{j-2})^7, & x \in [x_{j-2}, x_{j-1}), \\ (x-x_{j-4})^7 - 8(x-x_{j-3})^7 & \\ +28(x-x_{j-2})^7 - 56(x-x_{j-1})^7, & x \in [x_{j-1}, x_j), \\ (x_{j+4}-x)^7 - 8(x_{j+3}-x)^7 & \\ +28(x_{j+2}-x)^7 - 56(x_{j+1}-x)^7, & x \in [x_j, x_{j+1}), \\ (x_{j+4}-x)^7 - 8(x_{j+3}-x)^7 & \\ +28(x_{j+2}-x)^7, & x \in [x_{j+1}, x_{j+2}), \\ (x_{j+4}-x)^7 - 8(x_{j+3}-x)^7, & x \in [x_{j+2}, x_{j+3}), \\ (x_{j+4}-x)^7, & x \in [x_{j+3}, x_{j+4}), \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The set

$$\Omega = \{B_{-3}(x), B_{-2}(x), B_{-1}(x), B_0(x), \ldots, B_N(x), \\ B_{N+1}(x), B_{N+2}(x), B_{N+3}(x)\}$$

forms a basis for the approximate solution, which will be defined over the interval $[0,1]$. Thus $\varpi = \text{Span}(\Omega)$ is a subspace of $C^2[0,1]$ and ϖ is $N+7$-dimensional. The values of B_j and its derivatives may be tabulated as in Table 1.

Now let $U(x,t) \in \varpi$ be the B-spline approximation to the exact solution $u(x,t)$ in the form

$$U(x,t) = \sum_{j=-3}^{N+3} c_j(t) B_j(x), \tag{9}$$

where $c_j(t)$ are time-dependent parameters determined from boundary conditions and collocation condition.

By substituting the trial function (8) into equation (9), the nodal values of $U, U^{'}, U^{''}, U^{'''}, U^{(4)}$, and $U^{(5)}$ are obtained in terms of the element parameters c_m by

Table 1: Values of $B_j(x)$ and its derivatives at the nodal points.

x	x_{j-4}	x_{j-3}	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}	x_{j+3}	x_{j+4}
$B_j(x)$	0	1	120	1191	2416	1191	120	1	0
$B_j'(x)$	0	$\frac{7}{h}$	$\frac{392}{h}$	$\frac{1715}{h}$	0	$\frac{-1715}{h}$	$\frac{-392}{h}$	$\frac{-7}{h}$	0
$B_j''(x)$	0	$\frac{42}{h^2}$	$\frac{1008}{h^2}$	$\frac{630}{h^2}$	$\frac{-3560}{h^2}$	$\frac{630}{h^2}$	$\frac{1008}{h^2}$	$\frac{42}{h^2}$	0
$B_j'''(x)$	0	$\frac{210}{h^3}$	$\frac{1680}{h^3}$	$\frac{-3990}{h^3}$	0	$\frac{3990}{h^3}$	$\frac{-1680}{h^3}$	$\frac{-210}{h^3}$	0
$B_j^{(4)}(x)$	0	$\frac{840}{h^4}$	0	$\frac{-7560}{h^4}$	$\frac{13440}{h^4}$	$\frac{-7560}{h^4}$	0	$\frac{840}{h^4}$	0
$B_j^{(5)}(x)$	0	$\frac{2520}{h^5}$	$\frac{-10080}{h^5}$	$\frac{12600}{h^5}$	0	$\frac{-12600}{h^5}$	$\frac{10080}{h^5}$	$\frac{-2520}{h^5}$	0

$$
\begin{aligned}
U_m &= c_{m-3} + 120c_{m-2} + 1191c_{m-1} + 2416c_m + 1191c_{m+1} + 120c_{m+2} + c_{m+3},\\
U_m' &= \frac{7}{h}(-c_{m-3} - 56c_{m-2} - 245c_{m-1} + 2451c_{m+1} + 56c_{m+2} + c_{m+3}),\\
U_m'' &= \frac{42}{h^2}(c_{m-3} + 24c_{m-2} + 15c_{m-1} - 80c_m + 15c_{m+1} + 24c_{m+2} + c_{m+3}),\\
U_m''' &= \frac{210}{h^3}(-c_{m-3} - 8c_{m-2} + 19c_{m-1} - 19c_{m+1} + 8c_{m+2} + c_{m+3}),\\
U_m^{(4)} &= \frac{840}{h^4}(c_{m-3} - 9c_{m-1} + 16c_m - 9c_{m+1} + c_{m+3}),\\
U_m^{(5)} &= \frac{2520}{h^5}(-c_{m-3} + 4c_{m-2} - 5c_{m-1} + 5c_{m+1} - 4c_{m+2} + c_{m+3}).
\end{aligned}
\tag{10}
$$

2.1 Numerical discretization

Let us consider a uniform mesh (x_i, t_j) to discretize the region $[0,1] \times [0,T]$. More precisely, each (x_i, t_j) is defined by $x_i = ih$, $i = 0, 1, 2, \ldots, N$ and $t_j = jk$ for $j = 0, 1, 2, \ldots$, where h and k are mesh sizes in the space and time directions, respectively (see [8]). Suppose that u^n is linearly interpolated between two time levels n and $n+1$ by using the Crank–Nicolson formula and then its time derivative is discretized by the forward finite difference formula:

$$
u = \frac{u^{n+1} + u^n}{2}, \quad u_t^n = \frac{u^{n+1} - u^n}{k}, \tag{11}
$$

where $u^n = u(x, t_n)$ and $u^0 = u(x, 0) = p(x)$. Substituting the above approximation into equation (1) and discretizing in time variable, we have

$$
u_{xt} = \gamma u + a u_{xxx} - b u_{xxxx} - u u_{xx} - u_x^2 + f(x,t),
$$

so

$$u_x^{n+1}-u_x^n=\frac{k}{2}\left[\gamma u^{n+1}+\gamma u^n+au_{xxx}^{n+1}+au_{xxx}^n-(uu_{xx})^{n+1}-(uu_{xx})^n\right.$$
$$-\left(u_x^2\right)^{n+1}-\left(u_x^2\right)^n-bu_{xxxx}^{n+1}-bu_{xxxx}^n$$
$$\left.+f(x,t_{n+1})+f(x,t_n)\right].$$

The nonlinear term is linearized by using the quasi-linearization formula as given below:

$$f\left(u^{n+1},u_x^{n+1}\right)=f(u^n,u_x^n)+(u^{n+1}-u^n)\frac{\partial f^n}{\partial u}+(u_x^{n+1}-u_x^n)\frac{\partial f^n}{\partial u_x}.$$

Thus, we have

$$(uu_{xx})^{n+1}=u^{n+1}u_{xx}^n+u^nu_{xx}^{n+1}-(uu_{xx})^n$$

and

$$\left(u_x^2\right)^{n+1}=2{u_x}^{n+1}{u_x}^n-\left(u_x^2\right)^n.$$

Now, we obtain by rearranging the terms that

$$\begin{aligned}&\theta_1u^{n+1}+\theta_2u_x^{n+1}+\theta_3u_{xx}^{n+1}-\theta_4u_{xxx}^{n+1}+\theta_5u_{xxxx}^{n+1}\\&\quad=\theta_6u^n+u_x^n+\theta_4u_{xxx}^n-\theta_5u_{xxxx}^n+\theta_7(f(x,t_{n+1})+f(x,t_n)),\end{aligned}\tag{12}$$

where

$$\theta_1=\frac{k}{2}\left(u_{xx}^n-\gamma\right),\quad\theta_2=\frac{7}{h}\left(1+ku_x^n\right),\quad\theta_3=\frac{42}{h^2}\left(\frac{k}{2}u^n\right)$$
$$\theta_4=\frac{210}{h^3}\left(\frac{k}{2}a\right),\quad\theta_5=\frac{840}{h^4}\left(\frac{k}{2}b\right),\quad\theta_6=\frac{k}{2}\gamma,\quad\theta_7=\frac{k}{2}.$$

By replacing the approximate solution U with u, and using the nodal values U and the derivatives of U, we deduce from equation (10) at the knots in equation (12) the following difference equation in the variable c:

$$\begin{aligned}A^*c_{i-3}^{n+1}+B^*c_{i-2}^{n+1}+C^*c_{i-1}^{n+1}+D^*c_i^{n+1}+E^*c_{i+1}^{n+1}+F^*c_{i+2}^{n+1}+G^*c_{i+3}^{n+1}\\=H(x_i,t_n)+H(x_i,t_{n+1}),\end{aligned}\tag{13}$$

where

$$\begin{aligned}A^*&=\theta_1-\theta_2+\theta_3+\theta_4+\theta_5,\\B^*&=120\theta_1-56\theta_2+24\theta_3+8\theta_4,\\C^*&=1191\theta_1-245\theta_2+15\theta_3-19\theta_4-9\theta_5,\\D^*&=2416\theta_1-80\theta_3+16\theta_5,\\E^*&=1191\theta_1+245\theta_2+15\theta_3+19\theta_4-9\theta_5,\end{aligned}$$

$$F^* = 120\theta_1 + 56\theta_2 + 24\theta_3 - 8\theta_4,$$
$$G^* = \theta_1 + \theta_2 + \theta_3 - \theta_4 + \theta_5,$$

$$\begin{aligned} H(x_i,t_n) &= \theta_6(c_{i-3} + 120c_{i-2} + 1191c_{i-1} + 2416c_i + 1191c_{i+1} + 120c_{i+2} + c_{i+3}) \\ &+ \left(\frac{7}{h}\right)(-c_{i-3} - 56c_{i-2} - 245c_{i-1} + 2451c_{i+1} + 56c_{i+2} + c_{i+3}) \\ &+ \theta_4\left(\frac{210}{h^3}\right)(-c_{i-3} - 8c_{i-2} + 19c_{i-1} - 19c_{i+1} + 8c_{i+2} + c_{i+3}) \\ &- \theta_5\left(\frac{840}{h^4}\right)(c_{i-3} - 9c_{i-1} + 16c_i - 9c_{i+1} + c_{i+3}) \\ &+ \theta_7 f(x_i,t_n), \end{aligned}$$

and

$$H(x_i,t_{n+1}) = \theta_7 f(x_i,t_{n+1}), \quad 0 \le i \le N.$$

System (13) consists of $(N+1)$ linear equations with $(N+7)$ unknowns,

$$(c_{-3}, c_{-2}, c_{-1}, c_0, \ldots, c_N, c_{N+1}, c_{N+2}, c_{N+3})^T.$$

To have a unique solution to the above system, we are required the over-specified conditions (7). Suppose that $\kappa = x_s$, $1 \le s \le N-1$, then

$$u(x_s,t) = h_1(t), \quad u_x(x_s,t) = h_2(t), \quad u_{xx}(x_s,t) = h_3(t), \tag{14}$$

where $t \in [0,T]$. If we consider $m = s$ in (10), then we have

$$\begin{aligned} h_1(t_{n+1}) &= c_{s-3}^{n+1} + 120c_{s-2}^{n+1} + 1191c_{s-1}^{n+1} + 2416c_s^{n+1} + 1191c_{s+1}^{n+1} + 120c_{s+2}^{n+1} + c_{s+3}^{n+1}, \\ h_2(t_{n+1}) &= \frac{7}{h}\left(-c_{s-3}^{n+1} - 56c_{s-2}^{n+1} - 245c_{s-1}^{n+1} + 2451c_{s+1}^{n+1} + 56c_{s+2}^{n+1} + c_{s+3}^{n+1}\right), \\ h_3(t_{n+1}) &= \frac{42}{h^2}\left(c_{s-3}^{n+1} + 24c_{s-2}^{n+1} + 15c_{s-1}^{n+1} - 80c_s^{n+1} + 15c_{s+1}^{n+1} + 24c_{s+2}^{n+1} + c_{s+3}^{n+1}\right), \\ g_1(t_{n+1}) &= c_{N-3}^{n+1} + 120c_{N-2}^{n+1} + 1191c_{N-1}^{n+1} + 2416c_N^{n+1} + 1191c_{N+1}^{n+1} + 120c_{N+2}^{n+1} + c_{N+3}^{n+1}, \\ g_2(t_{n+1}) &= \frac{7}{h}\left(-c_{N-3}^{n+1} - 56c_{N-2}^{n+1} - 245c_{N-1}^{n+1} + 2451c_{N+1}^{n+1} + 56c_{N+2}^{n+1} + c_{N+3}^{n+1}\right), \\ g_3(t_{n+1}) &= \frac{42}{h^2}\left(c_{N-3}^{n+1} + 24c_{N-2}^{n+1} + 15c_{N-1}^{n+1} - 80c_{m-1}^{n+1} + 15c_{N+1}^{n+1} + 24c_{N+2}^{n+1} + c_{N+3}^{n+1}\right). \end{aligned}$$

Hence, we derive that

$$AC = B, \tag{15}$$

is a system of $(N+7)$ linear equations with $(N+7)$ unknowns, where

$$\begin{aligned} A[1,s+1] &= A[1,s+7] = A[3,s+1] = A[2,s+7] = A[N+5,N+7] \\ &= A[3,s+7] = A[N+5,N+1] = A[N+6,N+7] \\ &= A[N+7,N+1] = A[N+7,N+7] = -A[N+6,N+1] = 1, \end{aligned}$$

and

$A[N+7,N+2]=A[N+7,N+6]=A[1,s+2]=A[1,s+6]=120,$
$A[N+7,N+3]=A[N+7,N+5]=A[1,s+3]=A[1,s+5]=1191,$
$A[N+7,N+4]=A[1,s+4]=2416,$
$A[2,s+6]=-A[2,s+2]=-A[N+6,N+2]=A[N+6,N+6]=56,$
$A[N+6,N+5]=-A[N+6,N+3]=-A[2,s+3]=A[2,s+5]=245,$
$A[3,s+2]=A[3,s+6]=A[N+5,N+2]=A[N+5,N+6]=24,$
$A[3,s+3]=A[3,s+5]=A[N+5,N+3]=A[N+5,N+5]=15,$
$A[3,s+4]=A[N+5,N+4]=-80.$

Therefore, we have

$$A=\begin{pmatrix}
0 & \dots & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & 0 & \dots \\
0 & \dots & -1 & -56 & -245 & 0 & 245 & 56 & 1 & 0 & \dots \\
0 & \dots & 1 & 24 & 15 & -80 & 15 & 24 & 1 & 0 & \dots \\
A^* & B^* & C^* & D^* & E^* & F^* & G^* & 0 & \dots & & \\
0 & A^* & B^* & C^* & D^* & E^* & F^* & G^* & 0 & \dots & \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
0 & \dots & A^* & B^* & C^* & D^* & E^* & F^* & G^* & 0 & \\
0 & \dots & & A^* & B^* & C^* & D^* & E^* & F^* & G^* & \\
0 & \dots & & 1 & 24 & 15 & -80 & 15 & 24 & 1 & \\
0 & \dots & & -1 & -56 & -245 & 0 & 245 & 56 & 1 & \\
0 & \dots & & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 &
\end{pmatrix}_{(N+7)\times(N+7)},$$

with

$$C=\Big(c_{-3}^{(n+1)},c_{-2}^{(n+1)},c_{-1}^{(n+1)},c_{0}^{(n+1)},\dots,c_{s}^{(n+1)},\dots,c_{N}^{(n+1)},c_{N+1}^{(n+1)},c_{N+2}^{(n+1)},c_{N+3}^{(n+1)}\Big)^T,$$

$$B=\Big(B_{-3}^{(n)},B_{-2}^{(n)},B_{-1}^{(n)},B_{0}^{(n)},\dots,B_{s}^{(n)},\dots,B_{N}^{(n)},B_{N+1}^{(n)},B_{N+2}^{(n)},B_{N+3}^{(n)}\Big)^T,$$

and

$$B_{-3}^{(n)}=h_1(t_{n+1}),\quad B_{-2}^{(n)}=\left(\frac{h}{7}\right)h_2(t_{n+1}),\quad B_{-1}^{(n)}=\left(\frac{h^2}{42}\right)h_3(t_{n+1}),$$
$$B_{i}^{(n)}=H(x_i,t_n)+H(x_i,t_{n+1}),\quad B_{N+1}^{(n)}=\left(\frac{h^2}{42}\right)g_3(t_{n+1}),$$
$$B_{N+2}^{(n)}=\left(\frac{h}{7}\right)g_2(t_{n+1}),\quad B_{N+3}^{(n)}=g_1(t_{n+1}),$$

where $0 \leq i \leq N$. We notice that the matrix A is ill-conditioned, so we obtain the solution of system (15) by using the Tikhonov regularization method.

2.2 The initial vector c^0

By the initial condition (4) combined with the boundary and over-specified conditions (7), the initial vector c^0 can be written as the following expressions:

$$\begin{aligned}
u(x_s,t_0) &= c^0_{s-3} + 120c^0_{s-2} + 1191c^0_{s-1} + 2416c^0_s + 1191c^0_{s+1} + 120c^0_{s+2} + c^0_{s+3}\\
&= h_1(t_0),\\
u_x(x_s,t_0) &= \frac{7}{h}\left(-c^0_{s-3} - 56c^0_{s-2} - 245c^0_{s-1} + 2451c^0_{s+1} + 56c^0_{s+2} + c^0_s\right)\\
&= h_2(t_0),\\
u_{xx}(x_s,t_0) &= \frac{42}{h^2}\left(c^0_{s-3} + 24c^0_{s-2} + 15c^0_{s-1} - 80c^0_s + 15c^0_{s+1} + 24c^0_{s+2} + c^0_{s+3}\right)\\
&= h_3(t_0),\\
u(x_i,t_0) &= c^0_{i-3} + 120c^0_{i-2} + 1191c^0_{i-1} + 2416c^0_i + 1191c^0_{i+1} + 120c^0_{i+2} + c^0_{i+3}\\
&= p(x_i),
\end{aligned}$$

where $0 \leq i \leq N$ and

$$\begin{aligned}
u_{xx}(x_N,t_0) &= \frac{42}{h^2}\left(c^0_{N-3} + 24c^0_{N-2} + 15c^0_{N-1} - 80c^0_{m-1} + 15c^0_{N+1} + 24c^0_{N+2} + c^0_{N+3}\right)\\
&= g_3(t_0),\\
u_x(x_N,t_0) &= \frac{7}{h}\left(-c^0_{N-3} - 56c^0_{N-2} - 245c^0_{N-1} + 2451c^0_{N+1} + 56c^0_{N+2} + c^0_{N+3}\right)\\
&= g_2(t_0),\\
u(x_N,t_0) &= c^0_{N-3} + 120c^0_{N-2} + 1191c^0_{N-1} + 2416c^0_N + 1191c^0_{N+1} + 120c^0_{N+2} + c^0_{N+3}\\
&= g_1(t_0).
\end{aligned}$$

This yields an $(N+7)\times(N+7)$-system of equations of the form of

$$AC^0 = \beta, \tag{16}$$

where

$$\begin{aligned}
A[1,s+1] &= A[1,s+7] = A[3,s+1] = A[2,s+7] = A[N+5,N+7]\\
&= A[3,s+7] = A[N+5,N+1] = A[N+6,N+7]\\
&= A[N+7,N+1] = A[N+7,N+7] = -A[N+6,N+1] = 1,
\end{aligned}$$

and

$A[N+7,N+2]=A[N+7,N+6]=A[1,s+2]=A[1,s+6]=120,$
$A[N+7,N+3]=A[N+7,N+5]=A[1,s+3]=A[1,s+5]=1191,$
$A[N+7,N+4]=A[1,s+4]=2416,$
$A[2,s+6]=-A[2,s+2]=-A[N+6,N+2]=A[N+6,N+6]=56,$
$A[N+6,N+5]=-A[N+6,N+3]=-A[2,s+3]=A[2,s+5]=245,$
$A[3,s+2]=A[3,s+6]=A[N+5,N+2]=A[N+5,N+6]=24,$
$A[3,s+3]=A[3,s+5]=A[N+5,N+3]=A[N+5,N+5]=15,$
$A[3,s+4]=A[N+5,N+4]=-80.$

Thus, we have

$$A=\begin{pmatrix}
0 & \dots & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & 0 & \dots \\
0 & \dots & -1 & -56 & -245 & 0 & 245 & 56 & 1 & 0 & \dots \\
0 & \dots & 1 & 24 & 15 & -80 & 15 & 24 & 1 & 0 & \dots \\
 & & & & & & & & & & \vdots \\
1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & 0 & \dots & & \\
0 & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 & 0 & \dots & \\
\vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\
0 & & \dots & & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1 \\
0 & & \dots & & 1 & 24 & 15 & -80 & 15 & 24 & 1 \\
0 & & \dots & & -1 & -56 & -245 & 0 & 245 & 56 & 1 \\
0 & & \dots & & 1 & 120 & 1191 & 2416 & 1191 & 120 & 1
\end{pmatrix}_{(N+7)\times(N+7)},$$

$$C^0=\begin{pmatrix} c^0_{-3} \\ c^0_{-2} \\ c^0_{-1} \\ c^0_0 \\ \vdots \\ c^0_N \\ c^0_{N+1} \\ c^0_{N+2} \\ c^0_{N+3} \end{pmatrix}_{(N+7)\times 1}, \quad \beta=\begin{pmatrix} h_1(t_0) \\ (\frac{h}{7})h_2(t_0) \\ (\frac{h^2}{42})h_3(t_0) \\ p(x_0) \\ vdots \\ p(x_N) \\ (\frac{h^2}{42})g_3(t_0) \\ (\frac{h}{7})g_2(t_0) \\ g_1(t_0) \end{pmatrix}_{(N+7)\times 1}.$$

Finally, the solution of (16) can be obtained by using the Tikhonov regularization method.

2.3 Convergence analysis

Here, we are going to check the convergence of our algorithm. Suppose that $U(x)=\sum_{j=-3}^{N+3} c_j B_j(x)$ is the B-spline collocation approximation of $u(x)$,

where $u(x)$ is the exact solution of (1) with the over-specific conditions (7) and initial condition (4). Due to round off errors in computations, we can assume that $\widehat{U}(x)$ is the computed spline for $U(x)$ in such a way that $\widehat{U}(x) = \sum_{j=-2}^{N+1} \hat{c}_j B_j(x)$, where

$$\widehat{C} = (\hat{c}_{-3}, \hat{c}_{-2}, \hat{c}_{-1}, \hat{c}_0, \ldots, \hat{c}_N, \hat{c}_{N+1}, \hat{c}_{N+2}, \hat{c}_{N+3}).$$

Following (15) for $\widehat{U}$, we have

$$A\widehat{C} = \widehat{B},$$

where

$$\widehat{B} = (h_1(t_{n+1}), h_2(t_{n+1}), h_3(t_{n+1}), \hat{\tau}_0, \hat{\tau}_1, \ldots, \hat{\tau}_N, g_3(t_{n+1}), g_2(t_{n+1}), g_1(t_{n+1})),$$

and

$$\begin{aligned}\hat{\tau}_i = {} & \theta_6(c_{i-3} + 120c_{i-2} + 1191c_{i-1} + 2416c_i + 1191c_{i+1} + 120c_{i+2} + c_{i+3}) \\ & + \left(\frac{7}{h}\right)(-c_{i-3} - 56c_{i-2} - 245c_{i-1} + 2451c_{i+1} + 56c_{i+2} + c_{i+3}) \\ & + \theta_4\left(\frac{210}{h^3}\right)(-c_{i-3} - 8c_{i-2} + 19c_{i-1} - 19c_{i+1} + 8c_{i+2} + c_{i+3}) \\ & - \theta_5\left(\frac{840}{h^4}\right)(c_{i-3} - 9c_{i-1} + 16c_i - 9c_{i+1} + c_{i+3}) \\ & + \theta_7\left(f(x_i, t_n) + f(x_i, t_{n+1})\right).\end{aligned}$$

Consequently, we have

$$A(C - \widehat{C}) = (B - \widehat{B}).$$

The following lemma will be important in our analysis.

Lemma 1. If $\{B_{-3}, B_{-2}, B_{-1}, B_0, \ldots, B_N, B_{N+1}, B_{N+2}, B_{N+3}\}$ is the septic B-spline, then

$$\left|\sum_{j=-3}^{N+3} B_j(x)\right| \leq 7456, \quad x \in [0, 1]. \tag{17}$$

Proof. At any node x_i, we have from Table 1 that

$$\left|\sum_{j=-3}^{N+3} B_j\right| \leq \sum_{j=-3}^{N+3} |B_j| = 1 + 120 + 1191 + 2416 + 1191 + 120 + 1 = 5040.$$

For any point x in each sub-interval $[x_{j-1}, x_j]$, we have analogously that

$$\sum_{j=-3}^{N+3} |B_j| \leq 7456.$$

This proves the lemma. □

To obtain a suitable estimate for the error

$$\|u(x) - U(x)\|_\infty ,$$

we need to bound the error terms $\left\|u(x) - \widehat{U}\right\|_\infty$ and $\left\|\widehat{U} - U(x)\right\|_\infty$ separately. To do so, we recall the following theorem.

Theorem 1. Suppose that $g(x) \in C^8[0,1]$ and $|g^8(x)| \leq L$, for all $x \in [0,1]$. Assume that

$$\Delta = \{0 = x_0 < x_1 < \cdots < x_N = 1\},$$

is the partition of $[0,1]$ of step size h. If $S_\Delta(x)$ is the unique spline function interpolate $g(x)$ at nodes $x_0, x_1, \ldots, x_N \in \Delta$, then there exist a constant $\lambda_j \leq 2$ such that we have for all $x \in [0,1]$, that

$$\|g^j(x) - S^j_\Delta(x)\|_\infty \leq \lambda_j L h^{8-j}, \qquad j = 0, \ldots, 7,$$

where $\|\cdot\|$ represents the L^∞-norm.

Proof. See [1] for a proof. □

Now first we find a bound on $\|B - \widehat{B}\|_\infty$. Following [20, Theorem 7] and applying Theorem 1, there exists $W = W_L > 0$ such that

$$\begin{aligned}\left\|B - \widehat{B}\right\|_\infty &\leq W\left(\left|U(x) - \widehat{U}(x)\right| + \left|U^{'}(x) - \widehat{U}^{'}(x)\right| + \left|U^{''}(x) - \widehat{U}^{''}(x)\right|\right.\\ &\quad \left. + \left|U^{'''}(x) - \widehat{U}^{'''}(x)\right| + \left|U^{(4)}(x) - \widehat{U}^{(4)}(x)\right|\right)\\ &\leq WL\lambda_0 h^8 + WL\lambda_1 h^7 + WL\lambda_2 h^6 + WL\lambda_3 h^5 + WL\lambda_4 h^4.\end{aligned}$$

Thus we can get

$$\|B - \widehat{B}\|_\infty \leq W_1 h^4, \tag{18}$$

where

$$W_1 = WL(\lambda_0 h^4 + \lambda_1 h^3 + \lambda_2 h^2 + \lambda_3 h + \lambda_4).$$

Since the matrix A is ill-conditioned, we have from the Tikhonov regularization solution that

$$\left(C - \widehat{C}\right) = \left[A^T A + \alpha\left(R^{(2)}\right)^T R^2\right]^{-1} A^T\left(B - \widehat{B}\right),$$

after taking the L^∞-norm,

$$\left\| C - \widehat{C} \right\|_\infty \leq \left\| \left[A^T A + \alpha \left(R^{(2)} \right)^T R^2 \right]^{-1} A^T \right\|_\infty \left\| B - \widehat{B} \right\|_\infty \leq W_2 h^4, \tag{19}$$

where

$$W_2 = W_1 \left\| \left[A^T A + \alpha \left(R^{(2)} \right)^T R^2 \right]^{-1} A^T \right\|_\infty .$$

Now we observe that

$$U(x) - \widehat{U}(x) = \sum_{i=-3}^{N+3} (c_i - \hat{c}_i) B_i(x).$$

By taking the L^∞-norm and using (17) and (19), we have

$$\begin{aligned} \left\| U(x) - \widehat{U}(x) \right\|_\infty = \left\| \sum_{i=-3}^{N+3} (c_i - \hat{c}_i) B_i(x) \right\|_\infty &\leq \| (c_i - \hat{c}_i) \|_\infty \left| \sum_{i=-3}^{N+3} B_i(x) \right| \\ &\leq 7456 W_2 h^4. \end{aligned} \tag{20}$$

Theorem 2. The time discretization process (11) that we use to discretize equation (1) in time, is convergent of the first order.

Proof. See [21] □

Theorem 3. Let $u \in C^8[0,1]$ be an exact solution of (1) such that

$$\left| \frac{\partial^8 u(x,t)}{\partial x^8} \right| \leq L,$$

for all x and t. If $U(x,t)$ is the numerical approximation by our method of u, then

$$\| u(x) - U(x) \|_\infty \leq O(k + h^4).$$

Proof. We have from Theorem 1 that

$$\| u(x) - U(x) \|_\infty \leq \lambda_0 L h^8. \tag{21}$$

Equations (20) and (21) imply that

$$\begin{aligned} \| u(x) - U(x) \|_\infty &\leq \left\| u(x) - \widehat{U}(x) \right\|_\infty + \left\| \widehat{U}(x) - U(x) \right\|_\infty \\ &\leq \lambda_0 L h^8 + 7456 W_2 h^4 = \Upsilon h^4, \end{aligned}$$

where $\Upsilon = \lambda_0 L h^4 + 7456 W_2$. Therefore we deduce from Theorem 2 that

$$\|u(x) - U(x)\|_\infty \leq \rho(k + h^4),$$

where ρ is a finite positive constant independent of k and h. This completes the proof.

□

2.4 Stability analysis

In this section, we will investigate the stability by applying the Von-Neuman stability analysis. Without loss of generality and for the sake of simplicity, we assume in equation (1) that $f(x, y) = 0$. On the other hand, as we have linearized the nonlinear term uu_x by considering u as a constant k_1 in equation (1), then the equation can be rewritten by

$$(u_t + bu_{xxx} + k_1 u_x - au_{xx})_x = \gamma u. \tag{22}$$

Substituting the approximation (11) into equation (22), by discretizing in time we have that

$$\begin{aligned} -P_1 u^{n+1} + u_x^{n+1} + P_2 u_{xx}^{n+1} - P_3 u_{xxx}^{n+1} + P_4 u_{xxxx}^{n+1} \\ = P_1 u^n + u_x^n - P_2 u_{xx}^n + P_3 u_{xxx}^n - P_4 u_{xxxx}^n. \end{aligned}$$

If $\frac{k}{2} = \tau$, then

$$P_1 = \tau\gamma, \quad P_2 = \frac{42}{h^2}\tau k_1 \quad P_3 = \frac{210}{h^3}\tau a, , \quad P_4 = \frac{840}{h^4}\tau b. \tag{23}$$

In terms of unknown time parameters c_i, the equation can be written from (10) by

$$\begin{aligned} &x_1 c_{i-3}^{n+1} + x_2 c_{i-2}^{n+1} + x_3 c_{i-1}^{n+1} + x_4 c_i^{n+1} + x_5 c_{i+1}^{n+1} + x_6 c_{i+2}^{n+1} + x_7 c_{i+3}^{n+1} \\ &\quad = x_8 c_{i-3}^n + x_9 c_{i-2}^n + x_{10} c_{i-1}^n + x_{11} c_i^n + x_{12} c_{i+1}^{n+1} + x_{13} c_{i+2}^{n+1} + x_{14} c_{i+3}^{n+1} \end{aligned}$$

where

$$\begin{aligned} x_1 &= -P_1 - 1 + P_2 + P_3 + P_4, \\ x_2 &= -120P_1 - 56 + 24P_2 + 8P_3, \\ x_3 &= -1191P_1 - 245 + 15P_2 - 19P_3 - 9P_4, \\ x_4 &= -2416P_1 - 80P_2 + 16\theta_4, \\ x_5 &= -1191P_1 + 245 + 15P_2 + 19P_3 - 9P_4, \\ x_6 &= -120P_1 + 56 + 24P_2 - 8P_3, \\ x_7 &= -P_1 + 1 + P_2 - P_3 + P_4, \end{aligned} \tag{24}$$

$$
\begin{aligned}
x_8 &= P_1 - 1 - P_2 - P_3 - P_4,\\
x_9 &= 120P_1 - 56 - 24P_2 - 8P_3,\\
x_{10} &= 1191P_1 - 245 - 15P_2 + 19P_3 + 9P_4,\\
x_{11} &= 2416P_1 + 80P_2 - 16\theta_4,\\
x_{12} &= 1191P_1 + 245 - 15P_2 - 19P_3 + 9P_4,\\
x_{13} &= 120P_1 + 56 - 24P_2 + 8P_3,\\
x_{14} &= P_1 + 1 - P_2 + P_3 - P_4.
\end{aligned}
$$

Now we substitute $c_m^n = A\zeta^n \exp(\mathrm{im}\beta h)$, where $i = \sqrt{-1}$, A is amplitude, h is the step length and β is the mode number. We get

$$\zeta = \frac{X_1 + iY_1}{X_2 + iY_2},$$

where

$$
\begin{aligned}
X_1 &= (x_8 + x_{14})\cos(3\beta h) + (x_9 + x_{13})\cos(2\beta h) + (x_{10} + x_{12})\cos(\beta h) + x_6,\\
X_2 &= (x_7 + x_1)\cos(3\beta h) + (x_2 + x_6)\cos(2\beta h) + (x_3 + x_5)\cos(\beta h) + x_4,\\
Y_1 &= (x_{14} - x_8)\sin(3\beta h) + (x_{13} - x_9)\sin(2\beta h) + (x_{12} - x_{10})\sin(\beta h),\\
Y_2 &= (x_7 - x_1)\sin(3\beta h) + (x_6 - x_2)\sin(2\beta h) + (x_5 - x_3)\sin(\beta h).
\end{aligned}
\tag{25}
$$

Hence, we have from (25) and (24) that $X_1 = X_2$. Setting

$$
\begin{aligned}
\rho &= 2\sin(3\beta h) + 112\sin(2\beta h) + 490\sin(\beta h),\\
\varphi &= P_3(2\sin(3\beta h) + 16\sin(2\beta h) - 38\sin(\beta h)),
\end{aligned}
\tag{26}
$$

we have that

$$Y_1 = \rho + \varphi, \quad Y_2 = \rho - \varphi.$$

To get the stability, we need to prove $|\zeta| < 1$, that is,

$$X_1^2 + (\rho + \varphi)^2 < X_1^2 + (\rho - \varphi)^2. \tag{27}$$

To this end, it is enough to show that $4\rho\varphi < 0$. If we rewrite (26) by

$$
\begin{aligned}
\rho &= 8\sin(\beta h)(\cos^2(\beta h) + 28\cos(\beta h) + 61),\\
\varphi &= 8P_3\sin(\beta h)(\cos(\beta h) + 5)(\cos(\beta h) - 1),
\end{aligned}
$$

then we obtain

$$4\rho\varphi = 256P_3\sin^2(\beta h)(\cos^2(\beta h) + 28\cos(\beta h) + 61)(\cos(\beta h) + 5)(\cos(\beta h) - 1).$$

By the positivity of the constants, $P_3 > 0$. The functions $\sin^2(\beta h)$, $\cos^2(\beta h) + 28\cos(\beta h) + 61$, and $\cos(\beta h) + 5$ are positive, but $\cos(\beta h) - 1$ is negative,

so $4\rho\varphi < 0$. Hence, we obtain that the seventh order B-spline collocation method is unconditionally stable.

3 Quintic B-splines collocation method

In this section, we are going to solve the inverse problem (1) by a new modification of the quintic B-splines collocation method with the over-specified conditions

$$\begin{aligned} u(\kappa,t) &= h_1(t), \\ u_x(\kappa,t) &= h_2(t), \end{aligned} \tag{28}$$

where $t \in [0, t_f]$ and $0 < \kappa < 1$ is a fixed number.

We define the quintic B-spline $B_i(x)$ for $j = -2(1)N+2$ by the following relation:

$$B_j(x) = h^{-5} \begin{cases} (x-x_{j-3})^5, & x \in [x_{j-3}, x_{j-2}), \\ (x-x_{j-3})^5 - 6(x-x_{j-2})^5, & x \in [x_{j-2}, x_{j-1}), \\ (x-x_{j-3})^5 - 6(x-x_{j-2})^5 + 15(x-x_{j-1})^5, & x \in [x_{j-1}, x_j), \\ (x_{j+3}-x)^5 - 6(x_{j+2}-x)^5 + 15(x_{j+1}-x)^5, & x \in [x_j, x_{j+1}), \\ (x_{j+3}-x)^5 - 6(x_{j+2}-x)^5, & x \in [x_{j+1}, x_{j+2}), \\ (x_{j+3}-x)^5, & x \in [x_{j+2}, x_{j+3}), \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

We note that the set

$$\Omega = \{B_{-2}(x), B_{-1}(x), B_0(x), \ldots, B_N(x), B_{N+1}(x), B_{N+2}(x)\},$$

forms a basis for an approximate solution that will be defined over the interval $[0,1]$. Hence, the set $\varpi = \text{Span}(\Omega)$ is a subspace of $C^2[0,1]$ and ϖ is $N+5$-dimensional. The values of B_j and its derivatives are tabulated in Table 2. Let $U(x,t) \in \varpi$ be the B-spline approximation to the exact solution $u(x,t)$ in the form

$$U(x,t) = \sum_{j=-2}^{N+2} c_j(t) B_j(x), \tag{30}$$

where $c_j(t)$ are time-dependent parameters determined by the boundary and collocation conditions.

Substituting the trial function (29) into (30), the nodal values of U, U', U'', U''', and $U^{(4)}$ are obtained in terms of the element parameters c_m by

Table 2: Values of $B_j(x)$ and its derivatives at the nodal points.

x	x_{j-3}	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}	x_{j+3}
$B_j(x)$	0	1	26	66	26	1	0
$B_j'(x)$	0	$\frac{5}{h}$	$\frac{50}{h}$	0	$\frac{-50}{h}$	$\frac{5}{h}$	0
$B_j''(x)$	0	$\frac{20}{h^2}$	$\frac{40}{h^2}$	$\frac{-120}{h^2}$	$\frac{40}{h^2}$	$\frac{20}{h^2}$	0
$B_j'''(x)$	0	$\frac{60}{h^3}$	$\frac{-120}{h^3}$	0	$\frac{120}{h^3}$	$\frac{-60}{h^3}$	0
$B_j^{(4)}(x)$	0	$\frac{120}{h^4}$	$\frac{-480}{h^4}$	$\frac{720}{h^4}$	$\frac{-480}{h^4}$	$\frac{120}{h^4}$	0

$$\begin{aligned}
U_m &= c_{m-2} + 26c_{m-1} + 66c_m + 26c_{m+1} + c_{m+2},\\
U_m' &= \frac{5}{h}(-c_{m-2} - 10c_{m-1} + 10c_{m+1} + c_{m+2}),\\
U_m'' &= \frac{20}{h^2}(c_{m-2} + 2c_{m-1} - 6c_m + 2c_{m+1} + c_{m+2}),\\
U_m''' &= \frac{60}{h^3}(-c_{m-2} + 2c_{m-1} - 2c_{m+1} + c_{m+2}),\\
U_m^{(4)} &= \frac{120}{h^4}(c_{m-2} - 4c_{m-1} + 6c_m - 4c_{m+1} + c_{m+2}).
\end{aligned} \tag{31}$$

3.1 Numerical discretization

To discretize the equation, similar to the previous section, we use (31) in

$$u_{xt} = \gamma u + a u_{xxx} - b u_{xxxx} - u u_{xx} - {u_x}^2 + f(x,t),$$

and obtain

$$\begin{aligned}
u_x^{n+1} - u_x^n = \frac{k}{2}\big[&\gamma u^{n+1} + \gamma u^n + a u_{xxx}^{n+1} + a u_{xxx}^n - (u u_{xx})^{n+1} - (u u_{xx})^n\\
&- \left(u_x^2\right)^{n+1} - \left(u_x^2\right)^n - b u_{xxxx}^{n+1} - b u_{xxxx}^n\\
&+ f(x,t_{n+1}) + f(x,t_n)\big].
\end{aligned}$$

The nonlinear term is also linearized by rearranging terms as given below:

$$\begin{aligned}
&\theta_1 u^{n+1} + \theta_2 u_x^{n+1} + \theta_3 u_{xx}^{n+1} - \theta_4 u_{xxx}^{n+1} + \theta_5 u_{xxxx}^{n+1} =\\
&\theta_6 u^n + u_x^n + \theta_4 u_{xxx}^n - \theta_5 u_{xxxx}^n + \theta_7(f(x,t_{n+1}) + f(x,t_n)),
\end{aligned} \tag{32}$$

where

$$\theta_1 = \frac{k}{2}\left(u_{xx}^n - \gamma\right), \quad \theta_2 = \frac{5}{h}(1 + ku_x^n), \quad \theta_3 = \frac{20}{h^2}\left(\frac{k}{2}u^n\right), \quad \theta_4 = \frac{60}{h^3}\left(\frac{k}{2}a\right),$$
$$\theta_5 = \frac{120}{h^4}\left(\frac{k}{2}b\right), \quad \theta_6 = \frac{k}{2}\gamma, \quad \theta_7 = \frac{k}{2}.$$

Again, by replacing the approximate solution U with u, and using the nodal values U and the derivatives of U, we obtain from equation (31) at the knots in equation (32) the following difference equation in the variable c:

$$A^* c_{i-2}^{n+1} + B^* c_{i-1}^{n+1} + C^* c_i^{n+1} + D^* c_{i+1}^{n+1} + E^* c_{i+2}^{n+1} = H(x_i, t_n) + H(x_i, t_{n+1}), \tag{33}$$

where

$$\begin{aligned}
A^* &= \theta_1 - \theta_2 + \theta_3 + \theta_4 + \theta_5,\\
B^* &= 26\theta_1 - 10\theta_2 + 2\theta_3 - 2\theta_4 - 4\theta_5,\\
C^* &= 66\theta_1 - 6\theta_3 + 6\theta_5,\\
D^* &= 26\theta_1 + 2\theta_2 + 2\theta_3 + 2\theta_4 - 4\theta_5,\\
E^* &= \theta_1 + \theta_2 + \theta_3 - \theta_4 + \theta_5,
\end{aligned}$$

$$\begin{aligned}
H(x_i, t_n) = {} & \theta_6\left(c_{i-2} + 26c_{i-1} + 66c_i + 26c_{i+1} + c_{i+2}\right)\\
& + \left(\frac{5}{h}\right)\left(-c_{i-2} - 10c_{i-1} + 10c_{i+1} + c_{i+2}\right)\\
& + \theta_4\left(\frac{60}{h^3}\right)\left(-c_{i-2} + 2c_{i-1} - 2c_{i+1} + c_{i+2}\right)\\
& - \theta_5\left(\frac{120}{h^4}\right)\left(c_{i-2} - 4c_{i-1} + 6c_i - 4c_{i+1} + c_{i+2}\right)\\
& + \theta_7 f(x_i, t_n),
\end{aligned}$$

and

$$H(x_i, t_{n+1}) = \theta_7 f(x_i, t_{n+1}), \qquad 0 \le i \le N.$$

System (33) contains $(N+1)$ linear equations with $(N+5)$ unknowns,

$$\left(c_{-2}, c_{-1}, c_0, \ldots, c_N, c_{N+1}, c_{N+2}\right)^T.$$

To have a unique solution to the above system, we are required the over-specified condition (28). Assume that $\kappa = x_s$, $1 \le s \le N-1$. Equation (6) holds and moreover

$$u(x_s, t) = h_1(t), \quad u_x(x_s, t) = h_2(t),$$

where $t \in [0, T]$. If we consider $m = s$ in (31), then we have

$$h_1(t_{n+1}) = c_{s-2}^{n+1} + 26c_{s-1}^{n+1} + 66c_s^{n+1} + 26c_{s+1}^{n+1} + c_{s+2}^{n+1},$$
$$h_2(t_{n+1}) = \frac{5}{h}\left(c_{s-2}^{n+1} + 10c_{s-1}^{n+1} - 10c_{s+1}^{n+1} - c_{s+2}^{n+1}\right),$$
$$g_1(t_{n+1}) = c_{N-2}^{n+1} + 26c_{N-1}^{n+1} + 66c_N^{n+1} + 26c_{N+1}^{n+1} + c_{N+2}^{n+1},$$
$$g_2(t_{n+1}) = \frac{5}{h}\left(c_{N-2}^{n+1} + 10c_{N-1}^{n+1} - 10c_{N+1}^{n+1} + c_{N+2}^{n+1}\right).$$

Consequently,

$$AC = B, \tag{34}$$

is a system of $(N+5)$ linear equations with $(N+5)$ unknown functions where

$$\begin{aligned}
&A[1,s+1] = A[1,s+5] = A[2,s+1] = A[N+4,N+5] = A[2,s+5]\\
&= A[N+4,N+1] = A[N+5,N+5] = A[N+5,N+1] = 1,\\
&A[N+5,N+2] = A[N+5,N+4] = A[1,s+2] = A[1,s+4] = 26,\\
&A[N+5,N+3] = A[1,s+3] = 66,\\
&A[N+4,N+2] = A[2,s+2] = -A[N+4,N+4] = -A[2,s+4] = 10,\\
&A[2,s+3] = A[N+4,N+3] = 0.
\end{aligned}$$

Thus, we can write the matrix A as

$$A = \begin{pmatrix}
0 & \dots & 1 & 26 & 66 & 26 & 1 & 0 & \dots \\
0 & \dots & 1 & 10 & 0 & -10 & 1 & 0 & \dots \\
A^* & B^* & C^* & D^* & E^* & 0 & \dots & \dots & \vdots \\
0 & A^* & B^* & C^* & D^* & E^* & 0 & \dots & \\
\vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
\vdots & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\
0 & \dots & & & A^* & B^* & C^* & D^* & E^* \\
0 & \dots & & & 1 & 10 & 0 & -10 & 1 \\
0 & \dots & & & 1 & 26 & 66 & 26 & 1
\end{pmatrix}_{(N+5)\times(N+5)},$$

with

$$C = \left(c_{-2}^{(n+1)}, c_{-1}^{(n+1)}, c_0^{(n+1)}, \dots, c_s^{(n+1)}, \dots, c_N^{(n+1)}, c_{N+1}^{(n+1)}, c_{N+2}^{(n+1)}\right)^T,$$

$$B = \left(B_{-2}^{(n)}, B_{-1}^{(n)}, B_0^{(n)}, \dots, B_s^{(n)}, \dots, B_N^{(n)}, B_{N+1}^{(n)}, B_{N+2}^{(n)}\right)^T$$

$$B_{-2}^{(n)} = h_1(t_{n+1}), \quad B_{-1}^{(n)} = \left(\frac{h}{5}\right) h_2(t_{n+1}), \quad B_i^{(n)} = H(x_i,t_n) + H(x_i,t_{n+1}),$$
$$B_{N+1}^{(n)} = \left(\frac{h}{5}\right) g_2(t_{n+1}), \quad B_{N+2}^{(n)} = g_1(t_{n+1}),$$

where $0 \leq i \leq N$. We note that A is an ill-conditioned matrix, so we obtain the solution of system (34) by using the Tikhonov regularization method.

3.2 The initial vector c^0

From the initial condition (4) the boundary and over-specified conditions (28) the initial vector c^0 can be obtained as the following expressions:

$$\begin{aligned}
u(x_s,t_0) &= c^0_{s-2}+26c^0_{s-1}+66c^0_s+26c^0_{s+1}+c^0_{s+2},\\
u_x(x_s,t_0) &= \frac{5}{h}\left(c^0_{s-2}+10c^0_{s-1}-10c^0_{s+1}+c^0_{s+2}\right)=h_2(t_0),\\
u(x_i,t_0) &= c^0_{i-2}+26c^0_{i-1}+66c^0_i+26c^0_{i+1}+c^0_{i+2}=p(x_i), \qquad 0\leq i\leq N,\\
u_x(x_N,t_0) &= \frac{5}{h}\left(c^0_{N-2}+10c^0_{N-1}-10c^0_{N+1}+c^0_{N+2}\right)=g_2(t_0),\\
u(x_N,t_0) &= c^0_{N-2}+26c^0_{N-1}+66c^0_N+26c^0_{N+1}+c^0_{N+2}=g_1(t_0).
\end{aligned}$$

This gives an $(N+5)\times(N+5)$-system of equations of the form of

$$AC^0=\beta, \tag{35}$$

where

$$\begin{aligned}
&A[1,s+1]=A[1,s+5]=A[2,s+1]=A[N+4,N+5]=A[2,s+5],\\
&=A[N+4,N+1]=A[N+5,N+5]=A[N+5,N+1]=1,\\
&A[N+5,N+2]=A[N+5,N+4]=A[1,s+2]=A[1,s+4]=26,\\
&A[N+5,N+3]=A[1,s+3]=66,\\
&A[N+4,N+2]=A[2,s+2]=-A[N+4,N+4]=-A[2,s+4]=10,\\
&A[2,s+3]=A[N+4,N+3]=0,
\end{aligned}$$

Thus, we have

$$A=\begin{pmatrix}
0 & \dots & 1 & 26 & 66 & 26 & 1 & 0 & \dots\\
0 & \dots & 1 & 10 & 0 & -10 & -1 & 0 & \dots\\
1 & 26 & 66 & 26 & 1 & 0 & \dots & \dots & \vdots\\
0 & 1 & 26 & 66 & 26 & 1 & 0 & \dots & \vdots\\
\vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
\vdots & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\
0 & & \dots & & 1 & 26 & 66 & 26 & 1\\
0 & & \dots & & 1 & 10 & 0 & -10 & 1\\
0 & & \dots & & 1 & 26 & 66 & 26 & 1
\end{pmatrix}_{(N+5)\times(N+5)},$$

$$C^0 = \begin{pmatrix} c^0_{-2} \\ c^0_{-1} \\ c^0_0 \\ \vdots \\ c^0_N \\ c^0_{N+1} \\ c^0_{N+2} \end{pmatrix}_{(N+5)\times 1}, \quad \beta = \begin{pmatrix} h_1(t_0) \\ (\frac{h}{5})h_2(t_0) \\ p(x_0) \\ \vdots \\ p(x_N) \\ (\frac{h}{5})g_2(t_0) \\ g_1(t_0) \end{pmatrix}_{(N+5)\times 1}.$$

Finally the solution of (35) can be obtained by using the Tikhonov regularization method.

3.3 Convergence analysis

Similar to the Convergence of the previous part, first we need to recall the following lemma.

Lemma 2. The B-splines $\{B_{-2}, B_{-1}, B_0, \ldots, B_N, B_{N+1}, B_{N+2}\}$ satisfy the following inequality:

$$\left| \sum_{j=-2}^{N+2} B_j(x) \right| \leq 186, \qquad (0 \leq x \leq 1). \tag{36}$$

Proof. We have from Table 2 at any node x_i, that

$$\left| \sum_{i=-2}^{N+2} B_i \right| \leq \sum_{i=-2}^{N+2} |B_i| = 120.$$

Similarly, we have for any point x in each sub-interval $[x_{i-1}, x_i]$ that

$$\sum_{j=-2}^{N+2} |B_j| \leq 186.$$

□

Theorem 4 ([6, 13])**.** Suppose that $f(x) \in C^6[0,1]$ and $|f^6(x)| \leq L$, for all $x \in [0,1]$. Assume that

$$\Delta = \{0 = x_0 < x_1 < \cdots < x_N = 1\},$$

is the partition of $[0,1]$ of step size h. If $S_\Delta(x)$ is the unique spline function interpolates $f(x)$ at nodes $x_0, x_1, \ldots, x_N \in \Delta$, then there exist a constant λ_j

such that we have for all $x \in [0,1]$ that

$$\left\| f^j(x) - S_\Delta^j(x) \right\|_\infty \leq \lambda_j L h^{6-j}, \qquad j = 0, 1, \ldots, 5.$$

where $\|\cdot\|$ represents the L^∞-norm.

By using the ideas of [20] and Theorem 4, there exists $W = W_L > 0$ such that

$$\begin{aligned}\left\| B - \widehat{B} \right\|_\infty &\leq W \left(\left| U(x) - \widehat{U}(x) \right| + \left| U^{'}(x) - \widehat{U}^{'}(x) \right| + \left| U^{''}(x) - \widehat{U}^{''}(x) \right| \right. \\ &\quad \left. + \left| U^{'''}(x) - \widehat{U}^{'''}(x) \right| + \left| U^{(4)}(x) - \widehat{U}^{(4)}(x) \right| \right) \\ &\leq WL\lambda_0 h^6 + WL\lambda_1 h^5 + WL\lambda_2 h^4 + WL\lambda_3 h^3 + WL\lambda_4 h^2.\end{aligned}$$

Thus we can get

$$\left\| B - \widehat{B} \right\|_\infty \leq W_1 h^2, \tag{37}$$

where $W_1 = WL\left(\lambda_0 h^4 + \lambda_1 h^3 + \lambda_2 h^2 + \lambda_3 h + \lambda_4\right)$. Since the matrix A is an ill-conditioned matrix, we have from the Tikhonov regularization solution that

$$\left(C - \widehat{C}\right) = \left[A^T A + \alpha \left(R^{(2)}\right)^T R^2\right]^{-1} A^T \left(B - \widehat{B}\right).$$

After taking the infinity-norm we find from (37) that

$$\left\| C - \widehat{C} \right\|_\infty \leq \left\| \left[A^T A + \alpha \left(R^{(2)}\right)^T R^2\right]^{-1} A^T \right\|_\infty \left\| B - \widehat{B} \right\|_\infty \leq W_2 h^2, \tag{38}$$

where

$$W_2 = W_1 \left\| [A^T A + \alpha (R^{(2)})^T R^2]^{-1} A^T \right\|_\infty.$$

Now we see that

$$U(x) - \widehat{U}(x) = \sum_{i=-2}^{N+2} (c_i - \hat{c}_i) B_i(x).$$

Using (36) and (38), we have after taking the L^∞-norm that

$$\begin{aligned}\left\| U(x) - \widehat{U}(x) \right\|_\infty = \left\| \sum_{i=-2}^{N+2} (c_i - \hat{c}_i)\, B_i(x) \right\|_\infty &\leq \|(c_i - \hat{c}_i)\|_\infty \left| \sum_{i=-2}^{N+2} B_i(x) \right| \\ &\leq 186 W_2 h^2,\end{aligned} \tag{39}$$

Theorem 5. Let $u(x,t) \in C^6[0,1]$ be the exact solution of (1) such that

$$\left|\frac{\partial^6 u(x,t)}{\partial x^6}\right| \leq L.$$

Assume that $U(x,t)$ is the numerical approximation by our methods, then

$$\|u(x) - U(x)\|_\infty \leq O(k + h^2).$$

Proof. We have from Theorem 4 that

$$\|u(x) - U(x)\|_\infty \leq \lambda_0 L h^6, \tag{40}$$

Equations (39) and (40) imply that

$$\begin{aligned}\|u(x) - U(x)\|_\infty &\leq \left\|u(x) - \widehat{U}(x)\right\|_\infty + \left\|\widehat{U}(x) - U(x)\right\|_\infty \\ &\leq \lambda_0 L h^6 + 186 W_2 h^2 = \Upsilon h^2,\end{aligned}$$

where $\Upsilon = \lambda_0 L h^4 + 186 W_2$ and $U(x,t)$ is the approximate solution by our present method. Then by Theorem 2 and the above considerations, we have

$$\|u(x) - U(x)\|_\infty \leq \rho(k + h^2),$$

where ρ is a finite positive constant independent of k and h. This completes the proof.

□

3.4 Stability analysis

Similar to the previous section, we use the Von-Neuman stability analysis. We assume again in equation (1) that $f(x,y) = 0$. On the other hand, as we have linearized the nonlinear term uu_x by considering u as a constant k_1 in the equation (1), then the equation can be rewritten by

$$(u_t + b u_{xxx} + k_1 u_x - a u_{xx})_x = \gamma u. \tag{41}$$

If we substitute the approximation (11) into the equation (41) and discretize it in time, we obtain that

$$\begin{aligned}-P_1 u^{n+1} + u_x^{n+1} + P_2 u_{xx}^{n+1} - P_3 u_{xxx}^{n+1} + P_4 u_{xxxx}^{n+1} \\ = P_1 u^n + u_x^n - P_2 u_{xx}^n + P_3 u_{xxx}^n - P_4 u_{xxxx}^n.\end{aligned}$$

If $\frac{k}{2} = \tau$, then we have

$$P_1 = \tau\gamma, \quad P_3 = \frac{60}{h^3}\tau a, \quad P_2 = \frac{20}{h^2}\tau k_1, \quad P_4 = \frac{120}{h^4}\tau b. \tag{42}$$

In terms of unknown time parameters c_i, the equation can be written from (31) by

$$\begin{aligned} &A^* c_{i-2}^{n+1} + B^* c_{i-1}^{n+1} + C^* c_i^{n+1} + D^* c_{i+1}^{n+1} + E^* c_{i+2}^{n+1} \\ &= F^* c_{i-2}^{n} + G^* c_{i-1}^{n} + H^* c_i^{n} + I^* c_{i+1}^{n+1} + J^* c_{i+2}^{n+1}, \end{aligned}$$

where

$$\begin{aligned} A^* &= -P_1 - 1 + P_2 - P_3 + P_4, \\ B^* &= -26P_1 + 10 + 2P_2 + 2P_3 - 4p_4, \\ C^* &= -66P_1 - 6P_2 + 6P_4, \\ D^* &= -26P_1 - 10 + 2P_2 - 2P_3 - 4P_4, \\ E^* &= -P_1 - 1 + P_2 + P_3 + P_4, \\ F^* &= P_1 + 1 - P_2 + P_3 - P_4, \\ G^* &= 26P_1 + 10 - 2P_2 - 2P_3 + 4P_4, \\ H^* &= 66P_1 + 6P_2 - 6P_4, \\ I^* &= 26P_1 - 10 - 2P_2 + 2P_3 + 4P_4, \\ J^* &= P_1 - 1 - P_2 - P_3 - P_4. \end{aligned} \tag{43}$$

If we substitute $c_m^n = A\zeta^n \exp(im\beta h)$, where h is step length, A is amplitude, $i = \sqrt{-1}$, and β is mode number, then we get that

$$\zeta = \frac{X_1 + iY_1}{X_2 + iY_2},$$

where

$$\begin{aligned} X_1 &= (F^* + J^*)\cos(2\beta h) + (G^* + I^*)\cos(\beta h) + H^*, \\ X_2 &= (A^* + E^*)\cos(2\beta h) + (B^* + D^*)\cos(\beta h) + C^*, \\ Y_1 &= (J^* - F^*)\sin(2\beta h) + (I^* - G^*)\sin(\beta h), \\ Y_2 &= (E^* - A^*)\sin(2\beta h) + (D^* - B^*)\sin(\beta h). \end{aligned}$$

We have from (43) that $X_2 = -X_1$ and

$$\begin{aligned} Y_1 &= \sin(2\beta h)(-2 - 2P_3) + \sin(\beta h)(-20 + 4P_3), \\ Y_2 &= \sin(2\beta h)(-2 + 2P_3) + \sin(\beta h)(-20 - 4P_3). \end{aligned}$$

To show the stability, we need to prove $|\zeta| < 1$, that is,

$$X_1^2 + Y_1^2 < X_1^2 + Y_2^2 \tag{44}$$

We prove that $Y_1^2 - Y_2^2 < 0$, but we see that

$$Y_1^2 - Y_2^2 = 256P_3 \sin^2(\beta h)(\cos(\beta h) - 1)(\cos(\beta h) + 5).$$

As all arbitrary constants are positive, then $P_3 > 0$. Moreover, the functions $\sin^2(\beta h)$ and $\cos(\beta h) + 5$ are positive but $\cos(\beta h) - 1$ is negative, so $Y_1^2 - Y_2^2 < 0$. Hence, we obtain that the quintic B-spline collocation method is unconditionally stable.

4 Numerical result and illustrations

In this section, we will give some examples to investigate the applicability of our methods that we described in the previous sections. In general, there are two error sources for an inverse problem. The first one is the unavoidable bias deviation, known as deterministic error, and the second one is the variance caused by the enhancement of error estimation, known as stochastic error. The global impact of deterministic and stochastic errors is studied in the mean squared error or the total error as in [3]. Next, by considering the total error S defined by

$$S_{f_1} = \left[\frac{1}{N-1}\sum_{s=1}^{N}(f_1(t_{s+1}) - f_1^*(t_{s+1}))^2\right]^{\frac{1}{2}},$$

$$S_{f_2} = \left[\frac{1}{N-1}\sum_{s=1}^{N}(f_2(t_{s+1}) - f_2^*(t_{s+1}))^2\right]^{\frac{1}{2}},$$

we will compare the exact and the approximate solutions, where f_1 and f_2 are exact values, f_1^* and f_2^* are the estimated values, and the number of estimated values is denoted by N.

We should remark that there are no previous numerical results related to the nonlinear inverse problem (1) and (4)–(6) in the literature to compare our results, and one may use the other numerical methods to study this problem.

Example 1. In our first example, we consider the nonlinear inverse problem (1) and (4)–(6), where $a = 1$, $b = 5$, and $\gamma = 3$ with the initial data

$$u(x, 0) = \sin(x),$$

and the external force

$$f(x,t) = \cos(t+x) + \frac{3}{2}\sin(t+x) + \cos^2(t+x) - \sin^2(t+x).$$

An exact solutions of this problem is $u(x,t) = \sin(x+t)$ with

$$u(0,t) = f_1(t) = \sin(t), \quad u_x(0,t) = f_2(t) = \cos(t).$$

Table 3: Comparison between exact and numerical solutions of Example 1 for $u(0.1,t)$, $|u(0.1,t) - u^*(0.1,t)|$, by the Quintic and Septic B-spline methods with $N = 30, 50, 100$.

	N=30		N=50		N=100	
t	Quintic	Septic	Quintic	Septic	Quintic	Septic
0.1	$4.8e-04$	$4.2e-06$	$8.7e-04$	$6.5e-06$	$1.4e-03$	$8.5e-06$
0.2	$7.3e-04$	$4.4e-06$	$9.0e-04$	$4.9e-06$	$1.8e-03$	$3.7e-08$
0.3	$1.0-03$	$1.6e-06$	$1.7e-03$	$1.9e-06$	$2.3e-03$	$2.0e-06$
0.4	$1.4e-03$	$1.5e-05$	$2.2e-03$	$6.9e-06$	$3.0e-03$	$5.5e-06$
0.5	$1.7e-03$	$1.0e-05$	$2.7e-03$	$1.8e-08$	$3.5e-03$	$5.3e-08$
0.6	$1.9e-03$	$1.1e-05$	$2.9e-03$	$3.2e-06$	$3.8e-03$	$3.3e-06$
0.7	$2.3e-03$	$1.8e-05$	$3.3e-03$	$4.1e-05$	$4.3e-03$	$2.2e-06$
0.8	$2.6e-03$	$3.0e-06$	$3.7e-03$	$2.0e-06$	$4.7e-03$	$3.6e-06$
0.9	$2.8e-03$	$2.4e-06$	$3.9e-03$	$3.6e-06$	$4.9e-03$	$1.4e-07$
1	$3.0e-03$	$1.3e-06$	$4.2e-03$	$2.0e-06$	$5.2e-03$	$2.1e-06$
S_u	$5.9758e-05$	$1.0042e-06$	$8.7071e-05$	$1.0093e-06$	$1.1311e-04$	$1.0143e-06$

Table 4: Comparison between exact and numerical solutions of Example 1 for $f_1(t)$, $|f_1(t) - f_1^*(t)|$, by employing the Quintic and Septic B-spline methods with $N = 30, 50, 100$.

	N=30		N=50		N=100	
t	Quintic	Septic	Quintic	Septic	Quintic	Septic
0.1	$0.9e-04$	$1.9e-05$	$1.1e-04$	$5.9e-05$	$1.2e-04$	$1.9e-06$
0.2	$2.4e-04$	$2.2e-05$	$2.4e-04$	$3.5e-06$	$2.8e-04$	$7.3e-08$
0.3	$3.9e-04$	$3.1e-05$	$3.4e-04$	$8.5e-08$	$4.6e-04$	$5.0e-08$
0.4	$6.4e-04$	$3.4e-05$	$5.1e-04$	$9.4e-06$	$6.6e-04$	$1.4e-07$
0.5	$3.2e-04$	$2.8e-05$	$6.2e-04$	$1.3e-05$	$4.0e-04$	$2.0e-07$
0.6	$1.0e-03$	$2.7e-05$	$7.2e-04$	$3.6e-06$	$4.6e-04$	$1.8e-07$
0.7	$1.2e-03$	$2.3e-05$	$8.7e-04$	$4.7e-06$	$5.2e-04$	$4.5e-08$
0.8	$1.3e-03$	$2.2e-05$	$9.6e-04$	$2.4e-07$	$5.6e-04$	$3.3e-08$
0.9	$1.4e-03$	$1.4e-05$	$1.0e-03$	$3.7e-06$	$6.1e-04$	$4.3e-08$
1	$1.6e-03$	$1.5e-06$	$1.1e-03$	$3.7e-06$	$6.5e-04$	$4.6e-07$
S_{f_1}	$3.1833e-05$	$1.1633e-06$	$2.3106e-05$	$1.1575e-06$	$1.3381e-05$	$9.8944e-07$

Tables 3–5 show the total error S for some values of N for each method. In Figures 1 and 2, we plotted the total error S as a function of N. Also we took $\kappa = N^{-1}$, $T = 1$, $\Delta t = 0.001$, and the noisy data (input data + $0.001 \times \text{rand}(1)$).

Table 5: Comparison between exact and numerical solutions of Example 1 for $f_2(t)$, $|f_2(t) - f_2^*(t)|$, by the Quintic and Septic B-spline methods with $N = 30, 50, 100$.

	N=30		N=50		N=100	
t	Quintic	Septic	Quintic	Septic	Quintic	Septic
0.1	$4.9e-03$	$1.9e-03$	$3.6e-04$	$6.3e-03$	$1.2e-02$	$7.9e-06$
0.2	$1.2e-02$	$2.3e-03$	$8.1e-03$	$4.9e-03$	$1.9e-02$	$5.2e-04$
0.3	$1.7e-02$	$3.5e-03$	$1.3e-02$	$3.7e-03$	$2.6e-02$	$4.2e-04$
0.4	$2.3e-02$	$4.1e-03$	$2.2e-02$	$2.5e-03$	$3.3e-02$	$1.0e-04$
0.5	$2.8e-02$	$3.7e-03$	$2.8e-02$	$2.4e-03$	$4.0e-02$	$1.5e-04$
0.6	$3.4e-02$	$4.0e-03$	$3.4e-02$	$9.1e-04$	$4.6e-02$	$8.3e-05$
0.7	$3.9e-02$	$3.6e-03$	$4.1e-02$	$1.2e-03$	$5.2e-02$	$4.6e-04$
0.8	$4.3e-02$	$3.9e-03$	$4.6e-02$	$3.7e-04$	$5.6e-02$	$1.7e-05$
0.9	$4.7e-02$	$3.3e-03$	$5.1e-02$	$3.1e-04$	$6.1e-02$	$5.1e-05$
1	$5.2e-02$	$3.5e-03$	$5.5e-02$	$3.0e-04$	$6.5e-02$	$2.3e-05$
S_{f_2}	$9.6783e-04$	$7.7698e-05$	0.0012	$8.3921e-05$	0.0013	$1.5842e-06$

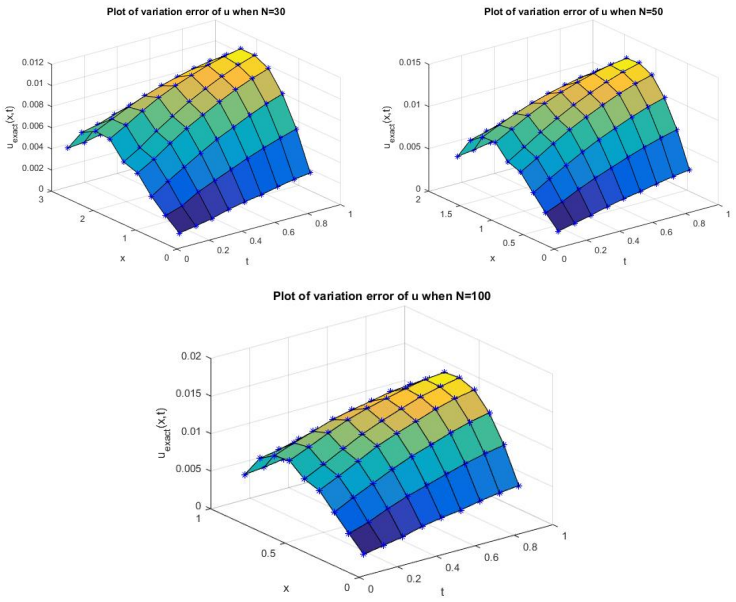


Figure 1: Plots of variation $u(x,t)$ of Example 1 by using the Quintic B-spline method when $N = 30, 50, 100$.

Example 2. In this example, we consider the inverse problem (1) and (4)–(6), where $a = 1$, $b = 3$ and $\gamma = 0.5$. The initial data is

$$u(x,0) = \operatorname{sech}(x),$$

while

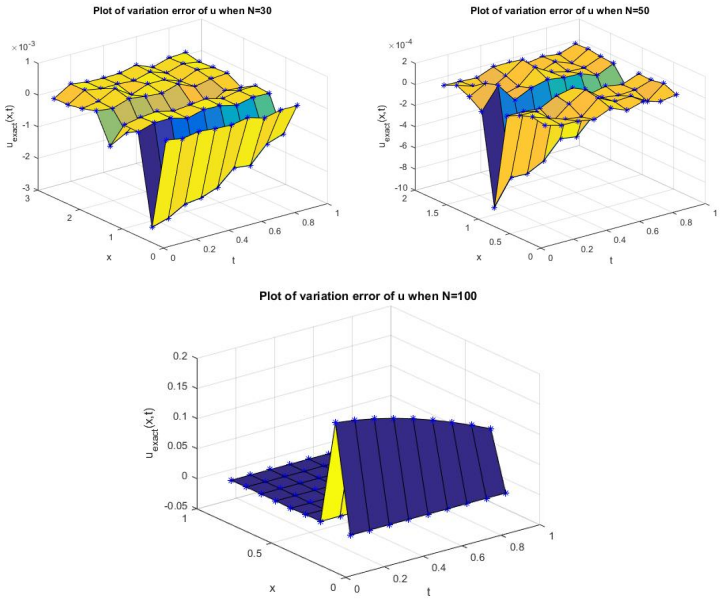


Figure 2: Plots of variation $u(x,t)$ of Example 1 by using the septic B-spline method when $N = 30, 50, 100$.

$$f(x,t) = -\frac{142\sinh(t-x)^2}{\cosh(t-x)^3} + \frac{\sinh(t-x)^2 - 6\sinh(t-x)^3 + 2\sinh(t-x)^2}{\cosh(t-x)^4} + \frac{120\sinh(t-x)^4}{\cosh(t-x)^5} + \frac{23}{\cosh(t-x)} + \frac{5\sinh(t-x)-1}{\cosh(t-x)^2},$$

is the external force. An exact solution of this problem is $u(x,t) = \operatorname{sech}(x-t)$ such that

$$u(0,t) = f_1(t) = \operatorname{sech}(-t),$$
$$u_x(0,t) = f_2(t) = \frac{\sinh(t)}{\cosh(t^2)}.$$

Tables 6 and 7 give the numerical results for conditions $u(0,t)$ and $u_x(0,t)$, respectively. To explain the accuracy of our methods, some illustrations are presented in Figures 3–6. Table 8 shows the numerical values of $u(x,t)$ at point $x = 0.1$. Figures 7 and 8 also present the comparison between the numerical and exact values of $u(x,t)$. Also we take $\kappa = 0.03, T = 1, \Delta t = 0.001$ and the noisy data (input data + $0.001 \times \text{rand}(1)$).

Table 7: Comparison between exact and numerical solutions of Example 2 for $f_2(t)$, $|f_2(t) - f_2^*(t)|$, by the Quintic and Septic B-spline methods with $N = 100$.

	Quintic B-spline			Septic B-spline		
t	$f_2(t)$	$f_2^*(t)$	$\|f_2(t) - f_2^*(t)\|$	$f_2(t)$	$f_2^*(t)$	$\|f_2(t) - f_2^*(t)\|$
0.1	0.099172	0.021154	$7.801727e-02$	0.099172	0.102857	$3.685425e-03$
0.2	0.193493	0.118409	$7.508375e-02$	0.193493	0.197758	$4.265252e-03$
0.3	0.278678	0.208712	$6.996601e-02$	0.278678	0.282586	$3.907888e-03$
0.4	0.351456	0.292363	$5.909310e-02$	0.351456	0.355666	$4.210337e-03$
0.5	0.409814	0.356937	$5.287678e-02$	0.409814	0.413920	$4.105471e-03$
0.6	0.453029	0.414228	$3.880057e-02$	0.453029	0.457373	$4.344355e-03$
0.7	0.481503	0.454315	$2.718771e-02$	0.481503	0.486087	$4.583750e-03$
0.8	0.496500	0.479423	$1.707765e-02$	0.496500	0.500676	$4.175531e-03$
0.9	0.499829	0.495907	$3.922220e-03$	0.499829	0.500025	$1.958806e-04$
1	0.493554	0.499845	$6.290468e-03$	0.493554	0.492724	$8.303327e-04$
S_{f_2}	-	-	0.0018	-	-	$1.3344e-04$
Execution time (second)	40.079962			45.367252		
Condition Number of Matrix A	$1.4332e+20$			$4.1816e+19$		

Table 6: Comparison between exact and numerical solutions of Example 2 for $f_1(t)$, $|f_1(t) - f_1^*(t)|$, by the Quintic and Septic B-spline methods with $N = 100$.

	Quintic B-spline			Septic B-spline		
t	$f_1(t)$	$f_1^*(t)$	$\|f_1(t) - f_1^*(t)\|$	$f_1(t)$	$f_1^*(t)$	$\|f_1(t) - f_1^*(t)\|$
0.1	0.995021	0.997416	$2.394818e-03$	0.995021	0.995007	$1.369526e-05$
0.2	0.980328	0.982549	$2.220868e-03$	0.980328	0.980312	$1.635728e-05$
0.3	0.956628	0.958612	$1.984447e-03$	0.956628	0.956613	$1.456806e-05$
0.4	0.925007	0.926788	$1.780935e-03$	0.925007	0.924991	$1.596319e-05$
0.5	0.886819	0.888307	$1.488184e-03$	0.886819	0.886804	$1.534899e-05$
0.6	0.843551	0.844734	$1.183566e-03$	0.843551	0.843534	$1.641813e-05$
0.7	0.796705	0.797546	$8.406108e-04$	0.796705	0.796696	$9.782520e-06$
0.8	0.747700	0.748193	$4.927829e-04$	0.747700	0.747694	$6.334825e-06$
0.9	0.697795	0.698057	$2.627784e-04$	0.697795	0.697795	$3.326556e-08$
1	0.648054	0.647956	$9.857496e-05$	0.648054	0.648057	$2.488186e-06$
S_{f_1}	-	-	$5.0435e-05$	-	-	$5.0706e-07$
Execution time(second)	40.079962			45.367252		
Condition Number of Matrix A	$1.4332e+20$			$4.1816e+19$		

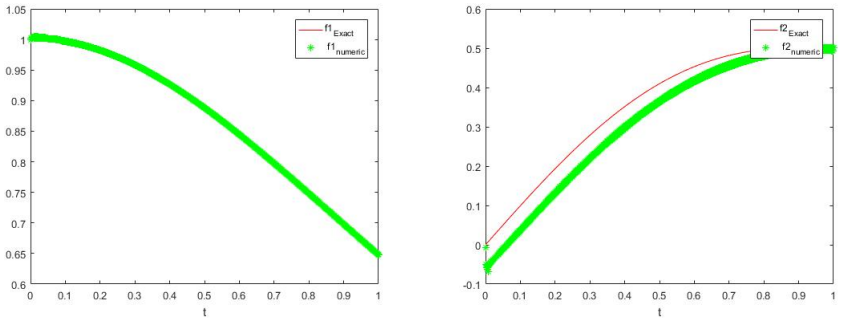


Figure 3: Comparison between exact and numerical values of (a) $u(0,t)$ and (b) $u_x(0,t)$ of Example 2 by the Quintic B-spline method.

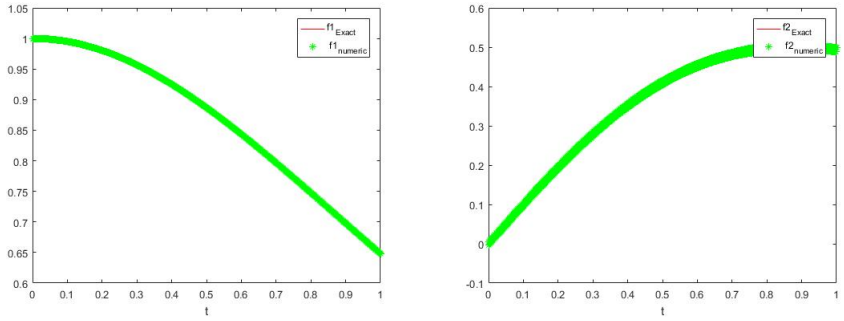


Figure 4: Comparison between exact and numerical values of (a) $u(0,t)$ and (b) $u_x(0,t)$ of Example 2 by the septic B-spline method.

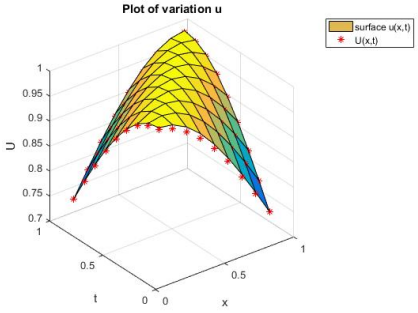


Figure 5: Comparison between exact and numerical values of $u(x,t)$ of Example 2 by the quintic B-spline method.

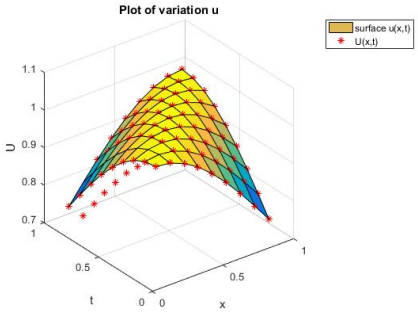


Figure 6: Comparison between exact and numerical values of $u(x,t)$ of Example 2 by the septic B-spline method.

Table 8: Comparison between exact and numerical values of $u(0.1,t)$ of Example 2 by the Quintic and Septic B-spline methods with $N = 100$.

	Quintic B-spline			Septic B-spline		
t	$u(0.1,t)$	$u^*(0.1,t)$	$\lvert u(0.1,t)-u^*(0.1,t)\rvert$	$u(0.1,t)$	$u^*(0.1,t)$	$\lvert u(0.1,t)-u^*(0.1,t)\rvert$
0.1	1.000000	0.996042	$3.958350e-03$	1.000000	1.000008	$8.493153e-06$
0.2	0.995021	0.991131	$3.889610e-03$	0.995021	0.994971	$4.942389e-05$
0.3	0.980328	0.976786	$3.542139e-03$	0.980328	0.980355	$2.729571e-05$
0.4	0.956628	0.953011	$3.616572e-03$	0.956628	0.956594	$3.413866e-05$
0.5	0.925007	0.922351	$2.656163e-03$	0.925007	0.925047	$3.991245e-05$
0.6	0.886819	0.884390	$2.429210e-03$	0.886819	0.886839	$1.991189e-05$
0.7	0.843551	0.841742	$1.808390e-03$	0.843551	0.843521	$2.948822e-05$
0.8	0.796705	0.795713	$9.927146e-04$	0.796705	0.796733	$2.729211e-05$
0.9	0.747700	0.747000	$7.002802e-04$	0.747700	0.747675	$2.541404e-05$
1	0.697795	0.697668	$1.267156e-04$	0.697795	0.697777	$1.739320e-05$
S_u	-	-	$9.9114e-05$	-	-	$8.2254e-07$

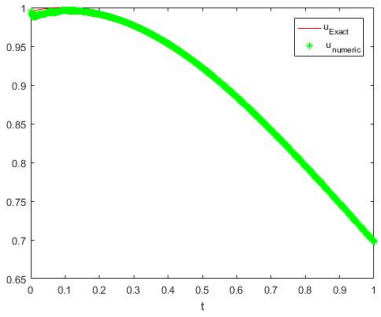


Figure 7: Plot of variation $u(0.1,t)$ of Example 2 by the Quintic B-spline method with $N = 100$.

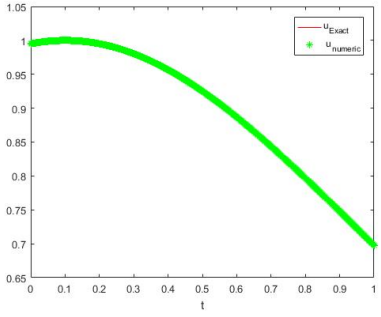


Figure 8: Plot of variation $u(0.1,t)$ of Example 2 by the septic B-spline method with $N = 100$.

Example 3. In our last example, we solve the inverse problem (4) and (4)–(6), where $a = 1$, $b = 2$, $\gamma = 0.5$ and the initial data is given by

$$u(x,0) = \cos(x),$$

The force function is

$$f(x,t) = \frac{5\cos(t-x)}{2} + \sin(t-x) - \cos(t-x)^2 + \sin(t-x)^2,$$

The function $u(x,t) = \cos(x-t)$, is an exact solution of this problem. Thus $u(0,t) = f_1(t) = \cos(-t)$ and $u(1,t) = f_2(t) = -\sin(-t)$. Tables 9 and 10 give the numerical results for $u(0,t)$ and $u_x(0,t)$, respectively. To explain the accuracy of our methods, some illustrations are presented in Figures 9–12. Table 11 shows the numerical values of $u(x,t)$ at point $x = 0.1$. Figures 13 and 14 also present the comparison between the numerical and exact values of solution $u(x,t)$. Also we take $\kappa = 0.03$, $T = 1$, $\Delta t = 0.001$, and the noisy data (input data+$0.001 \times$ rand(1)).

Table 9: Comparison between exact and numerical values for $f_1(t)$ of Example 3 by applying the Quintic and Septic B-spline methods with $N = 100$.

	Quintic B-spline			Septic B-spline		
t	$f_1(t)$	$f_1^*(t)$	$\lvert f_1(t) - f_1^*(t)\rvert$	$f_1(t)$	$f_1^*(t)$	$\lvert f_1(t) - f_1^*(t)\rvert$
0.1	0.995004	0.996818	$1.813881e-03$	0.995004	0.994960	$4.446296e-05$
0.2	0.980067	0.981816	$1.749680e-03$	0.980067	0.980097	$3.086339e-05$
0.3	0.955336	0.957039	$1.702169e-03$	0.955336	0.955366	$2.917672e-05$
0.4	0.921061	0.922675	$1.613838e-03$	0.921061	0.921150	$8.917805e-05$
0.5	0.877583	0.879143	$1.560911e-03$	0.877583	0.877252	$3.307205e-04$
0.6	0.825336	0.826760	$1.424698e-03$	0.825336	0.824996	$3.394589e-04$
0.7	0.764842	0.766191	$1.349309e-03$	0.764842	0.764432	$4.104127e-04$
0.8	0.696707	0.697931	$1.224282e-03$	0.696707	0.696210	$4.966064e-04$
0.9	0.621610	0.622693	$1.082930e-03$	0.621610	0.621069	$5.406920e-04$
1	0.540302	0.541248	$9.459846e-04$	0.540302	0.539825	$4.770495e-04$
S_{f_1}	-	-	$4.6373e-05$	-	-	$3.8840e-06$
Execution time(second)	40.562310			41.030530		
Condition Number of Matrix A	$1.4332e+20$			$1.3648e+19$		

Table 10: Comparison between exact and numerical values for $f_2(t)$ of Example 3 by applying the Quintic and Septic B-spline methods with $N = 100$.

	Quintic B-spline			Septic B-spline		
t	$f_2(t)$	$f_2^*(t)$	$\lvert f_2(t) - f_2^*(t)\rvert$	$f_2(t)$	$f_2^*(t)$	$\lvert f_2(t) - f_2^*(t)\rvert$
0.1	0.099833	0.038846	$6.098711e-02$	0.099833	0.103929	$4.095972e-03$
0.2	0.198669	0.139510	$5.915962e-02$	0.198669	0.194482	$4.186958e-03$
0.3	0.295520	0.238522	$5.699784e-02$	0.295520	0.292183	$3.337169e-03$
0.4	0.389418	0.335087	$5.433175e-02$	0.389418	0.378289	$1.112973e-02$
0.5	0.479426	0.427911	$5.151492e-02$	0.479426	0.466016	$1.340928e-02$
0.6	0.564642	0.516943	$4.769963e-02$	0.564642	0.546380	$1.826272e-02$
0.7	0.644218	0.599703	$4.451437e-02$	0.644218	0.692126	$4.790832e-02$
0.8	0.717356	0.677664	$3.969196e-02$	0.717356	0.774241	$5.688534e-02$
0.9	0.783327	0.748561	$3.476550e-02$	0.783327	0.846131	$6.280393e-02$
1	0.841471	0.810553	$3.091790e-02$	0.841471	0.902065	$6.059395e-02$
S_{f_2}	-	-	0.0016	-	-	$5.1323e-04$
Execution time (second)	40.562310			41.030530		
Condition Number of Matrix A	$1.4332e+20$			$1.3648e+19$		

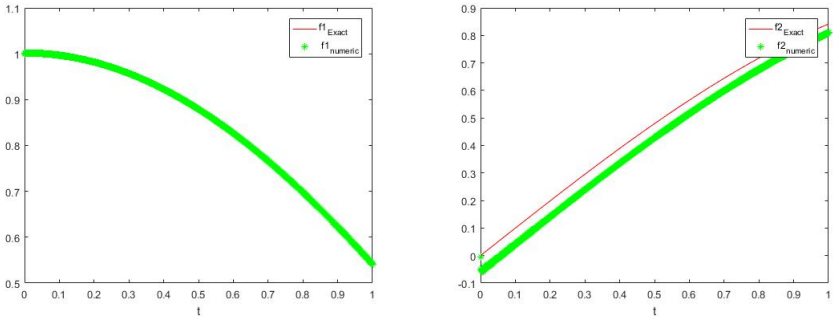


Figure 9: Comparison between exact and numerical values of (a) $u(0,t)$ and (b) $u_x(0,t)$ of Example 3 by the Quintic B-spline method.

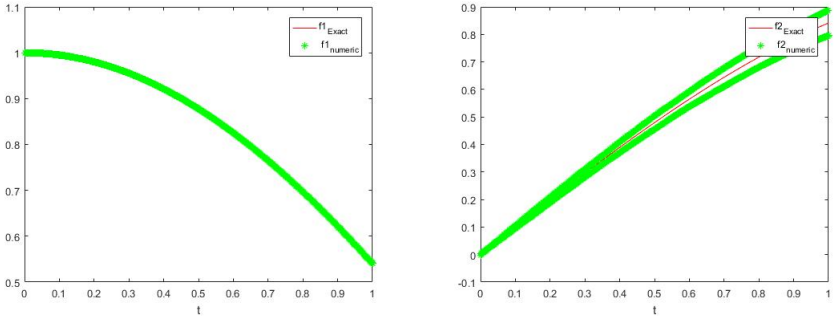


Figure 10: Comparison between exact and numerical values of (a) $u(0,t)$ and (b) $u_x(0,t)$ of Example 3 by the septic B-spline method.

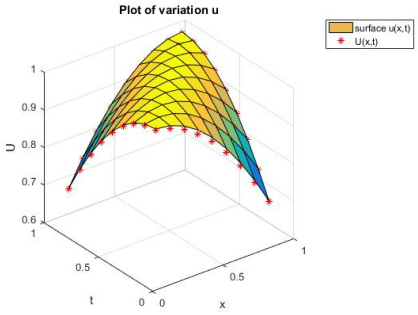


Figure 11: Comparison between exact and numerical values of $u(x,t)$ of Example 3 by the Quintic B-spline method.

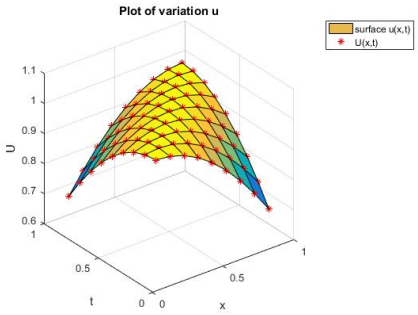


Figure 12: Comparison between exact and numerical values of $u(x,t)$ of Example 3 by the septic B-spline method.

Table 11: Comparison between exact and numerical values of $u(0.1,t)$ of Example 3 by the Quintic and Septic B-spline methods with $N = 100$.

	Quintic B-spline			Septic B-spline		
t	$u(0.1,t)$	$u^*(0.1,t)$	$\lvert u(0.1,t)-u^*(0.1,t)\rvert$	$u(0.1,t)$	$u^*(0.1,t)$	$\lvert u(0.1,t)-u^*(0.1,t)\rvert$
0.1	1.000000	0.996544	$3.455663e-03$	1.000000	0.999902	$9.781401e-05$
0.2	0.995004	0.991605	$3.399369e-03$	0.995004	0.995019	$1.462144e-05$
0.3	0.980067	0.976735	$3.331920e-03$	0.980067	0.980016	$5.030518e-05$
0.4	0.955336	0.952079	$3.257457e-03$	0.955336	0.955342	$5.215045e-06$
0.5	0.921061	0.917956	$3.104844e-03$	0.921061	0.921182	$1.214016e-04$
0.6	0.877583	0.874601	$2.981736e-03$	0.877583	0.877510	$7.290696e-05$
0.7	0.825336	0.822554	$2.781252e-03$	0.825336	0.825410	$7.484661e-05$
0.8	0.764842	0.762264	$2.577809e-03$	0.764842	0.764753	$8.910966e-05$
0.9	0.696707	0.694359	$2.347544e-03$	0.696707	0.696628	$7.884439e-05$
1	0.621610	0.619528	$2.082224e-03$	0.621610	0.621652	$4.249717e-05$
S_u	-	-	$9.5839e-05$	-	-	$2.8097e-06$

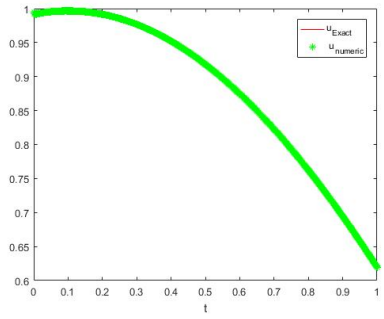


Figure 13: Plot of variation $u(0.1,t)$ in Example 3 by the Quintic B-spline method with $N = 100$.

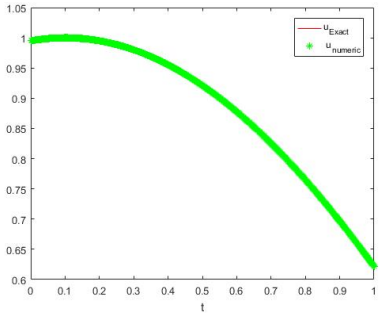


Figure 14: Plot of variation $u(0.1, t)$ in Example 3 by the Septic B-spline method with $N = 100$.

5 Conclusion

In this paper, we have successfully employed the septic B-spline and the quintic B-spline method to estimate unknown boundary conditions in an inverse problem related to the Ostrovsky–Burgers equation (1) and (4)–(6). We have discussed the convergence rate of our methods and have shown the rate of the septic B-spline method is $O(k + h^4)$, while $O(k + h^2)$ is the convergence rate for the quintic B-spline method. The stability of both methods was investigated. By comparing the numerical results, we showed that the accuracy and stability of the septic B-spline method are more than the ones for the quintic B-spline method. Since the associated coefficient matrix in the septic B-spline method and quintic B-spline method are usually ill-conditioned, we have used the Tikhonov regularization method to obtain a stable numerical approximation of the solution. The results are collected by using the MATLAB 7.10(R2016a), tested on a personal computer with Intel(R) Core(TM)2 Duo-CPU and 4-GB RAM.

References

1. Akram, G. and Siddiqi, S.S. *End conditions for interpolatory septic splines*, Int. J. Comput. Math. 82 (2005) 1525–1540.

2. Bhatta, D. *Use of modified Bernstein polynomials to solve KdV–Burgers equation numerically,* Appl. Math. Comput. 206 (2008) 457–464.

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

4. Chen, G.U. and Boyd, J.P. *Analytical and numerical studies of weakly nonlocal solitary waves of the rotation-modified Korteweg–de Vries equation,* Physica D 155 (2001) 201–222.

5. Chen, R. and Wu, Z. *Solving partial differential equation by using multiquadric quasi-interpolation,* Appl. Math. Comput. 186 (2007) 1502–10.

6. de Boor, C. *On the convergence of odd degree spline interpolation,* J. Approx. Theory 1 (1968) 452–463.

7. Esfahani, A. and Levandosky, S. *Solitary waves of the rotation-generalized Benjamin-Ono equation,* Discrete Contin. Dyn. Syst. 33 (2013) 663–700.

8. Foadian, S., Pourgholi, R. and Tabasi, S.H. *Cubic B-spline method for the solution of an inverse parabolic system,* Appl. Anal. 97 (2018) 438–465.

9. Galkin, V.N. and Stepanyants, Y.A. *On the existence of stationary solitary waves in a rotating field,* Appl. Math. Mech. 55 (1991) 939–943.

10. Gilman, O.A., Grimshaw, R. and Stepanyants, Y.A. *Approximate and numerical solutions of the stationary Ostrovsky equation,* Stud. Appl. Math. 95 (1995) 115–126

11. Grimshaw, R. *Internal solitary waves, environmental stratified flows*, Kluwer Academic Publishers, Dordrecht, 2001, Ch. 1, pp. 1–29.

12. Haq, S., Siraj-Ul-Islam, and Uddin, M. *A mesh-free method for the numerical solution of the KdV-Burgers equation,* Appl. Math. Model. 33 (2009) 3442–3449.

13. Hall, C.A. *On error bounds for spline interpolation,* J. Approx. Theory 1 (1968) 209–218.

14. Helal, M.A. and Mehanna, M.S. *A comparison between two different methods for solving KdV–Burgers equation,* Chaos Soliton Fract. 28 (2006) 320–326.

15. Khater, A.H., Temsah, R.S. and Hassan, M.M. *A Chebyshev spectral collocation method for solving Burgers'-type equations,* J. Comput. Appl. Math. 222 (2008) 333–350.

16. Mitsudera, H. and Grimshaw, R. *Effects of friction on a localized structure in a baroclinic current,* J. Physical Oceanography 23 (1993) 2265–2292.

17. Obregon, M.A. and Stepanyants, Y.A. *Oblique magneto-acoustic solitons 460 in rotating plasma,* Phys. Lett. A 249 (1998) 315–323.

18. Ostrovsky, L.A. *Nonlinear internal waves in a rotating ocean*, Oceanologia 18 (1978) 181–191

19. Ott, E. and Sudan, R.N. *Damping of solitary waves,* Phys. Fluids 13 (1970) 1432–1434.

20. Rudin, W. *Principles of mathematical analysis,* Third edition, International Series in Pure and Applied Mathematics, McGraw-Hill Book Co., New York-Auckland-Düsseldorf, 1976.

21. Smith, G.D. *Numerical solution of partial differential equation: finite difference method,* Learendom Press, Oxford 1978.

22. Wang, H. and Esfahani, A. *Well-posedness and asymptotic behavior of the dissipative Ostrovsky equation,* Evol. Equ. Control Theory 8 (2019) 709–735.

23. Xu, Y. and Shu, C-W. *Local discontinuous Galerkin methods for the Kuramoto Sivashinsky equations and the Ito-type coupled equations,* Comput. Methods Appl. Mech. 195 (2006) 3430–3447.

24. Zhan, J.M. and Li, Y.S. *Generalized finite spectral method for 1D burgers and KdV equations,* Appl. Math. Mech. (English Ed.) 27 (2006) 1635–1643.

How to cite this article
Ghanadian, F., Pourgholi, R., Tabasi, S. Numerical approximation for inverse problem of the Ostrovsky-Burgers equation. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 73-109. doi: 10.22067/ijnao.2021.70478.1036.

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 111-129
DOI:10.22067/ijnao.2021.71036.1044
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Preconditioned global GPBiCG method for solving saddle point problems with multiple right-hand sides and its convergence analysis

A. Taherian and F. Toutounian*

Abstract

We propose the preconditioned global generalized product-type method based on the preconditioned global BiCG method to solve nonsymmetric saddle point problems with multiple right-hand sides. We apply an indefinite preconditioner to enhance the convergence rate of the method. We also present some theoretical analysis and discuss the convergence of the PGl-GPBiCG method. Some useful properties of the preconditioned matrix are established. Moreover, we present the bounds for the residual norm of the PGl-GPBiCG method according to the residual norm of the global GMRES method that guarantees convergence. Finally, some numerical examples are presented to show the efficiency of the new method in comparison with the preconditioned global BiCGSTAB method, and a comparison with another preconditioner is also provided.

AMS subject classifications (2020): 65F10.

Keywords: Saddle point problems; Preconditioned global GPBiCG; Indefinite preconditioning; Multiple right-hand sides; Convergence analysis.

*Corresponding author
Received 18 June 2021; revised 21 August 2021; accepted 25 August 2021
Atefeh Taherian
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: a_taherian@mail.um.ac.ir

Faezeh Toutounian
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran
and
The Center of Excellence on Modeling and Control Systems, Ferdowsi University of Mashhad, Iran. e-mail: toutouni@math.um.ac.ir

1 Introduction

Many problems in science and engineering applications such as mixed or mixed-hybrid finite element discretization of partial differential equations in computational fluid dynamics [13, 18, 31, 32, 41] and constrained optimization [29, 42, 43] eventuate in systems with multiple right-hand sides as

$$\begin{bmatrix} A & B \\ \epsilon B^T & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{n\times m}$ has a full column rank, $X \in \mathbb{R}^{n\times s}$, $Y \in \mathbb{R}^{m\times s}$, $F_1 \in \mathbb{R}^{n\times s}$, and $F_2 \in \mathbb{R}^{m\times s}$. To convenience, we denote system (1) as follows:

$$\mathcal{A}\mathcal{X} = \mathcal{B}, \tag{2}$$

where $\mathcal{X} = [\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(s)}]$ and $\mathcal{B} = [b^{(1)}, \ldots, b^{(s)}]$. Under the assumptions mentioned above, the coefficient matrix $\mathcal{A}$ is nonsingular.

Several efficient iterative methods have been proposed during the recent decades to solve the saddle point problems, such as the Uzawa method [11, 12, 45], the Hermitian and skew-Hermitian splitting (HSS) iteration methods [3, 5, 9, 35, 37], SOR-type schemes [7, 19, 20, 26, 40, 47] and Krylov subspace methods [1, 2, 10, 21, 33, 34], and so on. In order to improve the efficiency of standard iterative solvers, many preconditioners have been presented in the literature, for example, block diagonal preconditioners [28, 38], constraint preconditioners [6, 25], block triangular preconditioners [4, 14, 36, 44], parametrized block triangular preconditioners [24], and HSS preconditioners [5, 9, 22, 30].

In this paper, we study the preconditioned global BiCG (PGl-BiCG) method [23] and preconditioned global GPBiCG (PGl-GPBiCG) method [46] for solving the linear systems with multiple right-hand sides (1). We concentrate on the use of the indefinite preconditioner

$$P = \begin{bmatrix} I & B \\ \epsilon B^T & 0 \end{bmatrix}. \tag{3}$$

This choice has been shown to be particularly effective on problems associated with constrained nonlinear programming [25, 27, 31]. In addition, in [33], the authors have shown that there is a tight connection between short term recurrence methods such as BiCG and the indefinite CG method used in [27]. More precisely, they are equivalent for a special choice of auxiliary vector, with which BiCG simplifies. In this paper, we discuss the convergence of the PGl-GPBiCG method and present two bounds for the residual norm of the method, which guarantee convergence.

The outline of the paper is as follows. In Section 2, we mention some properties of indefinite preconditioner. In Section 3, we review the PGl-

BiCG method. In Section 4, we describe the PGl-GPBiCG method to solve the system (1). Section 5 is devoted to the convergence analysis of the global GMRES method. In Section 6, we study some theoretical properties of the convergence of the PGl-GPBiCG method. In Section 7, numerical experiments confirm the described theoretical results. In Section 8, we present our conclusions.

Throughout this paper, we use the following notations. Inner product for two $n \times s$ matrices X and Y is defined as $\langle X, Y\rangle_F = \mathrm{tr}(X^T Y)$, where $\mathrm{tr}(Z)$ denotes the trace of the square matrix Z. The associated norm is the Frobenius norm denoted by $\|\cdot\|_F$. We will use the notation $\langle\cdot,\cdot\rangle_2$ for the usual inner product in $\mathbb{R}^n$, and the related norm will be denoted by $\|\cdot\|_2$. For a matrix $V \in \mathbb{R}^{n\times s}$, the block Krylov subspace $\mathcal{K}_k(A, V)$ is defined by $\mathcal{K}_k(A, V) = \mathrm{span}\{V, AV, A^2V, \ldots, A^{k-1}V\}$. Moreover, $Z \in \mathbb{K}_j(A, V)$ means that $Z = \sum_{i=0}^{j-1} \xi_i A^i V$, where $\xi_i \in \mathbb{R}$, for $i = 0, \ldots, j-1$. Finally, 0_s, I_s, and $0_{l\times s}$ will denote the zero, the identity, and zero matrices in $\mathbb{R}^{s\times s}$, $\mathbb{R}^{s\times s}$, and $\mathbb{R}^{l\times s}$, respectively. For brevity, we use the MATLAB-like notation $[v; w]$ to represent the vector $[v^T w^T]^T$.

2 Properties of the indefinite preconditioner

In this work, we use the global version of GPBiCG (Gl-GPBiCG)[46] for the solution of nonsymmetric saddle point problems with multiple right-hand sides (2). This method without a good preconditioner converges very slowly when applied to saddle point problems with multiple right-hand sides. In order to accelerate the convergence, we use the indefinite matrix P (defined in (3)) as a right preconditioner for the Gl-GPBiCG algorithm applied to the problem (1):

$$\mathcal{A}P^{-1}\begin{bmatrix}\tilde{X}\\ \tilde{Y}\end{bmatrix} = \mathcal{B}, \qquad \begin{bmatrix}X\\ Y\end{bmatrix} = P^{-1}\begin{bmatrix}\tilde{X}\\ \tilde{Y}\end{bmatrix}, \tag{4}$$

where

$$P^{-1} = \begin{bmatrix} I-\Pi & \frac{1}{\epsilon}B(B^TB)^{-1}\\ (B^TB)^{-1}B^T & -\frac{1}{\epsilon}(B^TB)^{-1}\end{bmatrix}, \qquad \mathcal{A}P^{-1} = \begin{bmatrix}G & S\\ 0 & I\end{bmatrix},$$

with $\Pi = B(B^TB)^{-1}B^T$, $G = A(I-\Pi)+\Pi$, and $S = \frac{1}{\epsilon}(A-I)B(B^TB)^{-1}$. Once an approximate solution $[\tilde{X}_k; \tilde{Y}_k]$ is determined, an approximate solution to the unpreconditioned problem is recovered as $[X_k; Y_k] = P^{-1}[\tilde{X}_k; \tilde{Y}_k]$. Choosing the vector $[\tilde{X}_0; \tilde{Y}_0] = [0; F_2]$ as the starting approximate solution, the initial residual is given by

$$R_0 = \begin{bmatrix}F_1\\ F_2\end{bmatrix} - \mathcal{A}P^{-1}\begin{bmatrix}\tilde{X}_0\\ \tilde{Y}_0\end{bmatrix} = \begin{bmatrix}F_1 - \frac{1}{\epsilon}(A-I)B(B^TB)^{-1}F_2\\ 0\end{bmatrix} = \begin{bmatrix}R_0^{(1)}\\ 0\end{bmatrix},$$

so that the second block component of R_0 is identically zero. Problem (4) can thus be reformulated as determining an approximation $[\bar{X}_k;\bar{Y}_k]$ to the solution $[\bar{X};\bar{Y}]$ of the system

$$\mathcal{A}P^{-1}\begin{bmatrix}\bar{X}\\ \bar{Y}\end{bmatrix}=R_0, \tag{5}$$

so that $[\tilde{X}_k;\tilde{Y}_k]=[\tilde{X}_0;\tilde{Y}_0]+[\bar{X}_k;\bar{Y}_k]$. In addition, for every $[U;0]\in\mathbb{R}^{(m+n)\times s}$, we have

$$\mathcal{A}P^{-1}\begin{bmatrix}U\\ 0\end{bmatrix}=\begin{bmatrix}GU\\ 0\end{bmatrix}\qquad\text{and}\qquad P^{-1}\begin{bmatrix}U\\ 0\end{bmatrix}=\begin{bmatrix}(I-\Pi)U\\ (B^TB)^{-1}B^TU\end{bmatrix}. \tag{6}$$

In [33], the authors proved the following proposition for showing that the matrix G does have a full set of eigenvectors Z and for giving a bound for condition number of Z.

Proposition 1. Let us assume that the matrix $G=A(I-\Pi)+\Pi$ has $n-m$ nonunit eigenvalues $\lambda_i, i=1,\ldots,n-m$. Let Z_2 be an orthonormal basis of span$\{\Pi\}$ and let the columns of $Y_1\in\mathbb{R}^{n\times(n-m)}$ be eigenvectors of $(I-\Pi)A(I-\Pi)$ corresponding to all its nonzero eigenvalues. Then there exists an eigenvector matrix in the form $Z=[Z_1,Z_2]$ of $A(I-\Pi)+\Pi$ such that

$$\kappa(Z)\leq(1+\|\gamma\|)^2\quad with\quad \|\gamma\|\leq\frac{\|A\|_2}{\min_i|\lambda_i-1|}, \tag{7}$$

where $\Lambda=diag(\lambda_i)$ and $\gamma=Z_2^TAY_1(\Lambda-I)^{-1}$.

The following proposition shows that the eigenvalues of $G=A(I-\Pi)+\Pi$ belong to a real positive interval when A is a symmetric positive definite matrix.

Proposition 2. Let τ be an eigenvalue of $G=A(I-\Pi)+\Pi$. Then either $\tau=1$ or τ is a nonzero eigenvalue of $(I-\Pi)A(I-\Pi)$.

Proof. The proof is similar to that of Proposition 5 in [31]. □

3 The preconditioned global BiCG method

In this section, we employ the PGl-BiCG method [23] to solve system (4). Let $X_0\in\mathbb{R}^{n\times s}$ be an initial guess with the residual $R_0=\mathcal{B}-\mathcal{A}^{-1}X_0$ and let $\tilde{R}_0$ be an arbitrary $n\times s$ matrix. At step k, the residual R_k generated by this algorithm is such that, R_k-R_0 lies in the right matrix Krylov subspace $\mathcal{K}_k(\mathcal{A},\mathcal{A}R_0)$ and R_k is F-orthogonal to the left matrix Krylov subspace $\mathcal{K}_k(\mathcal{A}^T,\tilde{R}_0)=\text{span}\{\tilde{R}_0,\mathcal{A}^T\tilde{R}_0,\ldots,\mathcal{A}^{T^{k-1}}\tilde{R}_0\}$.

For solving (1) by preconditioned Global Biconjugate Gradient (PGl-BiCG) algorithm by using the preconditioner P (defined in (3)) and equations (4) and (5), we choose $[\tilde{X}_0; \tilde{Y}_0] = [0; F_2]$ as an initial guess. So, $R_0 = \binom{R_0^{(1)}}{0}$. As in [33], we set $\tilde{R}_0 = P^{-1}R_0 = \binom{(I-\Pi)R_0^{(1)}}{(B^TB)^{-1}B^TR_0^{(1)}}$, $P_0 = R_0$, $\tilde{P}_0 = \tilde{R}_0$, and we obtain $\tilde{R}_k = P^{-1}R_k = \binom{(I-\Pi)R_k^{(1)}}{(B^TB)^{-1}B^TR_k^{(1)}}$ and $\tilde{P}_k = P^{-1}P_k = \binom{(I-\Pi)P_k^{(1)}}{(B^TB)^{-1}B^TP_k^{(1)}}$. Therefore, the iterates $\tilde{R}_k$ and $\tilde{P}_k$ can be computed explicitly from R_k and P_k, and the auxiliary "tilde" recurrence can be omitted. Now, by using the relations (6) and ignoring from the last m rows of the matrices that are zero matrices, we can summarize the PGl-BiCG algorithm for solving (2) as follows:

Algorithm 1: The right PGl-BiCG algorithm for solving (1)

1. Set $\begin{bmatrix}\tilde{X}_0\\ \tilde{Y}_0\end{bmatrix} = \begin{bmatrix}0\\ F_2\end{bmatrix}$ and $\begin{bmatrix}\bar{X}_0\\ \bar{Y}_0\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}$.
2. Compute $R_0^{(1)} = F_1 - \frac{1}{\epsilon}(A-I)B(B^TB)^{-1}F_2$ and set $R_0 = \begin{bmatrix}R_0^{(1)}\\ 0\end{bmatrix}$ and $P_0 = \begin{bmatrix}P_0^{(1)}\\ P_0^{(2)}\end{bmatrix} = \begin{bmatrix}R_0^{(1)}\\ 0\end{bmatrix}$
3. **for** $k = 0, 1, 2, \ldots$ until convergence
4. $\quad \alpha_k = \frac{<R_k^{(1)}, (I-\Pi)R_k^{(1)}>_F}{<GP_k^{(1)}, (I-\Pi)P_k^{(1)}>_F}$
5. $\quad \bar{X}_{k+1} = \bar{X}_k + \alpha_k P_k^{(1)}, \qquad \bar{Y}_{k+1} = O_{m\times s}$
6. $\quad R_{k+1}^{(1)} = R_k^{(1)} - \alpha_k G P_k^{(1)}, \qquad R_{k+1}^{(2)} = 0$
7. $\quad \beta_k = \frac{<R_{k+1}^{(1)}, (I-\Pi)R_{k+1}^{(1)}>_F}{<R_k^{(1)}, (I-\Pi)R_k^{(1)}>_F}$
8. $\quad P_{k+1}^{(1)} = R_{k+1}^{(1)} + \beta_k P_k^{(1)}, \qquad P_{k+1}^{(2)} = 0$
9. **end**
10. $\tilde{X}_{k+1} = \tilde{X}_0 + \bar{X}_{k+1}, \qquad \tilde{Y}_{k+1} = \tilde{Y}_0 + \bar{Y}_{k+1},$
11. $X_{k+1} = (I-\Pi)\tilde{X}_{k+1} + \frac{1}{\epsilon}B(B^TB)^{-1}\tilde{Y}_{k+1}, \qquad Y_{k+1} = (B^TB)^{-1}B^T\tilde{X}_{k+1} - \frac{1}{\epsilon}(B^TB)^{-1}\tilde{Y}_{k+1}$

In practical implementation of Algorithm 1, we can factorize B as $B = QR$ and use the relation $(B^TB)^{-1} = R^{-1}R^{-T}$. We also note that the cost of solving with R is very low due to the particular structure and sparsity of the matrix.

From Algorithm 1, the first block of residuals and first block of matrix directions can be expressed as follows:

$$R_k^{(1)PGl-BiCG} = \mathcal{R}_k(G)R_0^{(1)}, \quad P_k^{(1)PGl-BiCG} = \mathcal{P}_k(G)R_0^{(1)}, \tag{8}$$

where $\mathcal{R}_k(t)$, $\mathcal{P}_k(t) \in \mathbb{P}_k, \mathbb{P}_k$ is the set of polynomials $p_k(t)$ of degree k with scalar coefficients satisfying $p_k(0) = 1$. The polynomials $\mathcal{R}_k(t)$ and $\mathcal{P}_k(t)$ are related together with the recurrence formulas as follows:

$$\mathcal{R}_{k+1}(t) = \mathcal{R}_k(t) - \alpha_k t\mathcal{P}_k(t), \tag{9}$$
$$\mathcal{P}_{k+1}(t) = \mathcal{R}_{k+1}(t) + \beta_k\mathcal{P}_k(t). \tag{10}$$

4 The preconditioned global GPBiCG method

In the PGl-GPBiCG method [46], the matrix residual satisfies

$$R_k = \mathcal{H}_k(\mathcal{A}^{-1}P)\mathcal{R}_k(\mathcal{A}^{-1}P)R_0, \tag{11}$$

where $\mathcal{R}_k$ is defined as above and $\mathcal{H}_k$ is an accelerating scalar polynomial, which is computed as the following recurrence:

$$\begin{aligned} &\mathcal{H}_0(t) = 1, \quad \mathcal{G}_0(t) = \zeta_0, \\ &\mathcal{H}_k(t) = \mathcal{H}_{k-1}(t) - t\mathcal{G}_{k-1}(t), \\ &\mathcal{G}_k(t) = \zeta_k\mathcal{H}_k(t) + \eta_k\mathcal{G}_{k-1}(t), \quad k = 1, 2, \ldots. \end{aligned} \tag{12}$$

If the PGl-GPBiCG algorithm is written for system (5) with $R_0 = \begin{bmatrix} R_0^{(1)} \\ 0 \end{bmatrix}$, then the relation (11) can be written as

$$R_k = \begin{bmatrix} \mathcal{H}_k(G)\mathcal{R}_k(G)R_0^{(1)} \\ 0 \end{bmatrix}, \tag{13}$$

and for solving (1.2), the PGl-GPBiCG algorithm can be summarized as Algorithm 2.

5 Convergence analysis of the global GMRES method

In this section, we consider the block linear system $GY = C$, where $G \in \mathbb{R}^{n\times n}$ and $Y, C \in \mathbb{R}^{n\times s}$, and we recall some convergence properties of the global GMRES method, which is needed in what follows. We consider the case where the matrix G is diagonalizable. Let $G = ZDZ^{-1}$, where D is a diagonal matrix whose elements are the eigenvalues $\lambda_1, \ldots, \lambda_n$ and Z is the eigenvector matrix. In [8], the following upper bounds for the Frobenius norm of the kth residual $R_k = C - GY_k$ of the global GMRES method was given. Here $R_0 = C - GY_0$ denotes the residual corresponding to the initial solution Y_0.

Theorem 1. [8] Let $\mathbb{P}_k$ be the set of polynomials of degree less or equal than k, and let $\kappa_2(Z) = \|Z\|_2\|Z^{-1}\|_2$. Then we have the following results:

$$\|R_k\|_F \leq \kappa_2(Z)\|R_0\|_F \min_{p\in\mathbb{P}_k; p(0)=1} (\max_{\lambda\in Sp(G)} |p(\lambda)|), \tag{14}$$

Algorithm 2: The right PGl-GPBiCG algorithm

1. Set $\begin{bmatrix}\tilde{X}_0\\ \tilde{Y}_0\end{bmatrix}=\begin{bmatrix}0\\ F_2\end{bmatrix}$ and $\begin{bmatrix}\bar{X}_0\\ \bar{Y}_0\end{bmatrix}=\begin{bmatrix}0\\ 0\end{bmatrix}$.
2. Compute $R_0^{(1)}=F_1-\frac{1}{\epsilon}(A-I)B(B^TB)^{-1}F_2$ and set $R_0=\begin{bmatrix}R_0^{(1)}\\ 0\end{bmatrix}$,
3. Set $P_0=R_0^{(1)}$, $T_{-1}=W_{-1}=P_{-1}=U_{-1}=0_{n\times s}$, $\beta_{-1}=0$
4. **for** $k=0,1,2,\ldots$ until convergence
5. $P_k=R_k^{(1)}+\beta_{k-1}(P_{k-1}-U_{k-1})$
6. $\alpha_k=\frac{<R_k^{(1)},(I-\Pi)R_0^{(1)}>_F}{<GP_k,(I-\Pi)R_0^{(1)}>_F}$
7. $Y_k=T_{k-1}-R_k^{(1)}-\alpha_kW_{k-1}+\alpha_kGP_k$
8. $T_k=R_k^{(1)}-\alpha_kGP_k$
9. $\zeta_k=\frac{<Y_k,Y_k>_F<GT_k,T_k>_F-<Y_k,T_k>_F<GT_k,Y_k>_F}{<GT_k,GT_k>_F<Y_k,Y_k>_F-<Y_k,GT_k>_F<GT_k,Y_k>_F}$
10. $\eta_k=\frac{<GT_k,GT_k>_F<Y_k,T_k>_F-<Y_k,GT_k>_F<GT_k,T_k>_F}{<GT_k,GT_k>_F<Y_k,Y_k>_F-<Y_k,GT_k>_F<GT_k,Y_k>_F}$
    (if $k=0$, then $\eta_k=0,\zeta_k=\frac{<GT_k,T_k>_F}{<GT_k,GT_k>_F}$
11. $U_k=\zeta_kGP_k+\eta_k(T_{k-1}-R_k^{(1)}+\beta_{k-1}U_{k-1})$
12. $Z_k=\zeta_kR_k^{(1)}+\eta_kZ_{k-1}-\alpha_kU_k$
13. $\tilde{X}_{k+1}=\tilde{X}_k+\alpha_kP_k+Z_k$, $\quad\bar{Y}_{k+1}=O_{m\times s}$
14. $R_{k+1}^{(1)}=T_k-\eta_kY_k-\zeta_kGT_k$ $\quad R_{k+1}^{(2)}=0$
15. $\beta_k=\frac{\alpha_k}{\zeta_k}\frac{<R_{k+1}^{(1)},(I-\Pi)R_0^{(1)}>}{<R_k^{(1)},(I-\Pi)R_0^{(1)}>}$
16. $W_k=GT_k+\beta_kGP_k$
17. **end**
18. $\tilde{X}_{k+1}=\tilde{X}_0+\bar{X}_{k+1}$, $\quad\tilde{Y}_{k+1}=\tilde{Y}_0+\bar{Y}_{k+1}$,
19. $X_{k+1}=(I-\Pi)\tilde{X}_{k+1}+\frac{1}{\epsilon}B(B^TB)^{-1}\tilde{Y}_{k+1}$, $\quad Y_{k+1}=(B^TB)^{-1}B^T\tilde{X}_{k+1}-\frac{1}{\epsilon}(B^TB)^{-1}\tilde{Y}_{k+1}$

where $Sp(G)$ is the set of eigenvalues of matrix G.

Theorem 2. [8] Let the initial residual R_0 be decomposed as $R_0=Z\beta$, where β is an $n\times s$ matrix whose columns are denoted by $\beta^{(1)},\ldots,\beta^{(s)}$. Then

$$\|R_k\|_F^2\leq\frac{\|Z\|_2^2}{e_1^T(V_{k+1}^T\tilde{\mathcal{D}}V_{k+1})^{-1}e_1},$$

where

$$\tilde{\mathcal{D}}=diag\{\sum_{i=1}^{s}|\beta_1^{(i)}|^2,\ldots,\sum_{i=1}^{s}|\beta_n^{(i)}|^2\}\quad\text{and}\quad V_{k+1}=\begin{pmatrix}1&\lambda_1&\cdots&\lambda_1^k\\ \vdots&\vdots&&\vdots\\ 1&\lambda_n&\cdots&\lambda_n^k\end{pmatrix}.$$

The coefficients $\beta_1^{(i)},\ldots,\beta_n^{(i)}$ are the components of the vector $\beta^{(i)}$, and e_1 is the first unit vector of $\mathbb{R}^{k+1}$.

From the fact that the eigenvalues of the matrix $G = A(I - \Pi) + \Pi$ are positive, the min-max problem in the bound of Theorem 1 can be explicitly solved. Denoting by $\lambda_{\min}$ and $\lambda_{\max}$ the (positive) extreme eigenvalues of the matrix G, the following asymptotic bound holds:

$$\|R_k\|_F \leq \kappa_2(Z)\|R_0\|_F(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^k, \tag{15}$$

where $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ stands for the ratio of the extremal (real) eigenvalues of the matrix $G = A(I - \Pi) + \Pi$ (and so it is not its condition number!)[34].

6 Convergence behavior of PGl-GPBiCG

In this section, we propose an upper bound for the residual norm of the PGl-GPBiCG method. First, we derive an upper bound for the Frobenius norm of the kth residual $R_k^{PGl-BiCG}$ of the preconditioned global BiCG method. Using the steps 6 and 8 of Algorithm 1 and defining $\bar{R}_{k+1}^{(1)} = [R_0^{(1)} \quad R_1^{(1)} \quad \dots \quad R_k^{(1)}]$, where $R_i^{(1)}$, $i = 0, \dots, k$, are the first block of the PGl-BiCG residual matrices, we can easily show that

$$G\bar{R}_k^{(1)} = \bar{R}_k^{(1)}\hat{T}_k - \frac{1}{\alpha_{k-1}}R_k^{(1)}\bar{E}_k^T,$$

with $\bar{E}_k^T = [0_s, \dots, 0_s, I_s] \in \mathbb{R}^{s\times ks}$ and $\hat{T}_k = \hat{L}_k\hat{\Lambda}_k^{-1}\hat{U}_k$, where

$$\hat{L}_k = \begin{pmatrix} I_s & & & & \\ -I_s & I_s & & & \\ & -I_s & \ddots & & \\ & & \ddots & I_s & \\ & & & -I_s & I_s \end{pmatrix}, \qquad \hat{U}_k = \begin{pmatrix} I_s & -\beta_0 I_s & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & I_s & -\beta_{k-2}I_s \\ & & & & I_s \end{pmatrix},$$

$\hat{\Lambda}_k = \text{diag}[\alpha_0 I_s, \dots, \alpha_{k-1}I_s]$, and α_i and β_i, $i = 0, 1, \dots, k$, are scalars obtained from the PGl-BiCG algorithm. In addition, $\hat{T}_k$ is an invertible tridiagonal block matrix such that $\bar{E}_k^T\hat{T}_k^{-1}\bar{E}_1 = \alpha_{k-1}I_s$, where $\bar{E}_1^T = [I_s, \dots, 0_s, 0_s] \in \mathbb{R}^{s\times ks}$.

By assuming that the matrix $\bar{R}_{k+1}^{(1)}$ is of full rank and $(\bar{R}_{k+1}^{(1)})^+ = ((\bar{R}_{k+1}^{(1)})^T\bar{R}_{k+1}^{(1)})^{-1}(\bar{R}_{k+1}^{(1)})^T$, as in [39], an upper bound for the norm of the first block of the residual of PGl-GPBiCG can be stated in the following theorem.

Theorem 3. Suppose that $G\bar{R}_k^{(1)} = \bar{R}_k^{(1)}\hat{T}_k - \frac{1}{\alpha_{k-1}}R_k^{(1)}\bar{E}_k^T$ with $\bar{E}_k^T\hat{T}_k^{-1}\bar{E}_1 = \alpha_{k-1}I_s$, and $\hat{V} = [I_{ks} \quad 0_{ks\times s}](\bar{R}_{k+1}^{(1)})^+$. If the matrix $\bar{R}_{k+1}^{(1)}$ is of full rank,

then for any polynomial $p(t)$ of degree not exceeding k with $p(0) = 1$, we have

$$\|R_k^{(1)PGl-BiCG}\|_F \leq \|\bar{N}_k\|_F \, \| \, p(G)R_0^{(1)} \, \|_F, \tag{16}$$

where $\bar{N}_k = I_n - (\bar{R}_k^{(1)} - R_k^{(1)}\bar{F}_k^T)\hat{V}$ with $\bar{F}_k^T = [I_s, I_s, \ldots, I_s] \in \mathbb{R}^{s\times ks}$.

Proof. The proof is similar to that of Theorem 1 in [39]. □

Now, we attempt to obtain an upper bound for the residual norm of the PGl-GPBiCG method. As in [39], we define the parameters $\bar{\zeta}_{j+1} = \zeta_j$, $\bar{\eta}_{j+1} = \eta_j$, and matrices

$$H_{j+1} = \mathcal{H}_j(G)R_k^{(1)PGl-BiCG}, \qquad G_{j+1} = \frac{1}{\zeta_j}\mathcal{G}_j(G)R_k^{(1)PGl-BiCG}, \quad j = 0, \ldots, k.$$

Using the relations (12), the iterates H_j and G_j can be computed by the recurrence formulas:

$$\begin{aligned} H_1 &= R_k^{(1)PGl-BiCG}, \quad G_1 = R_k^{(1)PGl-BiCG}, \\ H_{j+1} &= H_j - \bar{\zeta}_j G G_j, \\ G_{j+1} &= H_{j+1} + \frac{\bar{\eta}_{j+1}}{\bar{\zeta}_{j+1}}\bar{\zeta}_j G_j, \quad j = 1, 2, \ldots, k. \end{aligned} \tag{17}$$

By assuming that $\bar{\zeta}_j \neq 0, j = 1, \ldots, k+1$, we have $H_{j+1}, G_{j+1} \in \mathbb{K}_{j+1}(G, R_k^{(1)PGl-BiCG})$. Moreover, we obtain

$$H_{k+1} = R_k^{(1)PGl-GPBiCG}. \tag{18}$$

Under the assumption that all the generated coefficients ζ_j are not zero and the grade μ of $R_k^{(1)PGl-BiCG}$ with respect to G is not less than k, the recurrence formulas (17) determine the matrices

$$\bar{H}_k = [H_1, H_2, \ldots, H_k] \qquad \text{and} \qquad \bar{G}_k = [G_1, G_2, \ldots, G_k],$$

whose matrix columns H_j and G_j, $j = 1, \ldots, k$, are linear independent.

As in [39], by defining $\gamma_{j+1} = \bar{\zeta}_j\bar{\eta}_{j+1}\bar{\zeta}_{j+1}^{-1}$, for $j = 1, 2, \ldots, k$, the relations (17) can be written as follows:

$$G\bar{G}_k = \bar{H}_k\bar{L}_k\bar{\Lambda}_k^{-1} - \bar{\zeta}_k^{-1}H_{k+1}\bar{E}_k^T, \qquad \bar{H}_k = \bar{G}_k\bar{U}_k, \tag{19}$$

where $\bar{\Lambda}_k = \text{diag}[\bar{\zeta}_1 I_s, \ldots, \bar{\zeta}_k I_s]$, $\bar{E}_k^T = [0_s, \ldots, 0_s, I_s] \in \mathbb{R}^{s\times ks}$,

$$\bar{L}_k = \hat{L}_k, \quad \text{and} \quad \bar{U}_k = \begin{pmatrix} I_s & -\gamma_2 I_s & & \\ & I_s & & \\ & & \ddots & \\ & & \ddots & -\gamma_k I_s \\ & & & I_s \end{pmatrix}. \tag{20}$$

Combining two equations in (19), we get

$$G\bar{H}_k = \bar{H}_k \bar{S}_k - \bar{\zeta}_k^{-1} H_{k+1} \bar{E}_k^T, \tag{21}$$

with $\bar{S}_k = \bar{L}_k \bar{\Lambda}_k^{-1} \bar{U}_k$ and $\bar{E}_k^T \bar{S}_k^{-1} \bar{E}_1 = \bar{\zeta}_k I_s$, where $\bar{E}_1^T = [I_s, 0_s, \ldots, 0_s] \in \mathbb{R}^{s \times ks}$.

By assuming that the matrix $\bar{H}_{k+1}$ is of full rank and that $\bar{H}_{k+1}^+ = (\bar{H}_{k+1}^T \bar{H}_{k+1})^{-1} \bar{H}_{k+1}^T$, similar to Theorem 3, we can present the following theorem for the PGl-GPBiCG method.

Theorem 4. Let $G\bar{H}_k = \bar{H}_k \bar{S}_k - \bar{\zeta}_k^{-1} H_{k+1} \bar{E}_k^T$ with $\bar{E}_k^T \bar{S}_k^{-1} \bar{E}_1 = \bar{\zeta}_k I_s$, and let $\bar{V} = [I_{ks} \quad 0_{ks \times s}] \bar{H}_{k+1}^+$. If the matrix $\bar{H}_{k+1}$ is of full rank, then for any polynomial $p(t)$ of degree not exceeding k with $p(0) = 1$, we have

$$\| R_k^{(1)^{PGl-GPBiCG}} \|_F \leq \| \bar{M}_k \|_F \, \| \, p(G) R_k^{(1)^{PGl-BiCG}} \, \|_F, \tag{22}$$

where $\bar{M}_k = I_n - (\bar{H}_k - H_{k+1} \bar{F}_k^T) \bar{V}$ with $\bar{F}_k^T = [I_s, I_s, \ldots, I_s] \in \mathbb{R}^{s \times ks}$.

Proof. The proof is similar to that of Theorem 1 in [39]. □

Now, based on the above observations, we state the following theorem for bounding the residual norm of the PGl-GPBiCG method.

Theorem 5. With the notation of Theorems 3 and 4 for $R_0 = \begin{bmatrix} R_0^{(1)} \\ 0 \end{bmatrix}$ and assuming that $G = A(I - \Pi) + \Pi$ is diagonalizable, the right preconditioned global GPBiCG residual satisfies

$$\| R_k^{PGl-GPBiCG} \|_F \leq (\kappa_2(Z))^2 \| \bar{N}_k \|_F \| \bar{M}_k \|_F \| R_0 \|_F \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2k}, \tag{23}$$

where $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ stands for the ratio of the extremal (real) eigenvalues of matrix $G = A(I - \Pi) + \Pi$.

In addition, with the notation of Theorem 2 and $R_0^{(1)} = \beta Z$, we have

$$\| R_k^{PGl-GPBiCG} \|_F \leq \kappa_2(Z) \| \bar{N}_k \|_F \| \bar{M}_k \|_F \frac{\| Z \|_2}{\sqrt{(e_1^T (V_{k+1}^T \tilde{D} V_{k+1})^{-1} e_1)}} \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \tag{24}$$

where $\tilde{D}$ and V_{k+1} are defined in Theorem 2.

So, the convergence is guaranteed.

Proof. Since the inequality (22) holds for any polynomial $p(t)$ of degree not exceeding k with $p(0) = 1$, the use of the residual polynomial of the global GMRES ($p_k^{Gl-GMRES}(G)$) method in (22) implies that

$$\begin{aligned}\|R_k^{PGl-GPBiCG}\|_F &\leq \|\bar{M}_k\|_F \parallel p_k^{Gl-GMRES}(G)R_k^{(1)PGl-BiCG} \parallel_F \\ &\leq \kappa_2(Z)\|\bar{M}_k\|_F \parallel R_k^{(1)PGl-BiCG} \parallel_F (\tfrac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^k,\end{aligned}$$

The second inequality follows from (15). Now, from the fact that (16) also holds for any polynomial $p(t)$ of degree not exceeding k with $p(0) = 1$, we get

$$\|R_k^{(1)PGl-GPBiCG}\|_F \leq \kappa_2(Z)\|\bar{N}_k\|_F\|\bar{M}_k\|_F \parallel p_k^{Gl-GMRES}(G)R_0^{(1)} \parallel_F (\tfrac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^k. \tag{25}$$

This together with the inequality (15) implies that

$$\|R_k^{(1)PGl-GPBiCG}\|_F \leq (\kappa_2(Z))^2\|\bar{N}_k\|_F\|\bar{M}_k\|_F\|R_0^{(1)}\|_F(\tfrac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})^{2k}. \tag{26}$$

Since $R_k^{(2)} = 0$, so we have the inequality (23).

Finally, due to Theorem 2, the inequality (25) yields the inequality given in (24). □

7 Numerical results

In this section, we illustrate the efficiency of the PGl-GPBiCG method for solving system (1). All the numerical experiments were performed in MATLAB R2017b with PC-Intel(R) Core(TM) i7, CPU 3.60 GHz, 16.00 GB of RAM. In all the examples, the starting guess was taken to be zero. The stopping criterion

$$\frac{\|R_k\|_F}{\|R_0\|_F} \leq 10^{-9}$$

was used.

Example 1. In this example, a set of 6 problems was taken from the University of Florida Sparse Matrix Collection[17] as the matrix A. These matrices with their generic properties are given in Table 1. Also, we consider $\epsilon = 1$, $B = \text{rand(n, m)}$, $F_1 = \text{rand}(n, s)$, and $F_2 = \text{rand}(m, s)$, where the function $\text{rand}(l, k)$ creates an $l \times k$ random matrix with coefficients uniformly distributed in [0, 1].

We compare the Frobenius norm of the residuals (RES), the number of iterations (Iter), and the CPU time in seconds (CPU) required for convergence of the PGl-GPBiCG and PGl-BiCGSTAB methods for $s = 5, 10, 20$.

	Matrix	Property		
		order	nnz	spd
1	gr_30_30	900	7744	yes
2	nos3	960	15844	yes
3	1138_bus	1138	4054	yes
4	bcsstm11	1473	1473	yes
5	Dubcova1	16129	253009	yes
6	bodyy4	17546	121550	yes

Table 1: Test problems information

The results obtained by these algorithms are presented in Table 2. As can be seen, the methods are similar in terms of the Frobenius norm of residuals ($\| R_k \|_F$), the number of iterations of the PGl-GPBiCG method is less than that of the PGl-BiCGSTAB method and the CPU time required for convergence in the PGl-GPBiCG method is similar or lower than that in the PGl-BiCGSTAB method except, for examples, nos3 with $s = 10, 20$.

		PGl-BiCGSTAB			PGl-GPBiCG		
Matrix \ s		5	10	20	5	10	20
gr_30_30	Iter	47	49	49	38	38	38
	CPU	0.02	0.04	0.07	0.02	0.03	0.06
	RES	1.1082e-07	3.4408e-07	6.3886e-07	2.6191e-07	4.2295e-07	6.1987e-07
nos3	Iter	192	183	193	167	180	172
	CPU	0.16	0.17	0.23	0.16	0.24	0.30
	RES	2.7939e-07	7.3530e-07	8.0708e-07	3.9410e-07	7.6350e-07	1.0669e-06
1138_bus	Iter	8364	4011	5393	2793	2677	2543
	CPU	9.61	6.02	10.72	4.23	4.84	6.33
	RES	4.8730e-06	6.4503e-06	9.1015e-06	3.2011e-06	5.1722e-06	1.0123e-05
bcsstm11	Iter	335	381	441	216	335	250
	CPU	0.98	1.20	1.72	0.59	1.05	1.11
	RES	4.0741e-07	2.0770e-07	4.6741e-07	4.0252e-07	5.4683e-07	2.0180e-07
Dubcova1	Iter	105	99	99	85	85	86
	CPU	30.48	35.21	40.87	24.62	31.08	37.33
	RES	1.2814e-06	2.1262e-06	3.1051e-06	1.5025e-06	2.1308e-06	3.2268e-06
bodyy4	Iter	182	194	170	165	166	165
	CPU	68.18	80.66	80.75	62.54	69.50	79.63
	RES	1.5969e-06	2.1782e-06	3.2651e-06	1.6013e-06	2.3639e-06	3.1067e-06

Table 2: Numerical results for Example 1 with $s = 5, 10, 20$

Example 2. In this example, we consider the Stokes equation [16, 15] as

$$\begin{aligned} -\nu\Delta\mathbf{u} + \bigtriangledown p &= \tilde{f}, \quad \text{in } \Omega \\ \bigtriangledown \cdot \mathbf{u} &= \tilde{g}, \quad \text{in } \Omega \\ \mathbf{u} &= 0, \quad \text{on } \partial\Omega \\ \int_\Omega p(x)dx &= 0, \end{aligned} \tag{27}$$

where $\Omega = (0,1) \times (0,1) \subset \mathbb{R}^2$, $\partial\Omega$ is the boundary of Ω, ν is the viscosity scalar, and u and p denote the velocity and the pressure, respectively. By discretizing (27), we obtain the system of linear equations as

$$\begin{bmatrix} A & B \\ -B^T & 0 \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} F_1 \\ -F_2 \end{bmatrix},$$

in which

$$A = \begin{bmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{bmatrix} \in \mathbb{R}^{2q^2 \times 2q^2}, \quad B = \begin{bmatrix} I \otimes F \\ F \otimes I \end{bmatrix} \in \mathbb{R}^{2q^2 \times q^2},$$

where T and F are tridiagonal matrices given by

$$T = \frac{\nu}{h^2} \cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{q \times q}, \quad F = \frac{1}{h} \cdot \text{tridiag}(-1, 1, 0) \in \mathbb{R}^{q \times q},$$

and $\otimes$ denotes the Kronecker product. Also, $h = \frac{1}{q+1}$ is the discretization mesh size. We set $n = 2q^2$ and $m = q^2$. Hence, the total number of variables is $n + m = 3q^2$. We choose the right-hand side such that the exact solution of saddle point problem is a matrix of ones. For this example, we test three different ν's, that is, $\nu = 0.01, 0.1, 1$ and $q = 16, 32$.

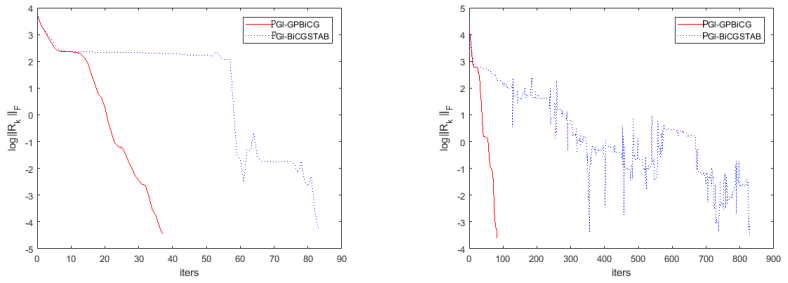
In Table 3, we list the Frobenius norm of the residuals (RES), the number of iterations (Iter), and the CPU time in seconds (CPU) required for the convergence of the PGl-GPBiCG and PGl-BiCGSTAB methods with $\nu = 1, 0.1, 0.01$, $q = 16, 32$, and $s = 5$. As expected, the number of iterations of PGl-GPBiCG is less than that of PGl-BiCGSTAB. The CPU time obtained for PGl-GPBiCG is smaller than the one for PGl-BiCGSTAB except for $\nu = 0.01$ and $q = 16$.

		PGl-BiCGSTAB		PGl-GPBiCG	
$\nu \setminus q$		16	32	16	32
0.01	Iter	38	74	23	47
	CPU	0.03	2.16	0.03	1.97
	RES	1.7694e-06	6.4937e-06	2.3360e-06	6.6436e-06
0.1	Iter	70	222	44	80
	CPU	0.04	3.10	0.03	2.20
	RES	9.3235e-07	1.8710e-05	4.2220e-06	2.3113e-05
1	Iter	83	828	37	82
	CPU	0.05	6.70	0.04	2.21
	RES	5.5359e-05	2.7881e-04	3.5071e-05	2.3870e-04

Table 3: Numerical results for Example 2 with $\nu = 0.01, 0.1, 1$

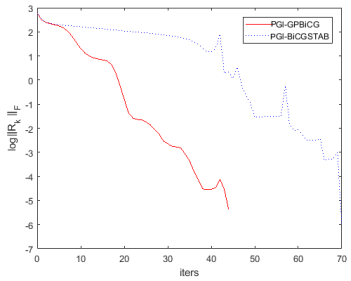
In Figure 1, we display the convergence history of the PGl-GPBiCG and PGl-BiCGSTAB algorithms for Stokes problem with $s = 5$, $q = 16, 32$, and

$\nu = 0.01, 0.1, 1$, respectively. In these figures, the horizontal axis is the number of iterations (iters) and the vertical axis is the logarithm of the Frobenius norm of residuals ($\log_{10} \parallel R_k \parallel_F$). The results show that whatever v becomes lower, the methods are smoother. Moreover, the PGl-GPBiCG method is more effective and smoother than the PGl-BiCGSTAB method.

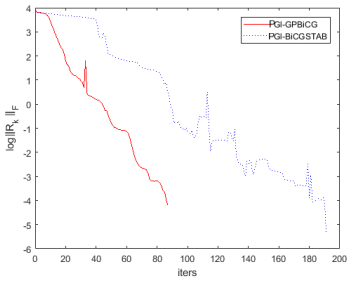


(a) $\nu = 1,\ q = 16$

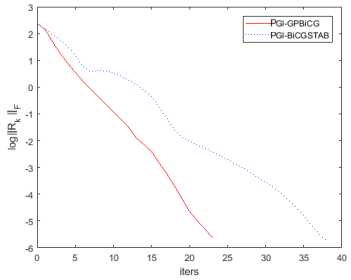
(b) $\nu = 1,\ q = 32$



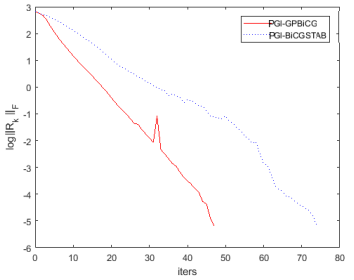
(c) $\nu = 0.1,\ q = 16$



(d) $\nu = 0.1,\ q = 32$



(e) $\nu = 0.01\ q = 16$



(f) $\nu = 0.01,\ q = 32$

Figure 1: Convergence history of the PGl-GPBiCG algorithm for Stokes problem with $s = 5$ and different values of ν and q.

Finally, we compare the numerical results obtained by the Gl-GPBiCG method with the preconditioner (3) and the preconditioner given in [1],

$$P_{\epsilon,\alpha,Q} = \begin{bmatrix} A & B \\ \epsilon B^T & \alpha Q \end{bmatrix}$$

with $Q = I$. For $q = 16, 32, 64$, $\nu = 1$, $\alpha = 0.05, 0.1, 0.5, 1, 10$, and $s = 5$ the numerical results are given in Table 4. As expected, the preconditioner discussed in this paper needs less CPU time than the preconditioner given in [1] except for $q = 64$ and $\alpha = 0.5, 1, 10$.

		PGl-GPBiCG (pre. in [1])					PGl-GPBiCG (pre. (3))
$q \setminus \alpha$		0.05	0.1	0.5	1	10	
	Iter	196	88	25	22	35	37
16	CPU	0.66	0.35	0.11	0.10	0.14	0.04
	RES	2.7200e-04	6.6880e-05	4.2144e-05	7.7757e-05	6.2674e-05	3.5071e-05
	Iter	276	109	38	35	53	82
32	CPU	17.10	6.87	2.50	2.35	3.53	2.21
	RES	0.0013	6.1728e-04	1.5556e-4	2.5773e-04	1.3334e-04	2.3870e-04
	Iter	318	162	56	56	91	201
64	CPU	315.67	162.91	60.20	60.15	94.10	128.57
	RES	0.0060	0.0017	0.0021	0.0014	2.5978e-04	0.0015

Table 4: Numerical results for Example 2 with $\alpha = 0.05, 0.1, 0.5, 1, 10$

8 Conclusion

In this paper, we applied the Gl-GPBiCG method to solve the nonsymmetric saddle point problems with multiple right-hand sides. By using the indefinite preconditioner (3), we accelerated the convergence rate of the method. Also, we studied some theoretical properties of the PGl-GPBiCG method and presented two bounds for the residual norm of the method, which guarantee the convergence. As expected, the experimental results showed that the number of iterations (Iter) of the PGl-GPBiCG method is less than that of the PGl-BiCGSTAB method. The CPU time (CPU) of the PGl-GPBiCG method is sometimes more than that of the PGl-BiCGSTAB method because of the existence of parameters ζ_j and η_j. In addition, the PGl-GPBiCG method is more effective and smoother than the PGl-BiCGSTAB method.

Acknowledgements

We would like to thank the referees for their valuable remarks and helpful suggestions that improved the presentation.

References

1. Badahmane, A., Bentbib, A.H., and Sadok, H. *Preconditioned global Krylov subspace methods for solving saddle point problems with multiple right-hand sides*, Electron. Trans. Numer. Anal. 51 (2019), 495–511.

2. Badahmane, A., Bentbib, A.H. and Sadok, H. *Preconditioned Krylov subspace and GMRHSS iteration methods for solving the nonsymmetric saddle point problems,* Numer. Algorithms, 84 (2020), 1295–1312.

3. Bai Z.-Z. and Benzi, M. *Regularized HSS iteration methods for saddle-point linear systems,* BIT Numerical Mathematics, 57 (2017), 287–311.

4. Bai, Z.-Z., Golub, G.H., Lu, L.-Z. and Yin, J.-F. *Block triangular skew-Hermitian splitting methods for positive-definite linear systems,* SIAM J. Sci. Comput. 26 (2004), 844–863.

5. Bai, Z.-Z., Golub, G.H. and Pan, J.-Y. *Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems,* Numer. Math. 98 (2004), 1–32.

6. Bai, Z.-Z., Ng, M.K. and Wang, Z.-Q. *Constraint preconditioners for symmetric indefinite matrices,* SIAM J. Matrix Anal. Appl. 31 (2009), 410–433.

7. Bai, Z.-Z., Parlett, B.N. and Wang, Z.-Q. *On generalized successive overrelaxation methods for augmented linear systems,* Numer. Math. 102 (2005), 1–38.

8. Bellalij, M., Jbilou, K. and Sadok, H. *New convergence results on the global GMRES method for diagonalizable matrices,* J. Comput. Appl. Math. 219 (2008), 350–358.

9. Benzi M. and Golub, G.H. *A preconditioner for generalized saddle point problems,*SIAM J. Matrix Anal. Appl. 26 (2004), 20–41.

10. Benzi, M. and Wathen, A.J. S*ome preconditioning techniques for saddle point problems. Model order reduction: theory, research aspects and applications*, 195–211, Math. Ind., 13, Eur. Consort. Math. Ind. (Berl.), Springer, Berlin, 2008.

11. Bramble, J.H., Pasciak, J.E. and Vassilev, A.T. *Analysis of the inexact Uzawa algorithm for saddle point problems,* SIAM J. Numer. Anal. 34 (1997), 1072–1092.

12. Bramble, J.H., Pasciak, J.E. and Vassilev, A.T. *Uzawa type algorithms for nonsymmetric saddle point problems,* Math. Comp. 69 (2000), 667–689.

13. Brezzi, F. and Fortin, M. *Mixed and hybrid finite element methods,* Springer Series in Computational Mathematics, 15. Springer-Verlag, New York, 1991.

14. Cao, Z.-H. *Positive stable block triangular preconditioners for symmetric saddle point problems,* Appl. Numer. Math 57 (2007), 899–910.

15. Cao, Y., Du, J. and Niu, Q. *Shift-splitting preconditioners for saddle point problems,* J. Comput. Appl. Math. 272 (2014), 239–250.

16. Chen, C. and Ma, C. *A generalized shift-splitting preconditioner for saddle point problems,* Appl. Math. Lett. 43 (2015), 49–55.

17. Davis, T. and Hu, Y. *The university of Florida sparse matrix collection*, ACM Trans. Math. Software 38 (2011), no. 1, Art. 1, 25 pp.

18. Elman, H.C., Silvester, D.J. and Wathen, A.J. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics,* Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2005.

19. Feng, T.T., Chen, G.L. and Guo, X.P. *An accelerated SOR-Like method for generalized saddle point problems,* East Asian J. Appl. Math. 8 (2018), 70–81.

20. Golub, G.H., Wu, X. and Yuan, J.-Y. *SOR-like methods for augmented systems*, BIT 41 (2001), 71–85.

21. Gould, N., Orban D. and Rees, T. *Projected Krylov methods for saddle point systems,* SIAM J. Matrix Anal. Appl. 35 (2014), 1329–1343.

22. Guo, C. and Li, J., *A new preconditioner for solving weighted Toeplitz least squares problems,* Math. Appl. (Wuhan) 33 (2020), no. 1, 172–185.

23. Jbilou, K., Sadok, H. and Tinzefte, A. *Oblique projection methods for linear systems with multiple right-hand sides,* Electron Trans. Numer. Anal. 20 (2005), 119–138.

24. Jiang, M.-Q., Cao, Y. and Yao, L.-Q. *On parametrized block triangular preconditioners for generalized saddle point problems,* Appl. Math. Comput. 216 (2010), 1777–1789.

25. Keller, C., Gould, N.I.M. and Wathen, A.J. *Constraint preconditioning for indefinite linear systems,* SIAM J. Matrix Anal. Appl. 21 (2000), 1300–1317.

26. Li, C., Li, Z., Evans, D.J. and Zhang, T. *A note on an SOR-like method for augmented systems,* IMA J. Numer. Anal. 23 (2003), 581–592.

27. Lukšan, L. and Vlčcek, J. *Indefinitely preconditioned inexact Newton method for large sparse equality constrained non-linear programming problems,* Numer. Linear Algebra Appl., 5 (1998), 219–247.

28. Murphy, M.F., Golub, G.H. and Wathen, G.H. *A note on preconditioning for indefinite linear systems,* SIAM J. Sci. Comput 21 (2000), 1969–1972.

29. Nocedal, J. and Wright, S.J. *Numerical optimization,* Springer Series in Operations Research, Springer, Berlin 1999.

30. Pan, J.-Y., Ng, M.K. and Bai, Z.-Z. *New preconditioners for saddle point problems,* Appl. Math. Comput. 172 (2006), 762–771.

31. Perugia, I. and Simoncini, V. *Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations,* Numer. Linear Algebra Appl. 7 (2000), 585–616.

32. Perugia, I., Simoncini, V. and Arioli, M. *Linear algebra methods in a mixed approximation of magnetostatic problems,* SIAM J. Sci. Comput. 21 (1999), 1085–1101.

33. Rozložník, M. and Simoncini, V. *Krylov subspace methods for saddle point problems with indefinite preconditioning,* SIAM J. Matrix Anal. Appl. 24 (2002), 368–391.

34. Saad, Y. *Iterative Methods for Sparse Linear Systems,* SIAM, Philadelphia, 2003.

35. Salkuyeh, D.K. and Masoudi, M. *A new relaxed HSS preconditioner for saddle point problems,* Numer. Algor. 74 (2017), 781–795.

36. Simoncini, V. *Block triangular preconditioners for symmetric saddle-point problems,* Appl. Numer. Math. 49 (2004), 63–80.

37. Simoncini, V. and Benzi, D.M. *Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems,* SIAM J. Matrix Anal. Appl. 26 (2004/05), 377–389.

38. Sturler, E.D. and Liesen, J. *Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems,* SIAM J. Sci. Comput 26 (2005), 598–619.

39. Taherian, A. and Toutounian, F. *Block GPBi-CG method for solving non-symmetric linear systems with multiple right-hand sides and its convergence analysis,* J Numer. Algor. (2021), https://doi.org/10.1007/s11075-021-01097-7.

40. Wen, R., Wu R. and Guan, J. *Some generalizations of the new SOR-like method for solving symmetric Saddle point problems,* J. Inequal. Appl. 2018, Paper No. 145, 12 pp.

41. Wesseling, P. *Principles of computational fluid dynamics,* Vol. 29 of Springer Series in Computational Mathematics, Springer, Berlin 2001.

42. Wright, M.H. *Interior methods for constrained optimization,* Acta numerica, 1992, 341–407, Acta Numer., Cambridge Univ. Press, Cambridge, 1992.

43. Wright, S.J. *Primal-dual interior point methods,* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

44. Wu, X.-N., Golub, G.H., Cuminato, J.A. and Yuan, J.-Y. *Symmetric-triangular decomposition and its applications Part II: Preconditioners for indefinite systems.* BIT 48(2008), 139–162.

45. Yun, J.H. *Variants of the Uzawa method for saddle point problem,* Comput. Math. Appl. 65(2013), 1037–1046.

46. Zhang, J. and Dai, H. *Global GPBiCG method for complex non-Hermitian linear systems with multiple right-hand sides,* Comput. Appl. Math. 35 (2016), 171–185.

47. Zheng, Q. and Ma, C. *A new SOR-Like method for the saddle point problems,* Appl. Math. Comput. 233 (2014), 421–429.

How to cite this article
Taherian, A., Toutounian, F. Preconditioned global GPBiCG method for solving saddle point problems with multiple right-hand sides and its convergence analysis. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 111-129. doi: 10.22067/ijnao.2021.71036.1044

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 131-144
DOI:10.22067/ijnao.2021.70074.1028
https://ijnao.um.ac.ir/

Research Article

Multiple interpolation with the fast-growing knots in the class of entire functions and its application

I. Sheparovych

Abstract

The conditions for the sequence of complex numbers $(b_{n,k})$ are obtained, such that the interpolation problem $g^{(k-1)}(\lambda_n) = b_{n,k}$, $k \in \overline{1,s}$, $n \in \mathbb{N}$, where $|\lambda_k/\lambda_{k+1}| \leq \Delta < 1$, has a unique solution in some classes of entire functions g for which $M_g(r) \leq c_1 \exp((s-1)N(r) + N(\rho_1 r))$, where $N(r)$ is the counting function of the sequence (λ_n), $\rho_1 \in (\Delta; 1)$, and $c_1 > 0$. Moreover, these results have been applied to the description of the solution of the differential equation $f^{(s)} + A_0(z)f = 0$ for which (λ_n) is zero-sequence and the coefficient A_0 is an entire function from the mentioned class.

AMS subject classifications (2020): 30D15, 30E05.

Keywords: Interpolation problem; Entire function; Solution of differential equation.

1 Introduction

Let $(b_{n,k})$ be an arbitrary sequence of complex numbers and let (λ_k) be a sequence of distinct complex numbers without finite limit points. Set $N(r) = \int_0^r \frac{n(t)}{t} dt$, where $n(r)$ is equal to the number of points of the sequence (λ_k) in the disk $|z| < r$. Note that $N(r) = \sum_{|\lambda_k| \leq r} \log \frac{r}{|\lambda_k|}$. Let $g(z)$ be an entire function, and let $M_g(r) = \max\{|g(z)| : |z| = r\}$.

Received 26 April 2021; revised 2 September 2021; accepted 3 September 2021
Iryna Sheparovych
Department of Physics, Mathematics, Economy and Innovative Technologies, Ivan Franko State Pedagogical Institute of Drohobych, Drohobych, Ukraine. e-mail: isheparovych@ukr.net

Hel'fond [11] and Kaz'min [14] considered an interpolation problem $g(\lambda_k) = b_k$, $k \in \mathbb{N}$, with interpolation knots in $\lambda_k = q^{k-1}$ and $|q| > 1$. From their results, the next theorem follows.

Theorem A (Kaz'min). Let $\lambda_k = q^{k-1}, |q| > 1$. Then for every sequence (b_k) such that

$$\overline{\lim_{k\to\infty}} |q|^{-\frac{(k-1)}{2}} |b_k|^{1/k} \le r_1, \quad r_1 \in (\Delta; 1),$$

the interpolation problem $g(\lambda_k) = b_k$ has a unique solution in the class of entire functions g, that satisfy the condition

$$\ln M_g(r) \le \frac{\ln^2 \rho_1 r}{2 \ln |q|} + \frac{\ln r}{2} + c_2$$

for each $\rho_1 > r_1$ and some $c_2 > 0$ (here and farther c_i are positive constants).

The aim of this paper is to consider the interpolation problem

$$g^{(k-1)}(\lambda_n) = b_{k,n}, \qquad k \in \{1, \dots, s\}, n \in \mathbb{N}, \tag{1}$$

where $s \in \mathbb{N}$ and the sequence (λ_n) satisfies the condition

$$|\lambda_n / \lambda_{n+1}| \le \Delta, \quad n \in \mathbb{N}, \tag{2}$$

for some $\Delta \in (0; 1)$. It should be noted that in the case when $s = 2$, the problem was solved in [21] and the next assertion was proved.

Theorem B. Let (λ_k) be a sequence of complex numbers satisfying condition (2). Then for every sequences $(b_{n,1})$ and $(b_{n,2})$ such that for some $q \in (\Delta; 1)$,

$$|b_{k,1}| \le c_1 \exp\left(2N\left(q|\lambda_k|\right)\right), \ |\lambda_k||b_{k,2}| \le c_2 \exp\left(N\left(|\lambda_k|\right) + N\left(q|\lambda_k|\right)\right), \ k \in \mathbb{N},$$

the interpolation problem $g(\lambda_k) = b_{k,1}, g'(\lambda_k) = b_{k,2}, k \in \mathbb{R}$ has a unique solution in the class of entire functions g that satisfy the condition

$$M_g(r) \le c_3 \exp\left(N(r) + N(\rho_1 r)\right)$$

for each $\rho_1 \in (q; 1)$. The interpolation function has the form

$$g(z) = \sum_{k=1}^{\infty} \left(-\frac{L''(\lambda_k) b_{k,1}}{L'^3(\lambda_k)} \frac{L^2(z)}{z - \lambda_k} + \frac{b_{k,2}}{L'^2(\lambda_k)} \frac{L^2(z)}{z - \lambda_k} + \frac{b_{k,1}}{L'^2(\lambda_k)} \frac{L^2(z)}{(z - \lambda_k)^2} \right),$$

where $L(z) = \prod\limits_{j=1}^{\infty} \left(1 - \frac{z}{\lambda_j}\right)$.

In this article, this result is generalized. We prove the following theorem.

Theorem 1. Let (λ_k) be a sequence of nonzero numbers satisfying condition (2) for some $\Delta \in (0;1)$, then for every sequences $(b_{k,n})$, $k \in \overline{1,s}$, $n \in \mathbb{N}$, such that

$$|b_{n,k}| \leq c_1 k! |\lambda_n|^{-k} \exp\left(kN\left(|\lambda_n|\right) + (s-k)N\left(q|\lambda_n|\right)\right), \tag{3}$$

where $q \in (\Delta;1)$ and $c_1 > 0$, the interpolation problem (1) has an unique solution in the class of entire functions g, such that

$$M_g(r) \leq c_2 \exp((s-1)N(r) + N(\rho_1 r)) \tag{4}$$

for each $\rho_1 \in (q;1)$ and some $c_2 > 0$.

To prove this, we construct the function g by the known methods used in [5, 15, 16, 21]. The function has the form

$$g(z) = \sum_{n=1}^{\infty} \sum_{k=1}^{s} \frac{L^s(z)}{(s-k)!(z-\lambda_n)^k} \sum_{i=0}^{s-k} C_{s-k}^i b_{n,i} \gamma_{s,s-k-i}(\lambda_n), \tag{5}$$

where $L(z) = \prod\limits_{k=1}^{\infty} (1 - z/\lambda_k)$ and $\gamma_{s,j}(z) = \left(\frac{(z-\lambda_n)^s}{L^s(z)}\right)^{(j)}$.

2 Preliminaries

We need some lemmas.

Lemma 1. [21] Let (λ_k) be a sequence of distinct complex nonzero numbers, satisfying condition (2) for some $\Delta \in (0;1)$. Then there exists a constant $c > 1$ such that $n(\rho_2 r) \leq n(r) + c$ for each $\rho_2 \geq 1$ with $\rho_2 \Delta \leq 1$, and for every $\rho_1, 0 < \rho_1 \leq \rho_2$, the inequality

$$N(\rho_2 r) \leq N(\rho_1 r) + c \log \frac{\rho_2}{\rho_1} + n(r) \log \frac{\rho_2}{\rho_1} \tag{6}$$

is fulfilled.

Lemma 2. If (λ_k) is a sequence of distinct complex nonzero numbers such that the series $\sum\limits_{k=1}^{+\infty} 1/|\lambda_k|$ is convergent and $L(z) = \prod\limits_{k=1}^{\infty} (1 - z/\lambda_k)$, then for $l \geq 2$, it holds that

$$\frac{L^{(l)}(\lambda_k)}{L'(\lambda_k)} = \frac{l}{l-1} \sum_{n=1, n\neq k}^{\infty} \sum_{j=2}^{l} \frac{L^{(j-1)}(\lambda_k)}{L'(\lambda_k)} \frac{(-1)^j (l-j)!}{(\lambda_k - \lambda_n)^{l-j+1}}. \tag{7}$$

Proof. In [2, 13], there is the relation $\frac{L''(\lambda_k)}{L'(\lambda_k)} = 2\sum\limits_{n=1,n\neq k}^{\infty}\frac{1}{\lambda_k-\lambda_n}$, (it was justified in [21]). Now we argue similarly. The next relationships

$$L(z)=\sum_{i=1}^{\infty}\frac{1}{i!}L^{(i)}(\lambda_k)(z-\lambda_k)^i,\ \frac{L(z)}{(z-\lambda_k)}=\sum_{i=1}^{\infty}\frac{1}{i!}L^{(i)}(\lambda_k)(z-\lambda_k)^{i-1},$$

$$\left(\frac{L(z)}{z-\lambda_k}\right)^{(s)}=\sum_{i=s+1}^{\infty}\frac{1}{i(i-s-1)!}L^{(i)}(\lambda_k)(z-\lambda_k)^{i-s-1},$$

$$\left(\frac{L(z)}{z-\lambda_n}\right)^{(s)}=\sum_{j=0}^{s}(-1)^j(s-j)!L^{(j)}(z)(z-\lambda_n)^{-s+j-1},$$

are true. So, by differentiating $l-1$ times the equality $L'(z)=\frac{L(z)}{z-\lambda_k}+\sum\limits_{n=1,n\neq k}^{\infty}\frac{L(z)}{z-\lambda_n}$, we obtain

$$\begin{aligned}
L^{(l)}(z)=&\left(\frac{L(z)}{z-\lambda_k}\right)^{(l-1)}+\sum_{n=1,n\neq k}^{\infty}\left(\frac{L(z)}{z-\lambda_n}\right)^{(l-1)}\\
=&\sum_{i=l}^{\infty}\frac{1}{i(i-l)!}L^{(i)}(\lambda_k)(z-\lambda_k)^{i-l}\\
&+\sum_{n=1,n\neq k}^{\infty}\frac{1}{(z-\lambda_n)^l}\sum_{j=0}^{l-1}(-1)^j(l-j-1)!L^{(j)}(z)(z-\lambda_n)^j\\
=&\frac{1}{l}L^{(l)}(\lambda_k)+o(z-\lambda_k)\\
&+\sum_{n=1,n\neq k}^{\infty}\frac{1}{(z-\lambda_n)^l}\sum_{j=0}^{l-1}(-1)^j(l-j-1)!L^{(j)}(z)(z-\lambda_n)^j\\
=&\frac{1}{l}L^{(l)}(\lambda_k)+o(z-\lambda_k)\\
&+\sum_{n=1,n\neq k}^{\infty}\sum_{j=1}^{l}\frac{(-1)^{j-1}(l-j)!}{(z-\lambda_n)^{l-j+1}}L^{(j-1)}(z),\quad z\to\lambda_k.
\end{aligned}$$

Furthermore, for $z\to\lambda_k$, the next equality

$$\frac{(l-1)L^{(l)}(\lambda_k)}{lL'(\lambda_k)}=\sum_{n=1,n\neq k}^{\infty}\sum_{j=2}^{l}\frac{(-1)^j(l-j)!}{(\lambda_k-\lambda_n)^{l-j+1}}\frac{L^{(j-1)}(\lambda_k)}{L'(\lambda_k)}$$

holds, which completes the proof of Lemma 2. □

Lemma 3. If a sequence (λ_k) satisfies condition (2) for some $\Delta \in (0;1)$, then the next inequality is true

$$\left|\frac{L^{(j)}(\lambda_k)}{L'(\lambda_k)}\right| \le c(j)\left(\frac{k}{|\lambda_k|}\right)^{j-1}, \quad j \in \mathbb{N}\backslash\{1\}.$$

Proof. From (7), we can obtain

$$\left|\frac{L^{(j)}(\lambda_k)}{L'(\lambda_k)}\right| \le c(j)\left(\sum_{n=1,n\neq k}^{\infty}\frac{1}{|\lambda_k-\lambda_n|}\right)^{j-1}, \quad j \ge 2. \tag{8}$$

Indeed, for $j=2$, we have

$$\left|\frac{L''(\lambda_k)}{L'(\lambda_k)}\right| \le 2\left|\sum_{n=1,n\neq k}^{\infty}\frac{1}{\lambda_k-\lambda_n}\right| \le 2\sum_{n=1,n\neq k}^{\infty}\frac{1}{|\lambda_k-\lambda_n|}.$$

Let us use the induction to obtain (8). Assume that (8) is true for $j \le l$ and prove it for $j = l+1$:

$$\begin{aligned}
\left|\frac{L^{(l+1)}(\lambda_k)}{L'(\lambda_k)}\right| &\le \frac{l+1}{l}\left|\sum_{n=1,n\neq k}^{\infty}\sum_{j=2}^{l+1}(-1)^{j+1}\frac{L^{(j-1)}(\lambda_k)}{L'(\lambda_k)}\frac{(l+1-j)!}{(\lambda_k-\lambda_n)^{l-j+2}}\right| \\
&= \frac{l+1}{l}\left|\sum_{n=1,n\neq k}^{\infty}\left((-1)^{l}\frac{L^{(l)}(\lambda_k)}{L'(\lambda_k)}\frac{1}{(\lambda_k-\lambda_n)}+\sum_{j=2}^{l}(-1)^{j+1}\frac{L^{(j-1)}(\lambda_k)}{L'(\lambda_k)}\frac{(l+1-j)!}{(\lambda_k-\lambda_n)^{l-j+2}}\right)\right| \\
&\le \frac{l+1}{l}\left|\frac{L^{(l)}(\lambda_k)}{L'(\lambda_k)}\right|\sum_{n=1,n\neq k}^{\infty}\frac{1}{|\lambda_k-\lambda_n|} \\
&\quad + \frac{l+1}{l}\sum_{j=2}^{l}(l+1-j)!\left|\frac{L^{(j-1)}(\lambda_k)}{L'(\lambda_k)}\right|\sum_{n=1,n\neq k}^{\infty}\frac{1}{|\lambda_k-\lambda_n|^{l-j+2}} \\
&\le c(l)\left(\sum_{n=1,n\neq k}^{\infty}\frac{1}{|\lambda_k-\lambda_n|}\right)^{l} \\
&\quad + \sum_{j=2}^{l}(l+1-j)!c(j-1)\left(\sum_{n\neq k}\frac{1}{|\lambda_k-\lambda_n|}\right)^{j-2}\sum_{n\neq k}\frac{1}{|\lambda_k-\lambda_n|^{l-j+2}} \\
&\le c(l)\left(\sum_{n=1,n\neq k}^{\infty}\frac{1}{|\lambda_k-\lambda_n|}\right)^{l}\sum_{j=2}^{l+1}(l+1-j)! \le C(l)\left(\sum_{n\neq k}\frac{1}{|\lambda_k-\lambda_n|}\right)^{l}.
\end{aligned}$$

Thus, using a simple mathematical calculation, we have ($c_3 := c(j)$)

$$\left|\frac{L^{(j)}(\lambda_k)}{L'(\lambda_k)}\right| \le c_3\left(\frac{1}{|\lambda_k|}\sum_{n=1}^{k-1}\frac{1}{1-|\lambda_n/\lambda_k|}+\sum_{n=k+1}^{\infty}\frac{1}{|\lambda_n|\,(1-|\lambda_k/\lambda_n|)}\right)^{j-1}$$
$$\le \frac{c_3}{|\lambda_k|^{j-1}}\left(\sum_{n=1}^{k-1}\frac{1}{1-\Delta}+\sum_{n=k+1}^{\infty}\frac{|\lambda_k|}{|\lambda_n|(1-\Delta^{n-k})}\right)^{j-1}$$
$$\le \frac{c_3}{|\lambda_k|^{j-1}}\left(\frac{k-1}{1-\Delta}+\sum_{n=k+1}^{\infty}\Delta^{n-k}\right)^{j-1} \le c_4\left(\frac{k}{|\lambda_k|}\right)^{j-1},$$

where $c_4 := c_4(j,\Delta)$. □

Lemma 4. If a sequence (λ_k) satisfies condition (2), then for $\gamma_{s,j}(z) := \left(\frac{(z-\lambda_n)^s}{L^s(z)}\right)^{(j)}$, $j \in \overline{0,s-1}$, where $L(z) = \prod\limits_{n=1}^{\infty}\left(1-\frac{z}{\lambda_n}\right)$, the inequalities

$$|\gamma_{s,0}(\lambda_n)| \le c_3|\lambda_n|^s \exp(-sN_\lambda(|\lambda_n|)+cs), \tag{9}$$
$$|\gamma_{s,l}(\lambda_n)| \le c_4 s(s+1)\dots(s+l-1)|\lambda_n|^{s-l}n^l \exp(-sN(|\lambda_n|)+cs) \tag{10}$$

are fulfilled.

Proof. First, we prove (9). By the known equality (see, for example, [20]) $\log|\lambda_k L'(\lambda_k)| = N(|\lambda_k|)+O(1), k\in\mathbb{N}$, we have

$$\left|\lambda_n^{-s}\gamma_{s,0}(\lambda_n)\right| = \frac{1}{|\lambda_n L'(\lambda_n)|^s} = \exp(-sN(\lambda_n)+O(s)), \quad n\in\mathbb{N}.$$

Furthermore, let us prove (10). Since $L(z) = \sum\limits_{i=1}^{\infty}\frac{1}{i!}L^{(i)}(\lambda_n)(z-\lambda_n)^i$ and $L'(z) = \sum\limits_{i=1}^{\infty}\frac{i}{i!}L^{(i)}(\lambda_n)(z-\lambda_n)^{i-1}$, we have

$$L(z)-(z-\lambda_n)L'(z) = -\sum_{i=2}^{\infty}\frac{i-1}{i!}L^{(i)}(\lambda_n)(z-\lambda_n)^i,$$

$$\big(L(z)-(z-\lambda_n)L'(z)\big)^{(l)} = -\sum_{i=l}^{\infty}\frac{i(i-1)\dots(i-l+1)(i-1)}{i!}L^{(i)}(\lambda_n)(z-\lambda_n)^{i-l}.$$

Then

$$\gamma_{s,1}(z) = \left(\frac{(z-\lambda_n)^s}{L^s(z)}\right)' = s\frac{(z-\lambda_n)^{s-1}}{L^{s+1}(z)}(L(z)-(z-\lambda_n)L'(z))$$
$$= -s\frac{(z-\lambda_n)^{s+1}}{L^{s+1}(z)}\sum_{i=2}^{\infty}\frac{i-1}{i!}L^{(i)}(\lambda_n)(z-\lambda_n)^{i-2}$$
$$= -s\frac{(z-\lambda_n)^{s+1}}{L^{s+1}(z)}\left(\frac{1}{2!}L''(\lambda_n)+\frac{2}{3!}L'''(\lambda_n)(z-\lambda_n)+\cdots\right)$$
$$= -s\frac{(z-\lambda_n)^{s+1}}{2L^{s+1}(z)}(L''(\lambda_n)+O(z-\lambda_n)), \quad z\to\lambda_n;$$

$$\gamma_{s,1}(\lambda_n) = -\frac{sL''(\lambda_n)}{2!\,(L'(\lambda_n))^{s+1}} = -\frac{s}{2!\,(L'(\lambda_n))^{s}}\frac{L''(\lambda_n)}{L'(\lambda_n)}.$$

Thus,

$$\begin{aligned}
\gamma_{s,2}(z) &= \left(\left(\frac{z-\lambda_n}{L(z)}\right)^s\right)'' = -\left(s\left(\frac{z-\lambda_n}{L(z)}\right)^{s+1}\sum_{i=2}^{\infty}\frac{i-1}{i!}L^{(i)}(\lambda_n)(z-\lambda_n)^{i-2}\right)' \\
&= s\frac{(s+1)(z-\lambda_n)^{s+2}}{L^{s+2}(z)}\left(\sum_{i=2}^{\infty}\frac{i-1}{i!}L^{(i)}(\lambda_n)(z-\lambda_n)^{i-2}\right)^2 \\
&\quad - s\frac{(z-\lambda_n)^{s+1}}{L^{s+1}(z)}\sum_{i=3}^{\infty}\frac{(i-1)(i-2)}{i!}L^{(i)}(\lambda_n)(z-\lambda_n)^{i-3}; \\
\gamma_{s,2}(\lambda_n) &= \frac{s(s+1)}{(2!)^2\,(L'(\lambda_n))^{s+2}}\left(L''(\lambda_n)\right)^2 - \frac{2sL'''(\lambda_n)}{3!\,(L'(\lambda_n))^{s+1}} \\
&= \frac{s}{2\,(L'(\lambda_n))^{s}}\left(\frac{(s+1)}{2}\left(\frac{L''(\lambda_n)}{L'(\lambda_n)}\right)^2 - \frac{2L'''(\lambda_n)}{3L'(\lambda_n)}\right).
\end{aligned}$$

Continuing the process, we can obtain (by mathematical induction) a general formula, for $l \geq 1$,

$$\begin{aligned}
\gamma_{s,l}(\lambda_n) =& (-1)^l s(s+1)(s+2)\dots(s+l-1)\frac{(L''(\lambda_n))^l}{2^l\,(L'(\lambda_n))^{s+l}} \\
&+ (-1)^{l-1}C_l^2 s(s+1)(s+2)\dots(s+l-2)\frac{(L''(\lambda_n))^{l-2}\,L'''(\lambda_n)}{(2!)^2 3!\,(L'(\lambda_n))^{s+l-1}} + \cdots \\
&- \frac{s}{(l+1)}\frac{L^{(l+1)}(\lambda_n)}{(L'(\lambda_n))^{s+1}}.
\end{aligned}$$

It is not difficult to show that (by Lemma 3)

$$\begin{aligned}
\left|\gamma_{s,l}(\lambda_n)\right| \leq& c_4\frac{s(s+1)\dots(s+l-1)n^l}{|\lambda_n|^l|L'(\lambda_n)|^s} \\
\leq& c_4 s(s+1)\dots(s+l-1)|\lambda_n|^{s-l}n^l\exp(-sN(|\lambda_n|)+cs).
\end{aligned}$$

□

3 Proof of Theorem 1

Proof of Theorem 1. First, we will prove the *uniqueness*. Assume to the contrary that for some sequences $(b_{n,k})$ with the properties (3) in the class (4), there exist two different entire functions $g = f_1$ and $g = f_2$ that solve problem (1). Then the function $f = f_2 - f_1$ has zeros of order $m_k \geq s$ at all points λ_k and satisfies the condition (2) for some $\rho_1 < 1$. This implies from the Jensen inequality $\ln M_f(r) \geq sN(r) + O(1)$ that $N(r) \leq N(\rho_1 r) + c_0$. This is a contradiction, because $N(r) - N(\rho_1 r) \to +\infty$, if $\rho_1 < 1$ and $r \to +\infty$. Thus uniqueness is proved.

Now, we will prove that the function g of form (5) satisfies condition (4). Since (see [20, 18, 19]) for every $n \in \mathbb{N}$, $r \in [0; +\infty)$, and $\varepsilon > 0$,

$$\max\left\{\left|\frac{L(z)}{z-\lambda_n}\right| : |z| \le r\right\} \le c(\varepsilon)\frac{M_L((1+\varepsilon)r)}{r+|\lambda_n|}, \tag{11}$$

and

$$\exp\left(N(\lambda_n)\right) = \frac{|\lambda_n|^n}{\prod\limits_{k=1}^{n}|\lambda_k|}, \quad \exp\left(N\left(q|\lambda_n|\right)\right) = \frac{(q|\lambda_n|)^{n-1}}{\prod\limits_{k=1}^{n-1}|\lambda_k|} = q^{n-1}\frac{|\lambda_n|^n}{\prod\limits_{k=1}^{n}|\lambda_k|},$$

$q \in (\Delta; 1)$, applying Lemmas 2, 3, and 4 and conditions (3), (9), and (10), we obtain ($1 \le k \le s$; $0 \le i \le s-k$)

$$\begin{aligned}\left|C_{s-k}^{i} b_{n,i}\gamma_{s,s-k-i}(\lambda_n)\right| &\le c_4 i! C_{s-k}^{i} s(s+1)\dots(s+s-k-i-1)|\lambda_n|^k n^{s-k-i} \\ &\quad \times \exp\left(cs-(s-i)N\left(|\lambda_n|\right)+(s-i)N\left(q|\lambda_n|\right)\right) \\ &\le c_4\frac{(s-k)!(2s-k-i-1)!}{(s-k-i)!(s-1)!}|\lambda_n|^k n^{s-k-i}q^{(s-i)(n-1)}.\end{aligned}$$

Thus, by (11) and the equality $\ln M_L(r) = N(r)+O(1), r \in [0; +\infty)$, (see, for example, [21]), for each $k \in \overline{1,s}$ and some $s \in \mathbb{N}$, one has (with $\rho_2 := (1+\varepsilon)$)

$$\begin{aligned}I_{k,n} &:= \left|\frac{L^s(z)}{(z-\lambda_n)^k}\sum_{i=0}^{s-k} C_{s-k}^{i} b_{n,i}\gamma_{s,s-k-i}(\lambda_n)\right| \\ &\le \left|L^{s-k}(z)\right|\left|\frac{L^k(z)}{(z-\lambda_n)^k}\right|\sum_{i=0}^{s-k} C_{s-k}^{i}|b_{n,i}\gamma_{s,s-k-i}(\lambda_n)| \\ &\le c_4\exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_2 r)+cs\right)\left(\frac{|\lambda_n|q^n}{r+|\lambda_n|}\right)^k\frac{(s-k)!}{(s-1)!} \\ &\quad \times\sum_{i=0}^{s-k} q^{-s+i}(nq^n)^{s-k-i}\frac{(2s-k-i)!}{(s-k-i)!}.\end{aligned}$$

Furthermore, suppose that $|\lambda_m| \le r < |\lambda_{m+1}|$,that $|z| \le r$, and that $\rho_1 = \rho_2 q$. Then $m = n(r)$ and

$$\left(\frac{|\lambda_n|q^n}{r+|\lambda_n|}\right)^k \le \left(\frac{|\lambda_n|q^n}{r}\right)^k \le \left(\frac{|\lambda_n|q^n}{|\lambda_m|}\right)^k \le \left(q^n\Delta^{m-n}\right)^k = \Delta^{mk}\left(\frac{q}{\Delta}\right)^{nk},$$

So, proceeding from Lemma 1,

$$\begin{aligned}\exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_2 r)+cs\right)\left(\tfrac{|\lambda_n|q^n}{r+|\lambda_n|}\right)^k \\ \le \exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_1 r)+km\ln\tfrac{1}{q}+cs\right)\Delta^{mk}\left(\tfrac{q}{\Delta}\right)^{nk} \\ \le \exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_1 r)+cs\right)\left(\tfrac{\Delta}{q}\right)^{(m-n)k}\end{aligned}$$

for $m>n$. Hence, if $m>n$, then

$$\begin{aligned}I_{k,n} \le & c_4\exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_1 r)+cs\right)\left(\frac{\Delta}{q}\right)^{(m-n)k}\frac{(s-k)!}{(s-1)!} \\ & \times\sum_{i=0}^{s-k} q^{-s+i}(nq^n)^{s-k-i}\frac{(2s-k-i)!}{(s-k-i)!}.\end{aligned}$$

If $m<n$, then

$$\begin{aligned}&\exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_2 r)+cs\right)\left(\tfrac{|\lambda_n|q^n}{r+|\lambda_n|}\right)^k \\ &\quad\le \exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_1 r)+km\ln\tfrac{1}{q}+cs\right)q^{kn} \\ &\quad= \exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_1 r)+cs\right)q^{k(n-m)}\end{aligned}$$

and

$$\begin{aligned}I_{k,n} \le & c_4\exp\left((s-k)N_\lambda(r)+kN_\lambda(\rho_1 r)+cs\right) \\ & \times q^{k(n-m)}\tfrac{(s-k)!}{(s-1)!}\sum_{i=0}^{s-k} q^{-s+i}(nq^n)^{s-k-i}\tfrac{(s+s-k-i)!}{(s-k-i)!}.\end{aligned}$$

Therefore, since $nq^{n-1}\le 1$ for $n\ge n_0$ and

$$\sum_{n=1}^{m}\left(\frac{\Delta}{q}\right)^{(m-n)k} \le \frac{1}{1-(\Delta/q)^k}\le\frac{q}{q-\Delta},\quad \sum_{n=m+1}^{\infty} q^{k(n-m)}\le\frac{q^k}{1-q},$$

$$\sum_{k=1}^{s}\frac{1}{(s-1)!}\sum_{i=0}^{s-k}\frac{(2s-k-i)!}{(s-k-i)!}\le\frac{2s^2\Gamma(s-1)}{(s+1)!\Gamma(2s-1)}=:c(s),$$

we have

$$\begin{aligned}|g(z)| &\le \sum_{n=1}^{\infty}\sum_{k=1}^{s}\left|\frac{L^s(z)}{(s-k)!(z-\lambda_n)^k}\right|\sum_{i=0}^{s-k}C_{s-k}^i|b_{n,i}\gamma_{s,s-k-i}(\lambda_n)| \\ &= \sum_{n=1}^{m}\sum_{k=1}^{s}\frac{I_{n,k}}{(s-k)!}+\sum_{n=m+1}^{+\infty}\sum_{k=1}^{s}\frac{I_{n,k}}{(s-k)!}\end{aligned}$$

$$\leq c_5 \sum_{k=1}^{s} \frac{1}{(s-1)!} \exp\left((s-k)N(r) + kN(\rho_1 r) + cs\right)$$
$$\times \left(\sum_{n=1}^{m} \left(\frac{\Delta}{q}\right)^{(m-n)k} \sum_{i=0}^{s-k} q^{-s+i} (nq^n)^{s-k-i} \frac{(2s-k-i)!}{(s-k-i)!} \right.$$
$$\left. + \sum_{n=m+1}^{\infty} q^{k(n-m)} \sum_{i=0}^{s-k} q^{-s+i} (nq^n)^{s-k-i} \frac{(2s-k-i)!}{(s-k-i)!} \right)$$
$$\leq c(s,\Delta) \exp\left((s-1)N(r) + N(\rho_1 r)\right).$$

□

Remark: If (λ_k) satisfies (2), then class (4) can be defined by the condition

$$M_g(r) \leq c_2 \exp\left(sN(\rho_3 r)\right) \tag{12}$$

for every $\rho_3 \in (\sqrt[s]{\rho_1}; 1)$.

This statement follows from a similar one in [21, Remark 3].

4 One application to the differential equations

Multiple interpolation can be applied to the problem of oscillation of linear differential equations of higher order, in particular, of equations of the form

$$f^{(s)} + A_{s-1}(z) f^{(s-1)} + \cdots + A_1(z) f' + A_0(z) f = 0,$$

where $A_i(z) (i \in \overline{0, s-1}; s \in \mathbb{N} \backslash \{1\})$ are analytic functions. Many papers are devoted to this one and adjoining problems. See, for example, [3, 17, 7, 12, 9, 8].

Let us consider the equation

$$f^{(s)} + A_0(z) f = 0. \tag{13}$$

Problem. Let (λ_k) be a given sequence of distinct complex numbers having no finite limit points and satisfying (2). Does there exist an entire function $A_0(z)$ such that the differential equation (13) possesses a solution f with the zero-sequence (λ_k)? How does $M_f(r)$ increase?

Note, that in [21] that problem was considered for the case $s = 2$. The following statements were proved.

Corollary 1. Let (λ_k) be a sequence of complex numbers satisfies a condition (2) for some $\Delta < 1$. Then there exists an entire function $A_0(z)$ such that $f'' + A_0(z) f = 0$ possesses a solution f, an entire function with the zero-sequence (λ_k), and

$$\ln M_f(r) \le N(r) + c_3 \exp\left(2N(\rho_3 r)\right)$$

for each $\rho_3 \in (\rho_1; 1)$.

Corollary 2. If (λ_k) satisfies (2) for some $\Delta < 1$, then there exists an entire function $A_0(z)$ such that $f'' + A_0(z)f = 0$ possesses a solution f with the zero-sequence (λ_k) and

$$|A_0(z)| \le c_2 \exp\left(4N(R_1 r)\right)$$

for each $R_1 \in (\rho_1; 1)$ and all $r > 0$.

Following [2], we can set the solution of (13) in the form $f(z) = L(z)e^{w(z)}$, where L is an entire function with the simple zeros at points λ_n (for example, $L(z) = \prod_{n=1}^{\infty}(1 - z/\lambda_n)$), w is some entire functions from the class (4) (or (12)). It is not difficult to see (for the case $s = 2$, it was proved in [21, 9]) that $f = Le^w$ is a solution of (13) with zero sequence (λ_n), if w is a solution of the multiple interpolation problem

$$w^{(k+1)}(\lambda_n) = b_{n,k}, \qquad k \in \{0, \dots, s-2\}, n \in \mathbb{N},$$

where

$$\begin{aligned}
b_{n,0} =& w'(\lambda_n) = -\frac{L''(\lambda_n)}{2L'(\lambda_n)},\\
b_{n,1} =& w''(\lambda_n) = -\frac{L'''(\lambda_n)}{3L'(\lambda_n)} + \left(\frac{L''(\lambda_n)}{2L'(\lambda_n)}\right)^2,\\
&\vdots\\
b_{n,k} =& w^{(k+1)}(\lambda_n)\\
=& -\frac{1}{(k+2)L'(\lambda_n)} \times \left(L^{(k+2)}(\lambda_n) + \sum_{j=0}^{k-2} C_{k-2}^j w^{(j+2)} L^{(k-j)}(\lambda_n) \right.\\
&\left. + \sum_{j=0}^{k-1} C_k^j \left(2L^{(k+1-j)} w^{(j+1)} + L^{(k-j)}(w'^2)^{(j)}\right)(\lambda_n) \right).
\end{aligned} \tag{14}$$

This problem leads to (1) if we put $g(z) = w'(z)$. From (14), applying Lemmas 2 and 3, we have the next inequality

$$|b_{n,k}| \le c_4 \left(\frac{n}{|\lambda_n|}\right)^{k+1}$$

for every $k \in \overline{0, s-1}$ and $n \in \mathbb{N}$. Thus, $b_{n,k}$, $k \in \overline{0, s-1}$, satisfy the conditions (3) of Theorem 1, which proves the existence of the function w

from the class (4), and consequently, the existence of solution $f = Le^w$ of equation (13).

Therefore, in analogue of Corollaries 1 and 2, the next assertion is true.

Theorem 2. Let (λ_k) be a sequence of complex numbers satisfying condition (2) for some $\Delta < 1$. Then there exists an entire function $A_0(z)$ such that for every $s \geq 2$, equation (13) possesses an entire solution f with the sequence of zeros (λ_k) and for every $\rho_1 \in (q; 1)$ and all $r > 0$, we have

$$\ln M_f(r) \leq c_2 \exp((s-1)N(r) + N(\rho_1 r))$$

or

$$\ln M_f(r) \leq c_2 \exp(sN(\rho_3 r))$$

for every $\rho_3 \in (\rho_1; 1)$ and all $r > 0$.

In addition, $A_0(z) = -\frac{f^{(s)}(z)}{f(z)}$ satisfies the growth estimate

$$M_{A_0}(r) \leq c_3(s) \exp\left(s^2 N(\rho_3 r)\right)$$

for every $\rho_3 \in (\rho_1; 1)$ and all $r > 0$.

5 Conclusion

Problems of multiple interpolation in the classes of entire functions have been investigated in the works of many authors, for example [15, 16, 5, 10, 6, 1, 4].

In this article, there was shown that interpolation problem (1) has the unique solution in class (3) when interpolation knots grow fast (satisfy condition (2)).

Applying counting function $(N(r))$ of the sequence (λ_n) in the definition of the mentioned classes is the peculiarity of this result.

Also, the obtained result was applied to the oscillation problem of the differential equation $f^{(s)} + A_0(z)f = 0$. So, there was shown existing an entire function $A_0(z)$ such that the previous differential equation possesses a solution f with the zero-sequence (λ_k) and there was found growth order of f.

6 Acknowledgments

The author thanks the Reviewers for a careful reading of this paper, which significantly improved the previous version of the article.

References

1. Abanina, T.I.*Interpolation problem in spaces of entire functions of arbitrarily rapid growth,* Vinniti no. 5440–5488, Moscow, 1988 (in Russian).
2. Bank S. *A note on the zero–sequences of solutions of linear differential equations* Results in Mathematics, 13 (1988), 1–10.
3. Bank S. and Laine I. *On the oscillation theory of $f'' + A(z)f = 0$ where A is entire* Trans. Am. Math. Soc., 273(1) (1982), 287–307.
4. Bozhenko, O.A. and Maljutin, K.G. *The multiple interpolation problem in the class of entire functions of zero order*, Proceedings of the Institute of Mathematics of NAS of Ukraine, 10 (4-5) (2013), 412–423 (in Russian).
5. Bratishchev, A.V. *The interpolation problem in certain classes of entire functions* Sib. Mat. J. 17(1) (1976), 30–40 (in Russian).
6. Bratishchev, A.V. and Korobeinik, Yu.F. *The multiple interpolation problem in the space of entire functions of given proximate order* Izv. Akad. Nauk SSSR, Ser. Mat., 40(5) (1976), 1102–1127 (in Russian).
7. Cao T.-B. and Yi, H.-X. *The growth of solutions of linear differential equations with coefficients of iterated order in the unit disc*, J. Math. Anal. Appl., 319 (2006), 1, 278–294.
8. Chyzhykov, I. and Long J. *On non-separated zero sequences of solutions of a linear differential equation,* Proc. Edinburgh Math. Soc., 64 (2021), 247–261.
9. Chyzhykov, I. and Sheparovych, I. *Interpolation of analytics functions of moderate growth in the unit disk and zeros of solutions of a linear differential equation,* J. Math. Analysis and Applications, 2014, 319–333.
10. Firsakova, O.S.*Some problems connected with interpolation by means of entire functions*, Mathematics of the USSR-Izvestiya 120(3) (1958), 447–480 (in Russian).
11. Gel'fond, A. O. *Calculus of finite differences* Nauka, Moscow, 1967, Russian. English translation: Hindustan Publishing, Delhi, 1971.
12. Gröhn, J. and Heittokangas, J. *New findings on Bank-Sauer approach in oscillatory theory*, Constr. Approx., 35 (2012), 345–361.
13. Heittokangas J. and Laine I. *Solutions of $f'' + A(z)f = 0$ with prescribed sequences of zeros* Acta Mathematica Universitatis Comenianae, 74(2) (2005), 287–307.
14. Kaz'min Yu, A. *On some Gel'fond's problem*, Mat. sb. 132(4), 1973, 520–543 (in Russian).

15. Lapin, G.P. *Interpolation in the class of entire functions of finite order* Izv. vys. uch. zavedenij. Matematika, 12(5) (1959), 146–153 (in Russian).

16. Lapin, G.P. *On entire functions of finite order which assume, together with their derivatives given values at given points* Sib. Mat. J. 6(6) (1965), 1267–1281 (in Russian).

17. Tu J. and Long, T. *Oscillation of complex high order linear differential equations with coefficients of finite iterated order*, Electron. J. Qual. Theory Differ. Equ., 66 (2009), 1–13.

18. Vynnyts'kyi, B.V. *On the representation of entire functions by series* $\sum_{n=1}^{\infty} d_n f(\lambda_n z)$ Mat. zamet., 27(3) (1980), 361–372 (in Russian).

19. Vynnyts'kyi, B.V. *On the representation of entire functions by series* $\sum_{n=1}^{\infty} d_n f(\lambda_n z)$ Mat. zamet., 29(4) (1981), 503–516 (in Russian).

20. Vynnyts'kyi, B.V. *On the construction of an entire function of arbitrary order with given asymptotic properties*, Ukr. Mat. J., 38(2) (1986), 143–148 (in Russian).

21. Vynnyts'kyi, B.V. and Sheparovych I.B. *On interpolation problem with derivative in the space of entire functions with fast-growing interpolation knots* Mathem. Studii, 51(1) (2019) 50–58.

How to cite this article
Sheparovych, I. Multiple interpolation with the fast-growing knots in the class of entire functions and its application. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 131-144. doi: 10.22067/ijnao.2021.70074.1028

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 145-162
DOI:10.22067/ijnao.2021.70805.1040
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Fitted numerical method for singularly perturbed semilinear three-point boundary value problem

M. Gebeyehu, H. Garoma* and A. Deressa

Abstract

We consider a class of singularly perturbed semilinear three-point boundary value problems. An accelerated uniformly convergent numerical method is constructed via the exponential fitted operator method using Richardson extrapolation techniques to solve the problem. To treat the semilinear term, we use quasi-linearization techniques. The numerical results are tabulated in terms of maximum absolute errors and rate of convergence, and it is observed that the present method is more accurate and ε-uniformly convergent for $h \geq \varepsilon$, where the classical numerical methods fail to give a good result. It also improves the results of the methods existing in the literature. The method is shown to be second-order convergent independent of perturbation parameter ε.

AMS subject classifications (2020): 65L11, 65L12, 65L20.

Keywords: Singularly perturbed problem; Semilinear problem; Exponential fitted operator; Three-point boundary problem.

---

*Corresponding author
Received 4 June 2021; revised 12 September 2021; accepted 27 September 2021
Mulgeta Gebeyehu,
Department of Mathematics, College of Natural Sciences,Jimma university, Jimma, Ethiopia. e-mail: mgebeyehu20@gmail.com

Habtamu Garoma
Department of Mathematics, College of Natural Sciences,Jimma university, Jimma, Ethiopia. e-mail: habte200@gmail.com

Ayana Deressa
Department of Mathematics, College of Natural Sciences,Jimma university, Jimma, Ethiopia. e-mail: ayanaderesa@gmail.com

1 Introduction

Singularly perturbed differential equations are typically characterized by the presence of a small positive parameter ε multiplying some or all of the highest order terms in differential equations. Such types of problems arise frequently in mathematical models of different areas of physics, chemistry, biology, engineering science, economics, and even sociology.

Boundary value problems including nonlocal conditions, which connect the values of the unknown solution at the boundary with values in the interior, are known as nonlocal boundary value problems. The study of this kind of problem was initiated by Il'in and Miseev [18, 19], motivated by the work of Bitsadze and Samarskii [4] on nonlocal linear elliptic boundary value problems. These problems have been used to represent mathematical models of a large number of phenomena, such as problems of semiconductors in electronics, the vibrations of a guy wire of a uniform cross-section, heat transfer problems, problems of hydromechanics, catalytic processes in chemistry and biology, the diffusion-drift model of semiconducting devices, and some other physical phenomena; see [1, 17, 25]. The existence and uniqueness of the solutions to nonlocal boundary value problems have been studied by many authors [3, 20]. Some approaches for the numerical solution of singularly perturbed nonlocal boundary value problems have been proposed in [2, 6, 7, 11, 12, 13, 16, 21, 26]. Uniformly convergent numerical methods of order second and high for solving different singularly perturbed problems have been studied in [5, 9, 10, 14, 23, 27]. What makes the problem under consideration challenging is that, it contains the perturbation parameter in both diffusion and convection terms, that it contains semilinear source term, and that it has three-point nonlocal boundary condition. To the best of our knowledge, the problem under consideration has not been done using the exponentially fitted operator method. Motivated by the paper [8], we develop a uniformly convergent numerical method for solving the singularly perturbed problem under consideration.

2 Definition of the problem

Consider the following singularly perturbed problem of the form

$$Ly := \varepsilon^2 y''(x) + \varepsilon a(x) y'(x) - f(x, y(x)) = 0, \quad 0 < x < l, \tag{1}$$

with the given conditions

$$y(0) = A, \tag{2}$$

$$L_0 y := y(l) - \phi(y(l_1)) = 0, \quad 0 < l_1 < l, \tag{3}$$

where $\varepsilon, 0 < \varepsilon \ll 1$ is the perturbation parameter, A is a given constant, and the functions $a(x) \geq a > 0$ and $f(x, y)$ are sufficiently smooth on $[0, l]$ and $[0, l] \times R$, respectively. Moreover $0 < b \leq \frac{\partial f}{\partial y} \leq \beta < \infty$ and $|\frac{d\phi}{dy}| \leq k < 1$.

The solution of $y(x)$ of (1)–(3) has boundary layers at $x = 0$ and $x = l$ for ε near 0.

Equations of this type arise in mathematical problems in many areas of mechanics and physics. Among these are the Navier–Stokes equations of fluid flow at high Reynolds number, mathematical models of liquid crystal materials and chemical reactions, shear in second-order fluids, control theory, electrical networks, and other physical models [23, 24].

To obtain the numerical solution to (1)–(3), the well-known Newton's quasi-linearization techniques were used. This technique allows us to linearize the semilinear problem into a linear problem, whose solution $y^{(p)}(x)$ with a proper initial guess $y^{(0)}$ will converge to the original solution $y(x)$.

So, after applying quasi-linearization technique for $f(x, y(x))$, we rewrite (1)–(3) in the form

$$Ly := \varepsilon^2 y''(x) + \varepsilon a(x) y'(x) - b(x) y(x) = R(x), \qquad x \in \Omega = (0, l), \tag{4}$$

with the given conditions

$$y(0) = A, \tag{5}$$

$$L_0 y := y(l) - \phi(y(l_1)) = 0, \quad 0 < l_1 < l, \tag{6}$$

where $b(x) = \frac{\partial f}{\partial y}(x, \eta y(x))$, $0 < \eta < 1$, and $R(x) = f(x, \eta y(x)) - \eta y(x) b(x)$.

In this article, we analyze an exponentially fitted operator scheme with Richardson extrapolation techniques on uniform mesh for the numerical solution of (4)–(6). Uniform convergence is proved in the discrete maximum norm. Finally, we formulate the algorithm for solving the discrete problem and give the illustrative numerical results.

3 Properties of continuous solution

The following lemmas are necessary for the existence and uniqueness of the solution and for the problem to be well-posed.

Lemma 1. Let $y(x)$ be the solution of (1)–(3). If $a \in C^1[0, l]$, then the following inequality

$$\|y\|_\infty \leq C_0 \tag{7}$$

holds, where $C_0 = (1-\gamma)^{-1}(|A|+|B|+b^{-1}\|R\|_\infty)$, $B = \phi(0,)$ $R(x) = f(x, 0)$, and $\|y\|_\infty = \max_{[0,l]} |y(x)|$.

Proof. For proof refer [8]. □

Lemma 2. Let y_ε be the solution of (P_ε). Then, for $k = 0, 1, 2, 3, 4$,

$$| y_{\varepsilon}^{(k)}(x) | \leq C \left(1 + \varepsilon^{-k} \left(\exp\left(\frac{-c_0 x}{\varepsilon}\right) + \exp\left(\frac{-c_1(l-x)}{\varepsilon}\right)\right)\right), \quad x \in [0,l], \tag{8}$$

holds fore the solution $y(x)$ provided that $\frac{\partial f}{\partial y} - \varepsilon a'(x) \geq b$ and $\frac{\partial f}{\partial x} \leq C$ for $x \in [0,l]$ and $|y| \leq C$, where $c_0 = \frac{1}{2}[\sqrt{a^2(0)+4b} + a(0)]$ and $c_1 = \frac{1}{2}[\sqrt{a^2(l)+4b} - a(l)]$.

Proof. For proof refer [8]. □

4 Formulation of numerical scheme

We subdivide the domain $\Omega = [0,1]$ into N equal number of subintervals, each of length h. In this article, we develop the fitted operator finite difference method to find a numerical solution to the problem (4). We use the theory in the asymptotic method for developing the exponential factor.

In order to evaluate the fitting factor, we divide both sides of (4) by ε, and we obtain

$$\begin{cases} Ly(x) \equiv \varepsilon y''(x) + a(x)y' + p(x)y(x) = g(x), & 0 < x < \ell, \\ y(0) = A, & \\ L_0 y := y(\ell) = \theta, & 0 < \ell_1 < \ell, \end{cases} \tag{9}$$

where $p(x) = -\frac{b(x)}{\varepsilon}$, $g(x) = \frac{R(x)}{\varepsilon}$, and $\theta = \varphi(y(\ell_1))$.

4.1 Left end boundary layer problem

The boundary layer occurs on the left side of the domain, that is, near $x = 0$. To find the numerical solution of (9), we use the theory applied in the asymptotic method for solving singularly perturbed boundary value problems. From the theory of singular perturbations given by O'Malley [24] and using Taylor's series expansion for $a(x)$ about $x = 0$ and restriction to their first terms, we get the asymptotic solution as

$$y(x) = y_0(x) + \frac{a(0)}{a(x)}(A - y_0(0))e^{-\int_0^x \left(\frac{a(x)}{\varepsilon} - \frac{p(x)}{a(x)}\right)dx} + O(\varepsilon), \tag{10}$$

where $y_0(x)$ is the solution of the reduced problem (obtained by setting $\varepsilon = 0$) of (9) given by $a(x)y_0'(x) + p(x)y_0(x) = g(x)$, with $y(0) = A$ and $y_0(\ell) = \theta$, where $\theta = \varphi(y(l_1))$.

By taking Taylor series expansion for $a(x)$ and $p(x)$ about the point "0" and simplifying them, we obtain

$$y(x) = y_0(x) + (A - y_0(0))e^{-\left(\frac{a(0)}{\varepsilon} - \frac{p(0)}{a(0)}\right)x} + O(\varepsilon). \tag{11}$$

Now we divide the interval $[0, 1]$ into N equal subinterval of mesh size $h = \frac{1}{N}$ so that $x_i = ih$, $i = 0, 1, 2 \ldots, N$. From (11), we have

$$y(ih) = y_0(ih) + (A - y_0(0))e^{-\left(\frac{a(0)}{\varepsilon} - \frac{p(0)}{a(0)}\right)ih} + O(\varepsilon).$$

Taking limit as $h \to 0$ on both sides, we have

$$\lim_{h\to 0} y(ih) = y_0(0) + (A - y_0(0))e^{-\left(\frac{a(0)}{\varepsilon} - \frac{p(0)}{a(0)}\right)i\rho} + O(\varepsilon). \tag{12}$$

Also, we discretized $x_{i+1} = (i + 1)h = ih + h$ from (11), we obtain

$$y(ih + h) = y_0(ih + h) + (A - y_0(0))e^{-\left(\frac{a(0)}{\varepsilon} - \frac{p(0)}{a(0)}\right)ih+h} + O(\varepsilon).$$

Taking limit as $h \to 0$ on both sides, we have

$$\lim_{h\to 0} y(ih + h) = y_0(0) + (A - y_0(0))e^{-\left(\frac{a(0)}{\varepsilon} - \frac{p(0)}{a(0)}\right)i\rho+\rho} + O(\varepsilon). \tag{13}$$

Similarly for $x_{i-1} = (i - 1)h = ih - h$ from (11), we have

$$y(ih - h) = y_0(ih - h) + (A - y_0(0))e^{-\left(\frac{a(0)}{\varepsilon} - \frac{p(0)}{a(0)}\right)ih-h} + O(\varepsilon).$$

Taking limit as $h \to 0$ on both sides, we have

$$\lim_{h\to 0} y(ih - h) = y_0(0) + (A - y_0(0))e^{-\left(\frac{a(0)}{\varepsilon} - \frac{p(0)}{a(0)}\right)i\rho-\rho} + O(\varepsilon). \tag{14}$$

To handle the effect of the perturbation parameter, artificial viscosity (exponentially fitting factor $\sigma(\rho)$) is multiplied on the term containing the perturbation parameter as

$$\varepsilon\sigma(\rho)y_i'' + a(x_i)y_i' + p(x_i)y_i = g(x_i), \tag{15}$$

with boundary conditions $y(0) = A$ and $y(\ell)=\theta$. Next, we consider the difference approximation of (9) on a uniform grid $\overline{\Omega}^N = \{x_i\}_{i=0}^N$ and denote $h = x_{i+1} - x_i$.

Using the Taylor series expansion for $y(x)$ about the point x_i, this can be written as

$$y(x_i - h) \approx \; y_{i-1} = y_i - hy_i' + \frac{h^2}{2!}y_i'' + O(h^3), \tag{16}$$

$$y(x_i + h) \approx \; y_{i+1} = y_i + hy_i' + \frac{h^2}{2!}y_i'' + O(h^3). \tag{17}$$

For any mesh function y_i, from (16) and (17), we obtain the following finite difference operator:

$$\begin{cases} D^- y_i = \frac{y_i - y_{i-1}}{h}, \\ D^+ y_i = \frac{y_{i+1} - y_i}{h}, \\ D^0 y_i = \frac{y_{i+1} - y_{i-1}}{2h}, \\ D^+ D^- y_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}. \end{cases} \tag{18}$$

Then, applying the central and forward finite difference formula for the second and the first derivative, respectively, on (9), we get

$$\varepsilon\sigma(\rho)(D^+ D^- y(x_i)) + a(x_i)(D^+ y(x_i)) + p(x_i)y(x_i) = g(x_i). \tag{19}$$

Using the difference operator in (19), we have

$$L_\varepsilon^N y_i = g_i, \tag{20}$$

with boundary conditions $y(0) = A$ and $y(\ell) = \theta$.

From (19), we have

$$\varepsilon\sigma(\rho)\left(\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}\right) + a(x_i)\left(\frac{y_{i+1} - y_i}{h}\right) + p(x_i)y_i = g(x_i), \quad 1 \le i \le \frac{N}{2}, \tag{21}$$

where $\rho = \frac{h}{\varepsilon}$.

Multiplying (21) by h, considering h small, and truncating the term $(g(x_i) - p(x_i)y_i)h$, we get

$$\frac{\sigma}{\rho}\big(y_{i-1} - 2y_i + y_{i+1}\big) + a(x_i)\big(y_{i+1} - y_i\big) = h(g(x_i) - p_i y_i). \tag{22}$$

By evaluating the limit of (22) as h approaches to zero and $\lim_{h\to 0} h(g_i - p_i y_i) = 0$, we get

$$\sigma = -a(0)\rho\frac{\lim_{h\to 0}(y_{i+1} - y_i)}{\lim_{h\to 0}(y_{i+1} - 2y_i + y_{i-1})}. \tag{23}$$

Substituting (12), (13), and (14) into (23) for solving the fitting parameter σ and simplifying it, we obtain

$$\sigma_1 = \frac{\rho a(0)}{4}\left[\frac{(1 - e^{-(\frac{a^2(0) - \varepsilon b(0)}{a(0)}\rho)})}{\sinh^2[(\frac{a^2(0) - \varepsilon b(0)}{2a(0)}\rho)]}\right]. \tag{24}$$

4.2 Right end boundary layer problem

The boundary layer occurs on the right side of the domain, that is, near $x = 1$. Assume that $\overline{\Omega}^N$ denotes the partition of $[0, \ell]$ into N subintervals such that $0 = x_0,\ x_1, \ldots, x_N = 1$ with $x_i = ih,\ h = \frac{1}{N}\ \ i = 0, 1, 2, \ldots, N$. To find the numerical solution of (9), we use the theory applied in the asymptotic method for solving singularly perturbed boundary value problems. Assume that the problem has the right layer. From the theory of singular perturbations given by O'Malley [24] and using Taylor's series expansion for $a(x)$ about $x = 1$ and restriction to their first terms, we get the asymptotic solution as

$$y(x) = y_0(x) + \frac{a(1)}{a(x)}(\theta - y_0(1))e^{-\int_x^1 \left(\frac{a(x)}{\varepsilon} - \frac{p(x)}{a(x)}\right)dx} + O(\varepsilon), \tag{25}$$

where $y_0(x)$ is the solution of the reduced problem (obtained by setting $\varepsilon = 0$) of (9) given by $a(x)y_0'(x)+p(x)y_0(x) = g(x)$, with $y_0(1) = \theta$. By taking Taylor series expansion for $a(x)$ and $b(x)$ about the point "1" and simplifying them, we obtain

$$y(x) = y_0(x) + (\theta - y_0(1))e^{-\left(\frac{a(1)(1-x)}{\varepsilon}\right)} + O(\varepsilon). \tag{26}$$

Now we divide the interval $[0, 1]$ into N equal subintervals of mesh size $h = \frac{1}{N}$ so that $x_i = ih$, $i = 0, 1, 2 \ldots, N$. From (26), we have

$$\lim_{h\to 0} y(ih) = y_0(0) + (\theta - y_0(1))e^{\left(-a(1)(\frac{1}{\varepsilon} - i\rho)\right)} + O(\varepsilon). \tag{27}$$

Also, we discretization $x_{i+1} = (i + 1)h = ih + h$ From (26), we have

$$\lim_{h\to 0} y(ih + h) = y_0(0) + (\theta - y_0(1))e^{\left(-a(1)(\frac{1}{\varepsilon} - i\rho)\right)} + O(\varepsilon). \tag{28}$$

Similarly for $x_{i-1} = (i - 1)h = ih - h$, we have

$$\lim_{h\to 0} y(ih - h) = y_0(0) + (\theta - y_0(1))e^{\left(-a(1)(\frac{1}{\varepsilon} - i\rho)\right)} + O(\varepsilon). \tag{29}$$

Now we consider the second-order upwind finite difference scheme from (15). We have

$$\varepsilon\sigma(\rho)\left(\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}\right) + a_i\left(\frac{y_i - y_{i-1}}{h}\right) + p_i y_i = g_i. \tag{30}$$

Multiplying (30) by h, considering h small, and truncating the term $(g(x_i) - p(x_i)y_i)h$, we get

$$\frac{\sigma}{\rho}\left(y_{i-1} - 2y_i + y_{i+1}\right) + a_i\left(y_i - y_{i-1}\right) = h(g_i - p_i y_i). \tag{31}$$

By evaluating the limit of (31) as h approaches to zero and using $\lim_{h\to 0} h(g_i - p_i y_i) = 0$, we get

$$\sigma = -a(1)\rho \frac{\lim_{h\to 0}(y_i - y_{i-1})}{\lim_{h\to 0}(y_{i-1} - 2y_i + y_{i+1})}. \tag{32}$$

Substituting (27), (28), and (29) into (32) for solving the fitting parameter σ and rearranging it, we obtain

$$\sigma_2 = \frac{\rho a(1)}{4}\left[\frac{(e^{-(\frac{a^2(1)-\varepsilon b(1)}{a(1)}\rho)} - 1)}{\sinh^2[(\frac{a^2(1)-\varepsilon b(1)}{2a(1)}\rho)]}\right]. \tag{33}$$

Case (1): For the left layer, from the discrete form of (4) on the domain $[1, \frac{N}{2}]$, we have

$$\varepsilon^2 y_i'' + \varepsilon a_i y_i' - b_i y_i = R_i. \tag{34}$$

Using the central and forward finite difference formula on (34), we have

$$\varepsilon^2 \sigma_1(\rho)\left(\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}\right) + \varepsilon a_i\left(\frac{y_{i+1} - y_i}{h}\right) - b_i y_i = R_i.$$

Hence, the required finite difference scheme becomes

$$\left(\frac{\varepsilon^2\sigma_1(\rho)}{h^2}\right)y_{i-1} + \left(\frac{-2\varepsilon^2\sigma_1(\rho)}{h^2} - \varepsilon\frac{a_i}{h} - b_i\right)y_i + \left(\frac{\varepsilon^2\sigma_1(\rho)}{h^2} + \varepsilon\frac{a_i}{h}\right)y_{i+1} = R_i. \tag{35}$$

The numerical scheme in (35) can be written as a three-term recurrence relation by

$$E_i y_{i-1} + F_i y_i + G_i y_{i+1} = H_i, \qquad 1 \le i \le \frac{N}{2}, \tag{36}$$

where

$$E_i = \left(\frac{\varepsilon^2\sigma_1(\rho)}{h^2}\right), \qquad F_i = \left(\frac{-2\varepsilon^2\sigma_1(\rho)}{h^2} - \varepsilon\frac{a_i}{h} - b_i\right),$$
$$G_i = \left(\frac{\varepsilon^2\sigma_1(\rho)}{h^2} + \varepsilon\frac{a_i}{h}\right), \qquad H_i = R_i.$$

Case (2): For the right layer, from the discrete form of (4) on the domain $(\frac{N}{2}, N-1]$, we have

$$\varepsilon^2 y_i'' + \varepsilon a_i y_i' - b_i y_i = R_i. \tag{37}$$

Using the central and backward finite difference formula on (37), we have

$$\varepsilon^2 \sigma_2(\rho)\left(\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}\right) + \varepsilon a_i\left(\frac{y_i - y_{i-1}}{h}\right) - b_i y_i = R_i.$$

Hence, the required finite difference scheme becomes

$$\left(\frac{\varepsilon^2\sigma_2(\rho)}{h^2}-\varepsilon\frac{a(x_i)}{h}\right)y_{i-1}+\left(\frac{-2\varepsilon^2\sigma_2(\rho)}{h^2}+\varepsilon\frac{a_i}{h}-b_i\right)y_i+\left(\frac{\varepsilon^2\sigma_2(\rho)}{h^2}\right)y_{i+1}=R_i. \tag{38}$$

The numerical scheme in (38) can be written as a three-term recurrence relation by

$$E_iy_{i-1}+F_iy_i+G_iy_{i+1}=H_i, \qquad \frac{N}{2}<i\leq N-1, \tag{39}$$

where

$$E_i=\left(\frac{\varepsilon^2\sigma_2(\rho)}{h^2}-\varepsilon\frac{a_i}{h}\right), \qquad F_i=\left(\frac{-2\varepsilon^2\sigma_2(\rho)}{h^2}+\varepsilon\frac{a_i}{h}-b_i\right),$$
$$G_i=\left(\frac{\varepsilon^2\sigma_2(\rho)}{h^2}\right), \qquad H_i=R_i.$$

Next, in order to treat the boundary conditions, the following equations are obtained: For $i=1$, (36) becomes

$$F_1y_1+G_1y_2=H_1-E_1A. \tag{40}$$

For $i=N-1$, (36) becomes

$$E_{N-1}y_{N-2}+F_{N-1}y_{N-1}+G_{N-1}y_N=H_{N-1}. \tag{41}$$

From (6) and (41), we obtain

$$E_{N-1}y_{N-2}+F_{N-1}y_{N-1}+G_{N-1}\phi(y_{\frac{N}{2}})=H_{N-1}. \tag{42}$$

Therefore, the problem in (1) with given boundary conditions in (2)–(3), can be solved using the scheme (36), (39), (40), and (42), which forms an $N\times N$ system of algebraic equations.

5 Uniform convergence analysis

In this section, we need to show the discrete scheme in (36) and (39) satisfies the discrete minimum principle and uniform convergence.

Lemma 3 (Discrete Minimum Principle)**.** Let Y_i be any mesh function that satisfies $Y_0\geq 0, Y_N\geq 0$ and $L_\varepsilon Y_i\leq 0$, $i=1,2,3,\ldots,N-1$. Then $Y_i\geq 0$, for $i=0,1,2,\ldots,N$.

Proof. The proof is by contradiction. Let j be such that $Y_j=\min_i Y_i$, and suppose that $Y_j\leq 0$. Clearly, $j\notin\{0,N\}$. $Y_{j+1}-Y_j\geq 0$ and $Y_j-Y_{j-1}\leq 0$.

Therefore,

$$\begin{aligned} L_\varepsilon^N Y_j =& \varepsilon\left(\frac{Y_{j+1}-2Y_j+Y_{j-1}}{h^2}\right)+a_j\left(\frac{Y_{j+1}-Y_j}{h}\right)+p_jY_j,\\ =& \frac{\varepsilon}{h^2}(Y_{j+1}-2Y_j+Y_{j-1})+\frac{a_j}{h}(Y_{j+1}-Y_j)+p_jY_j,\\ =& \frac{\varepsilon}{h^2}((Y_{j+1}-Y_j)-(Y_j-Y_{j-1}))+\frac{a_j}{h}(Y_{j+1}-Y_j)+p_jY_j,\\ \geq& 0, \end{aligned} \tag{43}$$

where the strict inequality holds if $Y_{j+1}-Y_j>0$. This is a contradiction, and therefore $Y_j\geq 0$. Since j is arbitrary, we have $Y_i\geq 0$, for $i=0,1,2,\ldots,N$. □

We proved above that the discrete operator L_ε^N satisfies the minimum principle. Next, we analyze the uniform convergence analysis.

Theorem 1. Let $y(x_i)$ and Y_i be, respectively, the exact solution of (9) and numerical solutions of (20). Then for sufficiently large N, the following parameter uniform error estimate holds:

$$|L^N(y(x_i)-Y_i)|\leq Ch\left(1+\varepsilon^{-4}\max_{1\leq i\leq N-1}\left(\exp\left(\frac{-a(l-x_i)}{\varepsilon}\right)\right)\right). \tag{44}$$

Proof. Let us consider the local truncation error defined as

$$L^N(y(x_i)-Y_i)=\varepsilon\sigma_2(\rho)(y''(x_i)-D^+D^-y(x_i))+a(x_i)(y'(x_i)-D^-y(x_i)), \tag{45}$$

where $\varepsilon\sigma_2(\rho)=\frac{\rho a(1)}{4}\left[\frac{(e^{-(\frac{a^2(1)-\varepsilon b(1)}{a(1)}\rho)}-1)}{\sinh^2[(\frac{a^2(1)-\varepsilon b(1)}{2a(1)}\rho)]}\right]$, since $\rho=\frac{h}{\varepsilon}$.

For fixed $N=\frac{1}{h}$, taking the limit for $\varepsilon\to 0$, we obtain

$$\lim_{\varepsilon\to 0}\varepsilon\sigma(\rho)=\lim_{\varepsilon\to 0}\frac{\rho a(1)}{4}\left[\frac{(e^{-(\frac{a^2(1)-\varepsilon b(1)}{a(1)}\rho)}-1)}{\sinh^2[(\frac{a^2(1)-\varepsilon b(1)}{2a(1)}\rho)]}\right]=Ch,$$

where C is constant independent of h and ε.

Using Taylor series expansion, the bound for $y(x_{i-1})$ and $y(x_{i+1})$ at x_i is obtained as

$$\begin{cases} y(x_{i-1})=y(x_i)-hy'(x_i)+\frac{h^2}{2!}y''(x_i)-\frac{h^3}{3!}y^{(3)}(x_i)+\frac{h^4}{4!}y^{(4)}(x_i)+O(h^5),\\ y(x_{i+1})=y(x_i)+hy'(x_i)+\frac{h^2}{2!}y''(x_i)+\frac{h^3}{3!}y^{(3)}(x_i)+\frac{h^4}{4!}y^{(4)}(x_i)+O(h^5). \end{cases}$$

We obtain the bound for

$$\begin{cases} |D^+D^-y(x_i)|\leq C|y''(x_i)|,\\ |y''(x_i)-D^+D^-y(x_i)|\leq Ch^2|y^{(4)}(x_i)|. \end{cases} \tag{46}$$

Similarly, for the first derivative term, we have

$$|y'(x_i) - D^+y(x_i)| \leq Ch|y''(x_i)|, \tag{47}$$

where $|y^{(k)}(x_i)| = \sup_{x_i \in (x_0, x_N)} |y^{(k)}(x_i)|, \quad k = 2, 3, 4.$

Using the bounds in (46) and (47), we obtain

$$|L^N(y(x_i) - Y_i) \leq Ch^3|y^{(4)}(x_i)| + C|y''(x_i)|.$$

Now, using the bounds for the derivatives of the solution in Lemma (2), we have

$$\begin{aligned} |L^N(y(x_i) - Y_i)| &\leq Ch^3 \left(1 + \varepsilon^{-4} \exp\left(\frac{-a(l - x_i)}{\varepsilon}\right)\right) \\ &\quad + Ch \left(1 + \varepsilon^{-2} \exp\left(\frac{-a(l - x_i)}{\varepsilon}\right)\right) \\ &\leq Ch \left(1 + \varepsilon^{-4} \max_{1 \leq i \leq N-1} \exp\left(\frac{-a(l - x_i)}{\varepsilon}\right)\right) \end{aligned}$$

since $\varepsilon^{-4} \geq \varepsilon^{-2}$. □

Lemma 4. For a fixed mesh and for $\varepsilon \to 0$, it holds

$$\lim_{\varepsilon \to 0} \max_{1 \leq i \leq N-1} \frac{\exp\left(\frac{-ax_i}{\varepsilon}\right)}{\varepsilon^m} = 0, \quad m = 1, 2, 3, \ldots,$$

$$\lim_{\varepsilon \to 0} \max_{1 \leq i \leq N-1} \frac{\exp\left(\frac{-a(1-x_i)}{\varepsilon}\right)}{\varepsilon^m} = 0, \quad m = 1, 2, 3, \ldots.$$

Proof. Consider the partition $[0, 1] := \{0 = x_0 < x_1 < \cdots < x_{N-1} < x_N = 1\}$ for the interior grid points, we have

$$\max_{1 \leq i \leq N-1} \frac{\exp\left(\frac{-ax_i}{\varepsilon}\right)}{\varepsilon^m} \leq \frac{\exp\left(\frac{-ax_1}{\varepsilon}\right)}{\varepsilon^m} = \frac{\exp\left(\frac{-ah}{\varepsilon}\right)}{\varepsilon^m},$$

$$\max_{1 \leq i \leq N-1} \frac{\exp\left(\frac{-a(1-x_i)}{\varepsilon}\right)}{\varepsilon^m} \leq \frac{\exp\left(\frac{-a(1-x_{N-1})}{\varepsilon}\right)}{\varepsilon^m} = \frac{\exp\left(\frac{-ah}{\varepsilon}\right)}{\varepsilon^m},$$

as $x_1 = 1 - x_{N-1} = h$. The repeated application of L'Hospital's rule gives

$$\lim_{\varepsilon \longrightarrow 0} \frac{\exp\left(\frac{-ah}{\varepsilon}\right)}{\varepsilon^m} = \lim_{r = \frac{1}{\varepsilon} \to \infty} \frac{r^m}{\exp(ahr)} = \lim_{r = \frac{1}{\varepsilon} \to \infty} \frac{m!}{(ah)^m \exp(ahr)} = 0.$$

□

Theorem 2. Let $y(x_i)$ and Y_i be the exact solution of (9) and numerical solutions of (20), respectively. Then, the following error bound holds

$$\sup_{0<\varepsilon<<1} |(y(x_i) - Y_i)| \leq Ch. \tag{48}$$

Proof. By substituting the results in Lemma (4) into Theorem (1) and applying the discrete minimum principle, we obtain the required bound. □

Richardson extrapolation

Here, we apply the Richardson extrapolation technique to accelerate the rate of convergence of the proposed scheme. Richardson Extrapolation is a convergence acceleration technique, which involves a combination of two computed approximations of solution. From (48), we have

$$y(x_i) - Y_i \leq Ch, \tag{49}$$

where $y(x_i)$ and Y_i are exact and approximate solutions, respectively, and C is constant independent of ε and h. Let Ω^{2h} be the mesh obtained by bisecting each mesh interval in Ω^h and denote the approximation of the solution on Ω^{2h} by Y_i^{2h}. From (49), we have

$$y(x_i) - Y_i \cong Ch + R^h, \quad x_i \in \Omega^N, \tag{50}$$

So, this works for any $\frac{h}{2} \neq 0$ gives

$$y(x_i) - Y_i^{2h} \cong C\frac{h}{2} + R^{2h}, \quad x_i \in \Omega^{2h}, \tag{51}$$

where the remainders, R^h and R^{2h} are $O(h^2)$. Equations (50) and (51) lead to

$$y(x_i) - (2Y_i^{2h} - Y_i) \cong Ch^2,$$

which gives that

$$Y_i^{ext} = 2Y_i^{2h} - Y_i \tag{52}$$

is also an approximate solution. The total truncation error for the approximate solution in (52) becomes

$$\sup_{0<\varepsilon<<1} |(y(x_i) - Y_i)| \leq Ch^2. \tag{53}$$

6 Numerical example and results

To validate the established theoretical results, we perform numerical experiments using the model problems of the form in (1)–(3).

Example 1. Consider the model singularly perturbed boundary value problem

$$\varepsilon^2 y''(x) + \varepsilon(1 - \frac{x}{2})y'(x) - (1 + x^2 + 2y - \exp(-y)) = 0, \quad 0 < x < 1,$$

subject to the boundary conditions

$$y(0) = 0, \quad \phi(y) = \cos(\frac{\pi y}{4}) + 2, \quad l_1 = \frac{1}{2}.$$

Having $y_j \equiv y_j^h$ (the approximated solution obtained via fitted operator finite difference method) for different values of h and ε, the maximum errors. Since the exact solution is not available, the maximum errors (denoted by E_ε^h) are evaluated using the double mesh principle for fitted operator finite difference methods using formula

$$E_\varepsilon^N := \max_{0 \le j \le N} |y_j^h - y_{2i}^{2h}|.$$

Furthermore, we will tabulate the ε-uniform error

$$E^N = \max_{0 < \varepsilon \le 1} E_\varepsilon^N.$$

The numerical rate of convergence and the ε-uniform rate of convergence are computed using the formulas

$$R^N = \frac{\log(E^N) - \log(E^{2N})}{\log(2)}.$$

ε	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
10^{-4}	3.3238e-03	8.4445e-04	2.1268e-04	5.3355e-05	1.3361e-05
10^{-8}	3.3238e-03	8.4445e-04	2.1268e-04	5.3355e-05	1.3361e-05
10^{-12}	3.3238e-03	8.4445e-04	2.1268e-04	5.3355e-05	1.3361e-05
10^{-16}	3.3238e-03	8.4445e-04	2.1268e-04	5.3355e-05	1.3361e-05
10^{-20}	3.3238e-03	8.4445e-04	2.1268e-04	5.3355e-05	1.3361e-05
E^N	3.3238e-03	8.4445e-04	2.1268e-04	5.3355e-05	1.3361e-05
R^N	1.9767	1.9893	1.9950	1.9976	

Table 1: Maximum absolute errors and rates of convergence for Example 1.

	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
Present					
E^N	3.3238e-03	8.4445e-04	2.1268e-04	5.3355e-05	1.3361e-05
R^N	1.9767	1.9893	1.9950	1.9976	
Method in [8]					
E^N	0.01388073	0.00955129	0.00550127	0.00314426	0.00176922
R^N	0.69	0.75	0.81	0.91	

Table 2: Comparison of maximum absolute errors and rate of convergence for Example 1 at different number of mesh points.

	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
After					
E^N	3.3238e-03	8.4445e-04	2.1268e-04	5.3355e-05	1.3361e-05
R^N	1.9767	1.9893	1.9950	1.9976	
Before					
E^N	6.1832e-02	3.2578e-02	1.6711e-02	8.4619e-03	4.2576e-03
R^N	0.9245	0.9631	0.9817	0.9909	

Table 3: Maximum absolute errors and rate of convergence before and after Richardson extrapolation for Example 1 at different number of mesh points.

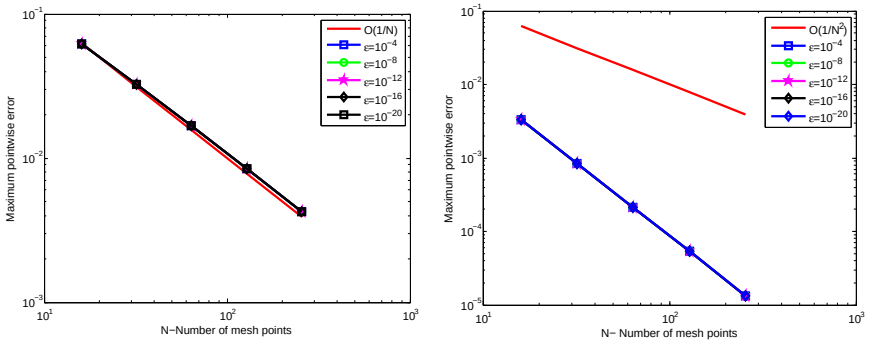


Figure 1: ε-uniform convergence of exponentially fitted operator method in log-log scale before and after Richardson Extrapolation, respectively.

7 Discussion and conclusion

This study introduced an accelerated exponentially fitted finite difference numerical method for solving singularly perturbed semilinear differential equations with nonlocal boundary conditions. The behavior of the continuous solution of the problem was studied and shown that it satisfies the continuous stability estimate and the derivatives of the solution are also bounded. The numerical scheme was developed on the uniform mesh using exponential fitted operator method in the given differential equation. The stability of the

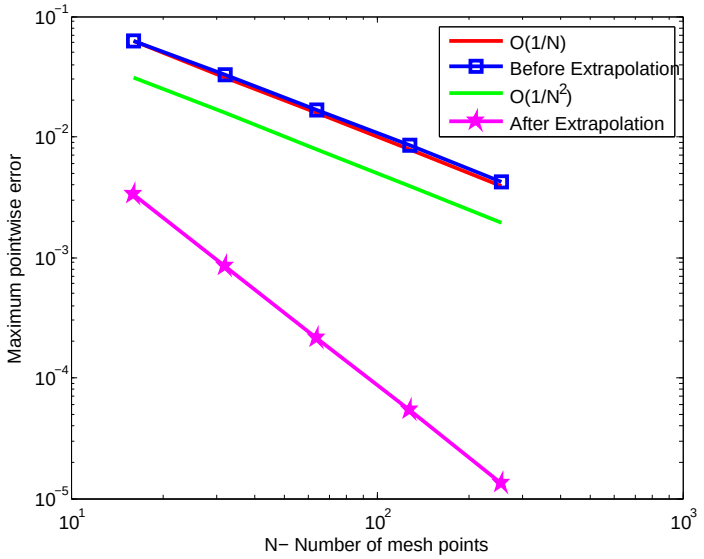


Figure 2: The ε-uniform convergence of the method in log-log scale before and after extrapolation at the same plot for Example 1.

developed numerical method was established, and its uniform convergence was proved. To validate the applicability of the method, a model problem was considered for numerical experimentation for different values of the perturbation parameter and mesh points. The numerical results were tabulated in terms of maximum absolute errors, numerical rate of convergence, and uniform errors (see Table 1) and to show the performance of our scheme after Richardson extrapolation, we compared both the maximum absolute error and the rate of convergence of the scheme before and after Richardson extrapolation (see Table 3). To visualize our result more, we plotted the log-log plot of maximum absolute error before and after Richardson in different plane and in the same plane in Figures 1 and 2, respectively. The method was shown to be ε-uniformly convergent with the order of convergence $O(h^2)$. The performance of the proposed scheme was investigated by comparing it with prior study (see Table 2). The proposed method gave more accurate, stable, and ε-uniform numerical results.

Acknowledgments

The author wishes to express their thanks to Jimma University, College of Natural Sciences, for technical support and the authors of the literature for the provided scientific aspects and ideas for this work.

References

1. Adzic, N. *Spectral approximation and nonlocal boundary value problems*, Novi Sad J. Math. 30 (3) (2000) 1–10.

2. Amiraliyev, G.M. and Cakir, M. *Numerical solution of the singularly perturbed problem with nonlocal condition*, Appl. Math. Mech. (English Edition) 23 (7) (2002) 755–764.

3. Benchohra, M. and Ntouyas, S.K.*Existence of solutions of nonlinear differential equations with nonlocal conditions*, J. Math. Anal. Appl. 252 (2000) 477–483.

4. Bitsadze, A.V. and Samarskii, A.A. *On some simpler generalization of linear elliptic boundary value problem*, Doklady Akademii Nauk SSSR. 185 (1969) 739–740.

5. Cakir, M.*Uniform second-order difference method for a singularly perturbed three-point boundary value problem*, Adv. Differential Equ. 2010 (2010) 1–13.

6. Cakir, M. *A numerical study on the difference solution of singularly perturbed semilinear problem with integral boundary condition*, Math. Model. Anal. 21 (5) (2016) 644–658.

7. Cakir, M. and Amiraliyev, G.M. *A finite difference method for the singularly perturbed problem with nonlocal boundary condition*, Appl. Math. Comput. 160 (2005) 539–549.

8. Cakir, M. and Amiraliyev, G.M.*Numerical solution of a singularly perturbed three-point boundary value problem*, Int. J. Comput. Math. 84(10) (2007) 1465–1481.

9. Cen, Z.*A second-order finite difference scheme for a class of singularly perturbed delay differential equations*, Int. J. Comput. Math. 87 (1) (2008) 173–185.

10. Cen, Z., Le, A. and Xu, A. *Parameter-uniform hybrid difference for solutions and derivatives in singularly perturbed initial value problems*, J. Comput. Appl. Math. 320(20) (2017) 176–192.

11. Ciegis, R. *The numerical solution of singularly perturbed nonlocal problem (in Russian)*, Lietuvas Matematica Rink 28 (1988) 144–152.

12. Ciegis, R. *The difference scheme for problems with nonlocal conditions*, Informatica (Lietuva) 2 (1991) 155–170.

13. Cimen, E. and Cakir, M. *Numerical treatment of nonlocal boundary value problem with layer behavior*, Bull. Belg. Math. Soc. Simon Stevin. 24 (2017) 339–352.

14. Clavero, C., Gracia, J.L. and Lisbona, F. *High order methods on Shishkin meshes for singular perturbation problems of convection-diffusion type*, Numer. Algorithms 22 (1999) 73–97.

15. Doolan, E.P., Miller, J.J.H. and Schilders, W. H.A. *Uniform numerical method for problems with initial and boundary layers*, Dublin, Boole Press, 1980.

16. Du, Z. and Kong, L. *Asymptotic solutions of singularly perturbed second-order differential equations and application to multi-point boundary value problems*, Appl. Math. Lett. 23 (9)(2010) 980–983.

17. Herceg, D. and Surla, K. *Solving a nonlocal singularly perturbed nonlocal problem by splines in tension*, Univ. u Novom Sadu Zb. Rad. Prirod.-Mat. Fak. Ser. Mathematics, 21 (2) (1991) 119–132.

18. Il'in, V.A. and Moiseev, E.I. *Nonlocal boundary value problem of the first kind for a Sturm-Liouville operator in its differential and finite difference aspects*, Differ. Equ. 23 (7)(1987) 803–810.

19. Il'in, V.A. and Moiseev, E.I. *Nonlocal boundary value problem of the second kind for a Sturm-Liouville operator*, Differ. Equ. 23 (1987) 979–987.

20. Jankowski, T.*Existence of solutions of differential equations with nonlinear multipoint boundary conditions*, Comput. Math. Appl. 47 (2004) 1095–1103.

21. Kudu, M. and Amiraliyev, G.M. *Finite difference method for a singularly perturbed differential equations with integral boundary condition*, Int. J. Math. Comput. 26 (3) (2015) 72–79.

22. Kumar, M. and Rao, C.S. *Higher order parameter-robust numerical method for singularly perturbed reaction-diffusion problems*, Appl. Math. Comput. 216 (2010) 1036–1046.

23. Nayfeh, A.H. *Introduction to perturbation techniques*, John Wiley and Sons, New York, NY, USA, 1993.

24. O'Malley, R.E. *Singular perturbation methods for ordinary differential equations*, New York, Springer Verlag, 1991.

25. Petrovic, N.*On a uniform numerical method for a nonlocal problem.* Univ. u Novom Sadu Zb. Rad. Prirod.-Mat. Fak. Ser. Mathematics, 21 (2) (1991) 133–140.

26. Sapagovas, M. and Ciegis, R.*Numerical solution of nonlocal problems (in Russian)*, Lietuvas Matematica Rink 27 (1987) 348–356.

27. Zheng, Q., Li, X. and Gao, Y. *Uniformly convergent hybrid schemes for solutions and derivatives in quasilinear singularly perturbed BVPs*, Appl. Numer. Math. 91 (2015) 46–59.

How to cite this article
Debela, H., Gebeyehu, M., Negessa, A. Fitted numerical method for singularly perturbed semilinear three-point boundary value problem. Iranian Journal of Numerical Analysis and Optimization, 2022; 12(1): 145-162. doi: 10.22067/ijnao.2021.70805.1040

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 163-171
DOI:10.22067/ijnao.2021.67913.1010
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Connected bin packing problem on traceable graphs

A. Nejoomi and A. Dolati*

Abstract

We consider a new extension of the bin packing problem in which a set of connectivity constraints should be satisfied. An undirected graph with a weight function on the nodes is given. The objective is to pack all the nodes in the minimum number of unit-capacity bins, such that the induced subgraph on the set of nodes packed in each bin is connected. After analyzing some structural properties of the problem, we present a linear time approximation algorithm for this problem when the underlying graph is traceable. We show that the approximation factor of this algorithm is 2 and this factor is tight. Finally, concerning the investigated structural properties, we extend the algorithm for more general graphs. This extended algorithm also has a tight approximation factor of 2.

AMS subject classifications (2020): 90C27, 68Q25, 68W25.

Keywords: Bin Packing Problem; Connectivity; Complexity Theory; Approximation Algorithms.

1 Introduction

The bin packing problem (BPP) has been studied extensively in the literature due to its numerous applications and its intriguing combinatorial structure. The BPP consists of packing a given set of items with different weights into

*Corresponding author
Received 14 January 2021; revised 15 July 2021; accepted 27 September 2021
Ahmad Nejoomi
Department of Computer Science, Shahed University, Tehran, Iran.
e-mail: a.nejoomi@shahed.ac.ir

Ardeshir Dolati
Department of Computer Science, Shahed University, Tehran, Iran.
e-mail: dolati@shahed.ac.ir

a minimum number of same bins so that the total weight packed in any bin does not exceed the capacity. The BPP is known to be NP-hard [9].

Various extensions of the BPP have been studied in the literature. In the *BPP with conflicts* [12], any two conflict items must not be packed in a same bin. The *BPP with precedence constraints* [6] requires that the set of given precedences among the items is satisfied. The aim in *BPP with fragile objects* [2] is to pack the given weighted fragile items into the minimum number of bins, such that the total weights of each bin do not exceed the fragility of the most fragile item in the bin. Böhm et al. [3] introduced the *colored BPP* in which each item has a color. As additional condition to the capacity constraint, we require that no two items of the same color are packed into a bin consecutively.

There are huge number of algorithms for the BPP in the literature. Among these methods, some basic and simple algorithms were mostly considered by many researchers.

- The next fit (NF) algorithm. Consider the items sorted in any predefined order. In each round, if an item fits inside the current bin, then the item is packed inside it. Otherwise, the current bin is closed, a new bin is opened, and the current item is packed inside the new current bin. The running time can be bounded by $O(n\log(n))$, where n is the number of items. The NF algorithm is a 2-approximation algorithm; see [13].

- The first fit (FF) algorithm. Suppose that the items are sorted in any order. In this order, the next item is always packed into the first bin, where it fits. The running time can be bounded by $O(n\log(n))$, where n is the number of items. The FF algorithm is a 1.7-approximation algorithm; see [8].

- The first fit decreasing (FFD) algorithm. Sort the items in non-increasing order of their sizes and then apply the first fit algorithm. Same as FF, the running time is bounded by $O(n\log(n))$, where n is the number of items. Dosa [7] showed that $FFD(I) \leq 11/9OPT(I) + 6/9$ and that this bound is tight.

In the connected variant of BPP, connected bin packing problem (CBPP), we consider the items as the nodes of a graph. For each instance of the CBPP, a graph $G = (V, E)$ and a set of infinite number of unit capacity bins are given. Each node $v \in V$ is characterized by the weight $w(v)$. Like BPP, the total weight packed in any bin could not exceed the unit. In this extension of BPP, each pair of nodes belonging to the same bin must be connected. In other words, the induced subgraph on the set of nodes packed in each bin is connected. The aim is to pack of the nodes in a minimum number of bins.

The CBPP arises in many real-world applications. When we need to pack a large number of spatial objects into a small number of contiguous regions [1], we have a CBPP.

Clearly, the basic BPP is a special case of the CBPP, in which the graph G is a complete graph. Since the BPP is strongly NP-hard, by restriction, the CBPP is also strongly NP-hard. By similar conclusions, the CBPP does not admit an approximation algorithm with a ratio smaller than $\frac{3}{2}$.

In other view, the CBPP is related to the balanced connected partitioning problem [4]. Then the CBPP can be seen as a variant of *problem of partitioning a weighted graph to connected subgraphs of almost uniform size* [11]. In this problem, the aim is to partition a given node weighted graph with integer weights in a minimum (or maximum) number of connected components. When the underlying graph is a tree, then the problem could be handled by the $O(n^6)$ time algorithm [10].

The "covering problem with capacitated subtrees" [5] is another related problem, in which vertices of a graph have to be covered by rooted subtrees.

The rest of the paper is organized as follows. In Section 2, we give a formal definition of the CBPP and some complexity issues. In Section 3, we present some approximation algorithms for the CBPP on traceable graphs and some general graphs.

2 Preliminaries and approximation

In this Section, after a formal definition of the CBPP and preliminaries, we give some structural properties of the problem.

Definition 1. Let $G = (V, E)$ be an undirected graph with a weight function $w : V \to [0, 1]$ on its nodes. The aim is to pack all the nodes in the minimum number of unit capacity bins, such that the set of nodes packed in each bin induces a connected subgraph (see Figure 1, for example).

For a given CBPP (i.e., a graph $G = (V, E)$ with a weight function $w : V \to [0, 1]$) and any partition of nodes $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$, we define the merged graph $G_{\mathcal{S}} = (V_{\mathcal{S}}, E_{\mathcal{S}})$ associated with $\mathcal{S}$ as follows: We merge elements of each subset S_i in the single node v_{S_i} after deleting edges among them and merge the resulting multiple parallel edges in a single edge. The weight of each merged nodes, denoted by $W_{\mathcal{S}}(\cdot)$, is the sum of weights of nodes merged in it (i.e., for all $S_i \in \mathcal{S}$, $W_{\mathcal{S}}(v_{S_i}) = \sum_{v \in S_i} w(v)$).

Theorem 1. Suppose that a graph $G = (V, E)$ with a weight function $w : V \to [0, 1]$ is given. Let $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$ be a feasible solution and let $G_{\mathcal{B}} = (V_{\mathcal{B}}, E_{\mathcal{B}})$ be the corresponding merged graph. If for each edge $(v_i, v_j) \in E_{\mathcal{B}}$, $W_{\mathcal{B}}(v_i) + W_{\mathcal{B}}(v_j) \geq 1$ and $G_{\mathcal{B}}$ has a maximal matching of size $\lfloor \frac{m}{2} \rfloor$, then $m \leq 2m^* - 1$, where m^* is the size of optimal packing.

Proof. Suppose that M is such matching. Then

$$\text{for all } (i, j) \in M, \quad W_{\mathcal{B}}(v_i) + W_{\mathcal{B}}(v_j) \geq 1.$$

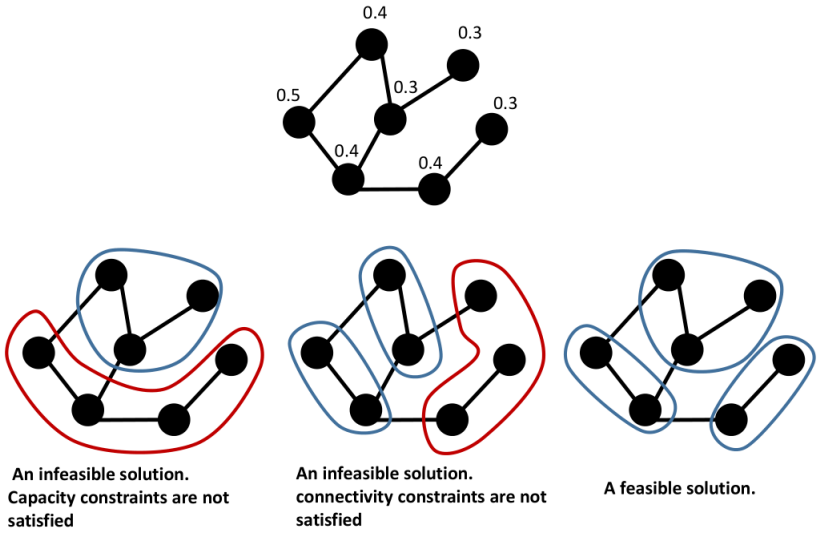


Figure 1: A CBBP example: Feasible and infeasible solutions

We have two cases:

- If m is an even number, then M is a maximum matching and

$$m^* \overset{\text{Trivial Upper Bound}}{\geq} \lceil \sum_{k \in V} W_B(v_k) \rceil \quad \geq \sum_{(i,j) \in M} (W_B(v_i) + W_B(v_j)) > \frac{m}{2}.$$

- If m is an odd number, then M is a matching of size $\frac{m-1}{2}$ and covers $m-1$ nodes. If we denote the uncovered node by v, then at least $\frac{m-1}{2}$ bins are needed for packing $G - v$. Also,

$$m^* > \frac{m-1}{2} \rightarrow m^* \geq \frac{m-1}{2} + 1 \rightarrow m \leq 2m^* - 1.$$

□

Now, we investigate a trivial feasible solution to a CBPP instance that the underling graph is union of two disjoint subgraphs and a cutting edge. Suppose that two corresponding feasible solutions to CBPP's on this subgraphs are given. A trivial feasible solution to the CBPP on a hole graph is the union of the given subsolutions. In the following theorem, we prove some results for this solution.

Theorem 2. Suppose that $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are two disjoint connected graphs with weight functions w_1 and w_2 in $[0, 1]$ and that B_1^* and B_2^* are optimal packings for them, respectively. Let $G = (V, E)$ be a graph with $V = V_1 \cup V_2$ and $E = E_1 \cup E_2 \cup \{(x, y)\}$, where $x \in V_1$ and $y \in V_2$. Let B^* be an optimal packing for G. Then

$$|B^*| \geq |B_1^*| + |B_2^*| - 1.$$

Proof. If x and y are packed separately in B^*, then the proof is trivial. Hence we suppose that they are packed in B_e, where $e = (x, y)$. Let $|B^*| = k_1 + k_2 + 1$, where k_1 and k_2 are the number of bins used for packing $G_1 - B_e$ and $G_2 - B_e$, respectively. Therefore,

$$k_1 \geq |B_1^*| - 1,$$

$$k_2 \geq |B_2^*| - 1,$$

so

$$|B^*| = k_1 + k_2 + 1 \geq |B_1^*| + |B_2^*| - 1$$

□

This result can be extended as the following theorem.

Theorem 3. Suppose that $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are two disjoint connected graphs with weight functions w_1 and w_2 in $[0, 1]$ and that B_1 and B_2 are feasible packings with approximation factor of α_1 and α_2 for them, respectively. Let $G = (V, E)$ be any graph with $V = V_1 \cup V_2$ and $E = E_1 \cup E_2 \cup \{(x, y)\}$, where $x \in V_1$ and $y \in V_2$. Then $B = B_1 \cup B_2$ is a feasible packing with approximation factor of $\alpha_{\max} = \max\{\alpha_1, \alpha_2\}$.

Proof. Let B^*, B_1^*, B_2^* are optimal packings of G, G_1, G_2, respectively. By Theorem 2,

$$|B^*| \geq |B_1^*| + |B_2^*| - 1.$$

On the other hand, $|B_1| \leq \alpha_1 |B_1^*|$ and $|B_2| \leq \alpha_2 |B_2^*|$. Without loss of generality, suppose that $\alpha_1 \geq \alpha_2$. Then we have

$$\alpha_1 |B^*| \geq \alpha_1 |B_1^*| + \frac{\alpha_1}{\alpha_2} \alpha_2 |B_2^*| - 1.$$

Thus

$$\alpha_1 |B^*| \geq |B_1| + \frac{\alpha_1}{\alpha_2} |B_2| - 1 \geq |B_1| + |B_2| - 1.$$

□

Corollary 1. The complete graph $G_0 = (V_0, E_0)$ and the tree $T = (V_T, E_T)$ with weight functions w_0 and w_T in $[0, 1]$ are given. Let B_0 is a packing for the corresponding BPP with weight function w_1 obtained from the decreasing first fit algorithm. Let B_T be an optimal packing for a CBPP on T and let

$G = (V, E)$ be any graph with $V = V_0 \cup V_T$ and $E = E_0 \cup E_T \cup \{(x, y)\}$, where $x \in V_0$ and $y \in V_T$. Then $B = B_0 \cup B_T$ is a feasible packing with approximation factor of $\alpha = \frac{3}{2}$.

Above theorems can be extended to any constant number of subgraphs.

Theorem 4. Suppose that for $i = 1, \ldots, k$, a connected graph $G_i = (V_i, E_i)$ with a weight function w_i in $[0, 1]$ and an optimal packing B_i^* are given. Let $G = (V, E)$ be a graph with $V = \bigcup_{i=1}^{k} V_i$ and $E = \bigcup_{i=1}^{k} E_i \cup \{E_T\}$, where $G < E_T >$ is any subtree of G with exactly one node form each V_i. Let B^* be an optimal packing for G. Then

$$|B^*| \geq \sum_{i=1}^{k} |B_i^*| - k + 1.$$

Theorem 5. Suppose that for $i = 1, \ldots, k$, a connected graph $G_i = (V_i, E_i)$ with a weight function w_i in $[0, 1]$ and a feasible packing B_i with approximation factor of α_i are given. Let $G = (V, E)$ be a graph with $V = \bigcup_{i=1}^{k} V_i$ and $E = \bigcup_{i=1}^{k} E_i \cup \{E_T\}$, where $G < E_T >$ is a subtree of G with exactly one node form each V_i. Then $B = \bigcup_{i=1}^{k} B_i$ is a feasible packing with approximation factor of $\alpha_{\max} = \max_{i=1}^{k} \alpha_i$.

3 CBPP on traceable graphs

A Hamiltonian path is a path that visits each node of the graph exactly once. A traceable graph is a graph possessing a Hamiltonian path. In this section, we consider a CBPP on traceable graphs.

Lemma 1. Let $G = (V, E)$ be a weighted traceable graph with weight functions w in $[0, 1]$ and a Hamiltonian path $P = \{v_1, v_2, \ldots, v_n\}$. We use the next fit algorithm for the BPP with items $\{v_1, v_2, \ldots, v_n\}$ in this order. The obtained packing $\mathcal{B}$ is a feasible packing for G with approximation factor 2. This approximation factor is tight.

Proof. Obviously, $G_{\mathcal{B}}$ has a Hamiltonian path and so a maximal matching of size $\lfloor \frac{|\mathcal{B}|}{2} \rfloor$. By Theorem 1, the obtained packing $\mathcal{B}$ is a feasible packing for G with approximation factor 2. Figure 2 shows a tight example in which the Hamiltonian path $\{1, 2, \ldots, 2k + 1\}$ is given, where $\epsilon > 0$ is a very small number. □

Corollary 2. The traceable graph $G_0 = (V_0, E_0)$ and the tree $T = (V_T, E_T)$ with weight functions w_0 and w_T in $[0, 1]$ are given. Let B_0 be a packing for the CBPP obtained from next fit algorithm. Let B_T be an optimal packing

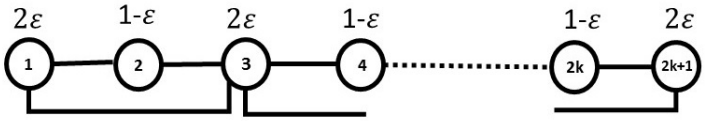


Figure 2: Tight-Example

for the CBPP on T and let $G = (V, E)$ be a graph with $V = V_0 \cup V_T$ and $E = E_0 \cup E_T \cup \{(x, y)\}$, where $x \in V_0$ and $y \in V_T$. Then $B = B_1 \cup B_2$ is a feasible packing with approximation factor of $\alpha = 2$. This approximation factor is tight.

Proof. The proof is clear. □

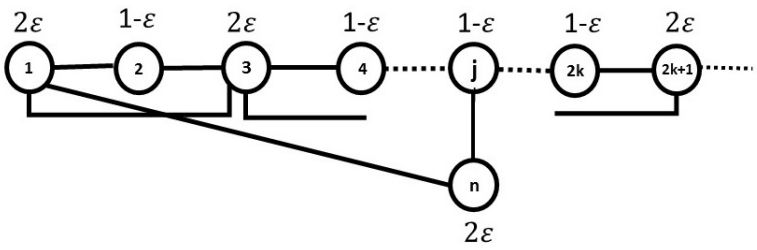


Figure 3: Tight-Example

Lemma 2. Suppose that $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are two disjoint connected traceable graphs with weight functions w_1 and w_2 in $[0, 1]$. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two feasible packings obtained by the next fit algorithm for G_1 and G_2 on their Hamiltonian path, respectively. Let $G = (V, E)$ be a graph with $V = V_1 \cup V_2$ and $E = E_1 \cup E_2 \cup \{(x, y)\}$, where $x \in V_1$ and $y \in V_2$. If $\mathcal{B}^*$ is an optimal packing for the CBPP on G and $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$, then $|\mathcal{B}^*| \geq \lfloor \frac{|\mathcal{B}|}{2} \rfloor + 1$.

Proof. Let $\mathcal{B}_1^*$ and $\mathcal{B}_2^*$ be optimal packings for G_1 and G_2, respectively. With a justification like Theorem 1, we could show that

$|\mathcal{B}_1^*| \geq \lfloor \frac{|\mathcal{B}_1|}{2} \rfloor + 1$,

$|\mathcal{B}_2^*| \geq \lfloor \frac{|\mathcal{B}_2|}{2} \rfloor + 1$.
So by Theorem 2, we have
$|\mathcal{B}^*| \geq |\mathcal{B}_1^*| + |\mathcal{B}_2^*| - 1 \geq \lfloor \frac{|\mathcal{B}_1|}{2} \rfloor + \lfloor \frac{|\mathcal{B}_2|}{2} \rfloor + 1 \geq \lfloor \frac{|\mathcal{B}|}{2} \rfloor + 1$.

□

Theorem 6. Suppose that a sequence of pair-wise disjoint connected traceable graphs $\{G_i = (V_i, E_i)\}_{i=1}^m$ is given. For each $i = 1, \ldots, m$, a weight function $w_i : V_i \to [0, 1]$ associated with graph G_i and a Hamiltonian path P_i is given. We assume that $\mathcal{B}_{P_i}$ is a feasible packing obtained by the next fit algorithm for G_i on its Hamiltonian path P_i. Let $G = (V, E)$ be any graph with $V = \bigcup_{i=1}^m V_i$, for all i $E_i \subset E$ and let the merged graph $G_S = (V_S, E_S)$ be a tree, where $S = \{V_1, V_2, \ldots, V_m\}$. If $\mathcal{B}^*$ is an optimal packing for the CBPP on G and $\mathcal{B} = \bigcup_{i=1}^m \mathcal{B}_{P_i}$, then $|\mathcal{B}^*| \geq \lfloor \frac{|\mathcal{B}|}{2} \rfloor + 1$.

4 Conclusion and future works

In this paper, we considered a generalization of the classical BPP, CBPP, in which, in addition to capacity constraints, a set of connectivity constraints has to be satisfied. The CBPP is a strongly NP-hard problem.

Suppose that $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are two disjoint node weighted graphs and that $G = G_1 \cup G_2 \cup (x, y)$, where $x \in V_1$ *and* $y \in V_2$. In this paper, we introduced a trivial feasible solution to the CBPP on G from solutions to CBPP's on G_1 and G_2 and proved some good results for the approximation factor of this solution, and we extended them. Then we presented a linear time approximation algorithm for this type of problems with approximation factor 2, when this disjoint graphs are traceable.

In perspective, our findings can be a starting point to tackle the CBPP. It would be interesting to try to design some approximation algorithms and lower bounds for the CBPP. Finally conclusion and some suggestions for more researches are given in this section.

References

1. AssunÇão, R.M., Neves, M.C., Câmara G. and Da Costa Freitas C.*Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees,* Int. J. Geogr. Inf. Sci. 20 (7) (2006) 797–811.

2. Bansal, N., Liu, Z. and Sankar, A. *Bin-packing with fragile objects and frequency allocation in cellular networks,* Wireless Netw. 15, (2009) 821–830.

3. Böhm, M., Dósa, G., Epstein, L., Sgall, J. and Veselý, P. *Colored bin packing: Online algorithms and lower bounds,* Algorithmica 80 (1) (2018) 155–184.

4. Chataigner, F., Salgado, L.R.B. and Wakabayashi, Y. *Approximation and inapproximability results on balanced connected partitions of graphs,* Discrete Math. Theor. Comput. Sci. 9 (1) (2007) 177–192.

5. Clautiaux, F., Guillot, J. and Pesneau, P. *Exact approaches for solving a covering problem with capacitated subtrees,* Comput. Oper. Res. 105 (2019) 85–101.

6. Dell'Amico, M., Díaz Díaz, J.C. and Iori, M. *The bin packing problem with precedence constraints,* Oper. Res. 60 (6) (2012) 1491–1504.

7. Dósa, G. *The tight bound of first fit decreasing bin-packing algorithm is FFD(I) 11/9OPT(I) + 6/9* Chen B., Paterson M., Zhang G. (eds) Combinatorics, Algorithms, Probabilistic and Experimental Methodologies. ESCAPE 2007. Lecture Notes in Computer Science, vol 4614. Springer, Berlin, Heidelberg, 2007.

8. Dósa, G. and Sgall, J. *First fit bin packing: a tight analysis,* 30th International Symposium on Theoretical Aspects of Computer Science, 538–549, LIPIcs. Leibniz Int. Proc. Inform., 20, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2013.

9. Garey, M.R. and Johnson, D.S. *Computers and intractability: A guide to the theory of NP-completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

10. Ito, T., Uno, T., Zhou, X. and Nishizeki, T. *Partitioning a weighted tree to subtrees of almost uniform size,* Algorithms and computation, 196–207, Lecture Notes in Comput. Sci., 5369, Springer, Berlin, 2008.

11. Ito, T., Zhou, X. and Nishizeki, T. *Partitioning a graph of bounded tree-width to connected subgraphs of almost uniform size,* J. Discrete Algorithms 4 (1) (2006) 142–154.

12. Jansen, K. *An approximation scheme for bin packing with conflicts,* J. Comb. Optim. 3 (4) (1999) 363–377.

13. Vazirani, V.V. *Approximation algorithms*, Berlin: Springer, 2003.

How to cite this article
Nejoomi, A., Dolati, A. Connected bin packing problem on traceable graphs. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 163-171. doi: 10.22067/ijnao.2021.67913.1010

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 173-186
DOI:10.22067/ijnao.2021.68160.1008
https://ijnao.um.ac.ir/

How to cite this article
Research Article

A heuristic algorithm to combat outliers and multicollinearity in regression model analysis

M. Roozbeh*, S. Babaie-Kafaki and M. Manavi

Abstract

As known, outliers and multicollinearity in the data set are among the important difficulties in regression models, which badly affect the least-squares estimators. Under multicollinearity and outliers' existence in the data set, the prediction performance of the least-squares regression method is decreased dramatically. Here, proposing an approximation for the condition number, we suggest a nonlinear mixed-integer programming model to simultaneously control inappropriate effects of the mentioned problems. The model can be effectively solved by popular metaheuristic algorithms. To shed light on importance of our optimization approach, we make some numerical experiments on a classic real data set as well as a simulated data set.

AMS subject classifications (2020): 62J05, 90C11, 90C59.

Keywords: Condition number; Linear regression; Penalty method; Metaheuristic algorithm; Nonlinear mixed-integer programming.

---

*Corresponding author
Received 7 January 2021; revised 27 July 2021; accepted 2 October 2021
Mahdi Roozbeh
Faculty of Mathematics, Statistics and Computer Science, P.O. Box: 35195-363, Semnan University, Semnan, Iran. e-mail: mahdi.roozbeh@semnan.ac.ir

Saman Babaie-Kafaki
Faculty of Mathematics, Statistics and Computer Science, P.O. Box: 35195-363, Semnan University, Semnan, Iran. e-mail: sbk@semnan.ac.ir

Monireh Manavi
Faculty of Mathematics, Statistics and Computer Science, P.O. Box: 35195-363, Semnan University, Semnan, Iran. e-mail: m.maanavi95@gmail.com

1 Introduction

As an effective statistical tool, multiple regression is widely and increasingly used in econometrics, engineering, social sciences, and so on. Generally, in a multiple regression model, the relationship between several independent (predictor) variables with a dependent (response) variable is investigated. The model is formulated by

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ is a vector of observations on the response variable, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times p}$ is a matrix of observations on the predictor variables, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ is a vector of unknown regression coefficients, and $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^\top$ is a vector of error terms with $\mathrm{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}$ and $\mathrm{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top) = \sigma^2\boldsymbol{I}_n$, where $\boldsymbol{I}_n$ is the unit matrix of order n and σ^2 is an unknown constant. The ordinary least-squares estimator (OLSE) of $\boldsymbol{\beta}$ is given by

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \ (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \\ &= \boldsymbol{S}^{-1}\boldsymbol{X}^\top\boldsymbol{y},\end{aligned}$$

where $\boldsymbol{S} = \boldsymbol{X}^\top\boldsymbol{X}$. In regression models with intercept β_0, we can simply rewrite (1) by considering $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^\top$ and $\boldsymbol{X} = (\boldsymbol{1}, \boldsymbol{X})$.

Some difficulties often appear in the regression analysis, such as collinearity between explanatory variables as well as outliers' existence in the data set. Generally, in the regression modeling an outlier is an observation point that fails to track the linear pattern of the data [10]. Outliers corrupt the OLSE; this fact motivated the researchers to investigate robust estimations [9]. As another problem in regression analysis, one may encounter with multicollinearity in the data set that is defined as the existence of nearly linear dependency among columns of the design matrix $\boldsymbol{X}$ [7]. In this situation, the matrix $\boldsymbol{S} = \boldsymbol{X}^\top\boldsymbol{X}$ has one or more small eigenvalues which causes the OLSE to perform poorly [8]. One effective approach to detect the outliers in a data set is the least trimmed squares (LTS) [9] in which the sum of smallest h-squared residuals is minimized rather than the complete sum of squares. Here, h is a prespecified threshold value and denotes the number of normal or good observations that are not outliers. If $z_i \in \{0, 1\}$ is the indicator demonstrating whether the ith observation is ordinary (nonoutlier) or not, then the model can be formulated by

$$\begin{aligned}&\min_{\boldsymbol{\beta},\boldsymbol{z}} \ \psi(\boldsymbol{\beta}, \boldsymbol{z}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top\boldsymbol{X}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \\ &s.t. \ \ \boldsymbol{z}^\top\boldsymbol{e} = h, \\ &\qquad z_i \in \{0, 1\}, \ i = 1, 2, \ldots, n,\end{aligned} \tag{2}$$

where $\boldsymbol{Z}$ is a diagonal matrix with the diagonal elements $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)^\top$, and $\boldsymbol{e} = (1, \ldots, 1)^\top \in \mathbb{R}^n$.

Let A be an arbitrary nonsingular matrix. Denoted by $K_p(A)$, the p-norm condition number of A defined by $\kappa_p(A) = \|A\|_p \|A^{-1}\|_p$ [12]. When A is positive definite and $p = 2$, then we get the spectral condition number, which can be determined as the ratio of the largest eigenvalue to the smallest eigenvalue of the matrix A [12]. As known, the condition number is an important factor to check the existence of the multicollinearity [12]. Based on this fact, Roozbeh, Babaie-Kafaki, and Naeimi Sadigh [8] developed an extension of the optimization model (2) to simultaneously control presence of the outliers and the multicollinearity in the data set; that is,

$$\begin{aligned} &\min_{\boldsymbol{\beta},\boldsymbol{z}} \ \psi(\boldsymbol{\beta}, \boldsymbol{z}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top \boldsymbol{Z}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \mu\kappa(\boldsymbol{X}^\top \boldsymbol{Z}\boldsymbol{X}) \\ &s.t. \ \ \boldsymbol{z}^\top \boldsymbol{e} = h, \\ &\qquad z_i \in \{0, 1\}, \ \ i = 1, 2, \ldots, n, \end{aligned} \tag{3}$$

in which $\kappa(\cdot)$ stands for the spectral condition number and $\mu > 0$ is called the penalty parameter. Here, the corresponding estimator is called the modified LTS counter multicollinearity (MLTSCM) estimator. In model (3), the additional term $\kappa(\boldsymbol{X}^\top \boldsymbol{Z}\boldsymbol{X}) > 0$ has been embedded as a penalty for generating inappropriate values for $z_1, z_2, \ldots, z_n$ and decreases the condition number of the final model to combat multicollinearity problem.

Here, we deal with a modified version of the regression model (3) with less computational cost in the objective function. This work is organized as follows. In Section 2, we introduce a penalized regression method. The method is then combined with the LTS method in Section 3, to combat the sparsity of the model. In Section 4, we deal with our approximate version of the mixed-integer nonlinear programming model (6) using a simple estimation of the spectral condition number. In Section 5, we provide some numerical experiments to show effectiveness of our model. Finally, concluding remarks are presented in Section 6.

2 Least absolute shrinkage and selection operator methodology

The amount of data we are faced with keeps growing. From around the late 1990s, wide data sets emerged, in which the number of variables far exceeds the number of observations. This was mainly due to our increasing ability to measure a large amount of information automatically [6].

Penalized regression can perform variable selection and prediction in a "Big Data" environment more effectively and efficiently in contrast to the other methods. Initially proposed by Tibshirani [11], the LASSO (least absolute shrinkage and selection operator) is based on minimizing mean squared

error, which is based on balancing the opposing factors of bias and variance to build the most predictive model. In fact, LASSO shrinks the regression coefficients toward zero by penalizing the regression model with an ℓ_1-norm penalty term, that is, the sum of the absolute value of the coefficients. LASSO regression is a simple technique to reduce model complexity and usually prevent over-fitting which may result from simple linear regression.

In the case of LASSO regression, the penalty term is embedded to force the coefficient estimates with minor contributions to the model to be exactly set to zero. This means that, LASSO can be also seen as an alternative to the subset selection methods for performing variable selection in order to reduce complexity of the model.

Ordinary least squares (OLS) regression chooses the coefficients by minimizing the residual sum of squares (RSS) as follows:

$$\min_{\boldsymbol{\beta}} RSS = \min_{\boldsymbol{\beta}} (\boldsymbol{y} - \hat{\boldsymbol{y}})^{\top} (\boldsymbol{y} - \hat{\boldsymbol{y}}) = \min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 \right\},$$

where $(x_{i1}, \ldots, x_{ip})$ can be called $\boldsymbol{x}_i^{\top}$. LASSO is an extension of the OLS, which adds a penalty to the RSS, being equal to sum of the absolute values of the nonintercept beta coefficients multiplied by the parameter λ that slows (when $\lambda < 1$) or accelerates (when $\lambda > 1$) the penalty. Therefore, the following optimization problem should be solved in LASSO problem:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

Figure 1 shows the constraint area of the LASSO method for $p = 2$, in which elliptical contours of the function are shown by the full. They are centered at the OLSE. The constraint region is the rotated square. LASSO solution is the first place that the contours touch the square, and this will sometimes occur at a corner, corresponding to a zero coefficient. LASSO is frequently used in practice since the ℓ_1 penalty allows us to shrink some coefficients to zero, that is, to produce sparse estimation models that are highly interpretable.

It is notable that increasing λ will increase bias and decrease variance. Likewise, decreasing λ reduces bias and increases variance. A big part of the building, the best models in LASSO deal with the bias-variance tradeoff. Bias refers to how correct (or incorrect) the model is. A very simple model that makes a lot of mistakes is said to have a high bias. A very complicated model that does well on its training data is said to have a low bias. There are several ways to choose the optimal λ, such as AIC, BIC, C_p, and so on [10]. For this purpose, one of the most popular methods is the cross validation (CV) method.

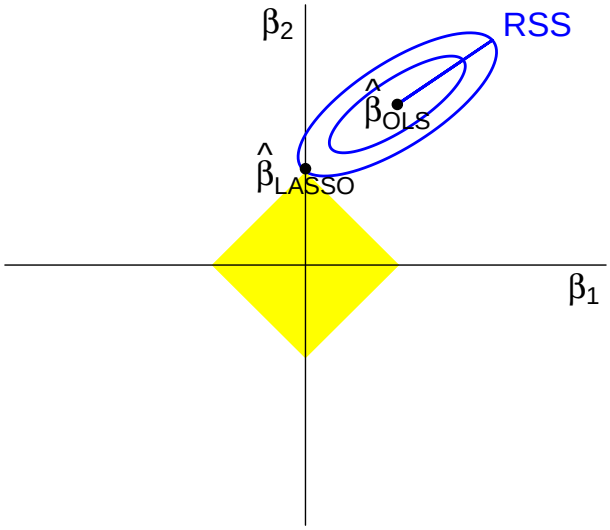


Figure 1: Constraint region of LASSO.

In order to find the optimal value of λ, a range of λ values is tested and the optimal value is chosen using CV, which involves

- separating the data into a training set and a test set;
- building the model in the training set;
- estimating the outcome in the test set using the model from the training set;
- calculating mean squared error (MSE) in the test set.

There are different types of cross validation method like Leave-P-Out, Leave-one-out, k-fold, standard k-fold, Monte Carlo, and so on [2]. One of the most important methods is the k-fold CV, which is one way to improve over the holdout method. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other $k-1$ subsets are put together to form a training set. Then, the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set $k-1$ times. The variance of the resulting estimate is reduced as k is increased. The

disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation. A variant of this method is to randomly divide the data into the test and the training set k different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

3 Sparse least trimmed squares method

As discussed, LASSO offers interpretable models but is not robust with respect to the outliers. The breakdown point of LASSO is $\frac{1}{n}$ (see [1] for more details); that is, only one single outlier can make the LASSO estimator completely unreliable. Therefore, robust alternatives are needed. In this situation, Alfons, Croux, and Gelper [1] suggested the sparse LTS estimator as follows:

$$\begin{aligned} &\min_{\boldsymbol{\beta},\boldsymbol{z}} \psi(\boldsymbol{\beta},\boldsymbol{z}) = \left\{(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top \boldsymbol{Z}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) + h\lambda\textstyle\sum_{j=1}^{p}|\beta_j|\right\} \\ &s.t.\ \ \boldsymbol{z}^\top \boldsymbol{e} = h, \\ &\qquad z_i \in \{0,1\},\ i = 1,2,\ldots,n, \end{aligned}$$

where λ is a penalty parameter. They showed that the breakdown point of this estimator is $\frac{n-h+1}{n}$.

4 A penalized nonlinear mixed-integer programming model in linear regression

As known, the condition number is an effective tool to check the existence of multicollinearity [12]. The OLSE performs poorly in the presence of multicollinearity. Also, the existence of multicollinearity may lead to wide confidence intervals for the individual parameters or their linear combinations, and can produce estimators with wrong signs.

As known, if $a_i, i = 1,\ldots,n$ denotes the ith column of A, then, for any i and j, it can be seen that

$$\kappa_p(A) \geq \frac{\|a_i\|_p}{\|a_j\|_p} \tag{4}$$

(see [12, Theorem 2.2.25]). Hence, we can write

$$\kappa_p(A) \geq \frac{\max\limits_{i=1,\ldots,n}\|a_i\|_p}{\min\limits_{j=1,\ldots,n}\|a_j\|_p} \stackrel{def}{=} \Psi_p(A). \tag{5}$$

Now, based on the above inequality, it is reasonable to select a set of h appropriate observations of the data that makes the computationally low cost term $\Psi_p(\boldsymbol{X}^\top \boldsymbol{Z}\boldsymbol{X})$ as small as possible. Hence, we propose the computationally low cost approximation $\kappa_p(A) \approx \Psi_p(A)$ and then, propose the following approximation of model (3):

$$\begin{aligned} &\min_{\boldsymbol{\beta},\boldsymbol{z}} \ \psi(\boldsymbol{\beta},\boldsymbol{z}) = (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top \boldsymbol{Z}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) + \mu\Psi(\boldsymbol{X}^\top \boldsymbol{Z}\boldsymbol{X}) \\ &s.t. \ \ \boldsymbol{z}^\top \boldsymbol{e} = h, \\ &\qquad z_i \in \{0,1\}, \ i = 1,2,\ldots,n. \end{aligned} \tag{6}$$

Here, the corresponding estimator is called the approximate LTS counter multicollinearity (ALTSCM) estimator. As seen, the given optimization model illustrates an NP-hard mixed-integer (having both continuous ($\boldsymbol{\beta}$) and (discrete) integer ($\boldsymbol{z}$) variables) nonlinear programming problem for which the classical methods are not practically efficient [4]. Note that in complexity theory, NP-hardness is viewed as strong evidence that a problem is not polynomially solvable [4]. As known, metaheuristic algorithms have attracted special attention in developing efficiently robust computational procedures for solving a vast variety of such problems. Among them there is the simulated annealing (SA) algorithm [4]. SA is a local search algorithm capable of escaping from local optima by use of random hill-climbing moves in the search process. It is very efficient in practice and well-developed in theory. Motivated by these, here we use the SA algorithm to approximately compute ALTSCM.

5 Numerical experiments

In this section, we investigate computational efficiency of the given estimator firstly on a real data set and then, on a simulated data set.

5.1 A real data set related to the riboflavin vitamin B2 production in Bacillus subtilis

To illustrate usefulness of the suggested strategies, we consider the data set about riboflavin vitamin B2 production in Bacillus subtilis, which can be found in R package "hdi" [5]. Riboflavin is one of the B vitamins, which are all water soluble. Riboflavin is naturally present in some foods, added to some food products, and available as a dietary supplement. This vitamin is an essential component of two major coenzymes, flavin mononucleotide (FMN; also known as riboflavin-5'-phosphate), and flavin adenine dinucleotide (FAD).

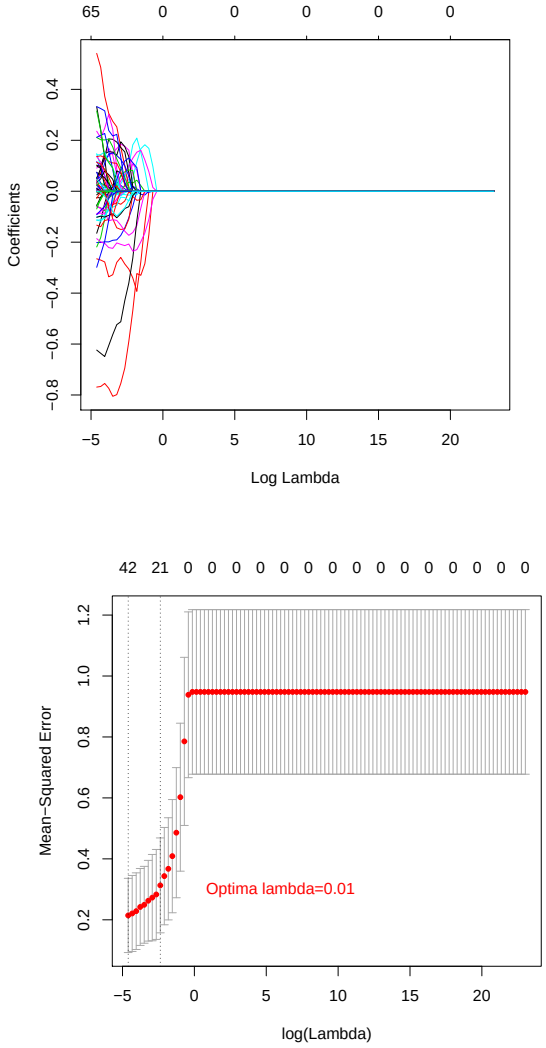


Figure 2: LASSO plots for the riboflavin data set.

There is a single real valued response variable, which is the logarithm of the riboflavin production rate. Furthermore, there are $p = 4088$ explanatory variables measuring the logarithm of the expression level of 4088 genes. There is also one rather homogeneous data set from $n = 7$ samples that were hybridized repeatedly during a fed batch fermentation process, where different engineered strains and strains grown under different fermentation conditions were analyzed. There is one rather homogeneous data set from $n = 71$ samples that were hybridized repeatedly during a fed batch fermentation process, where different engineered strains and strains grown under different fermentation conditions were analyzed. In Figure 2, the 10-fold cross-validation and

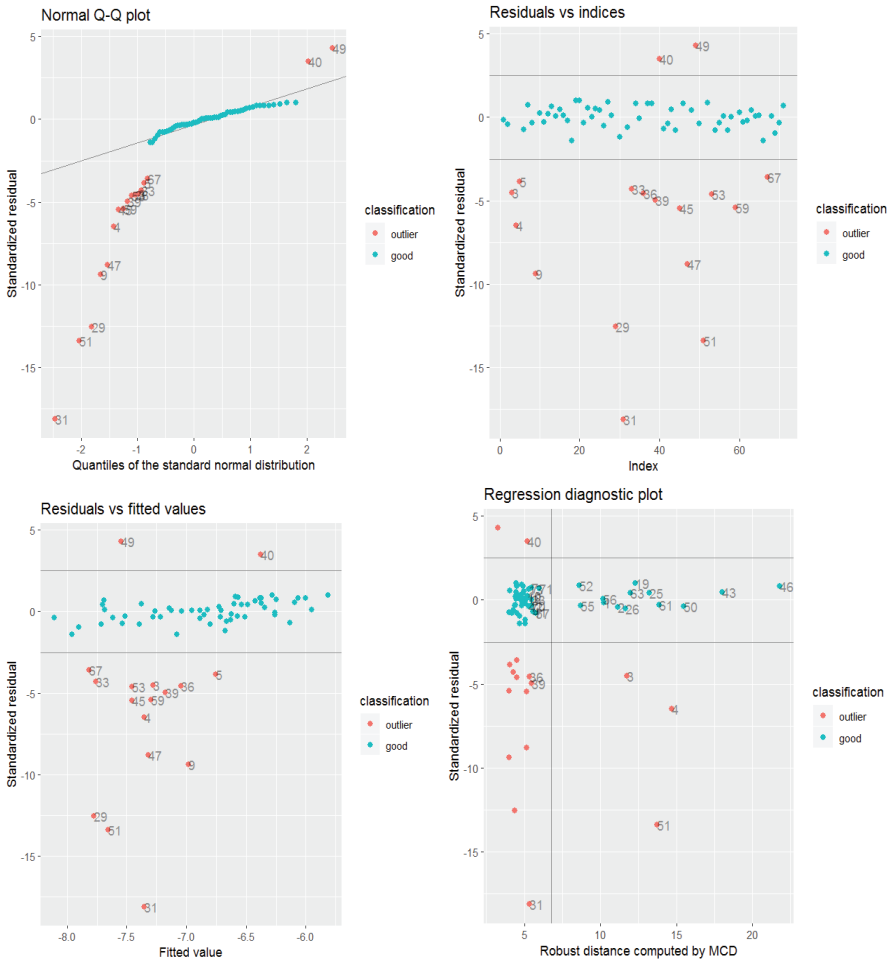


Figure 3: Outlier detection plots for the riboflavin data set.

the coefficients estimation diagrams for different values of the penalty parameter are depicted. We plotted Figure 2 to find the best value of the LASSO parameter (λ_n), which minimizes the CV criterion. As seen in Figure 2, the minimal mean squared error, estimated by CV, is achieved at $\lambda_n = 0.01$. The LASSO method selects 53 variables. Figure 3 produces the diagnostic plots for a sequence of regression models, such as submodels along a robust sparse least trimmed squares regression models for a grid of values for the penalty parameter. In the normal Q-Q plot of the standardized residuals, a reference line is drawn through the first and third quartiles. The index number of ob-

servations with the largest distances from that line are identified by a label (the observation number). In the plots of the standardized residuals versus their index or the fitted values, horizontal reference lines are drawn at 0 and ± 2.5. The index number of observations with the largest absolute values of the standardized residuals is identified by a label (the observation number). For the regression diagnostic plot, the robust Mahalanobis distances of the predictor variables are computed via the minimum covariance determinant (MCD) based on only those predictors with nonzero coefficients. Horizontal reference lines are drawn at ± 2.5 and a vertical reference line is drawn at the upper 97.5% quantile of the chi-squared distribution with p degrees of freedom, where p denotes the number of predictors with nonzero coefficients. The index number of observations with the largest absolute values of the standardized residuals and/or largest robust Mahalanobis distances are identified by a label (the observation number). According to Figure 3, it is clear that there exist some outliers in the data and so, it is necessary to use the robust methods same as the sparse LTS and ALTSCM methods for modeling the data.

We reported the results for the three methods listed in Table 1, in which we numerically calculated $\text{RSS} = (\boldsymbol{y} - \hat{\boldsymbol{y}})^\top (\boldsymbol{y} - \hat{\boldsymbol{y}})$ and $\text{R}^2 = 1 - \text{SSE}/\text{S}_{\text{yy}}$ with $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ and $\text{S}_{\text{yy}} = \sum_{i=1}^{n} z_i (y_i - \bar{y})^2$. They are measures for the error and goodness of prediction, respectively. It is clear that the penalized mixed-integer method performs better than the other methods according to the goodness of fit criteria.

5.2 A simulated data analysis

To examine the performance of the proposed estimators, we perform a simulation data study. For this purpose, the following model is considered with $n = 55$, $p = 450$, and $h = 41$ (see [3] for more details):

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where

$$\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)^\top,\ \boldsymbol{\beta}_1 = (-1.5, 2, 2.5, 4, -3, 5)^\top,\ \boldsymbol{\beta}_{2\ (444\times 1)} \overset{i.i.d.}{\sim} \mathcal{N}(0,1),$$

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{450} \sim \mathcal{N}_{450}(\mathbf{1}_{450}, \boldsymbol{I}_{450}),$$

$$\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2)^\top,\ \boldsymbol{\epsilon}_{1\ (h\times 1)} \sim \mathcal{N}(0,1),\ \boldsymbol{\epsilon}_{2\ ((n-h)\times 1)} \overset{i.i.d.}{\sim} t_2(8),$$

where $t_m(\delta)$ is the noncentral t-student distribution with m degrees of freedom and noncentrality parameter δ.

We produced the first h and the last $n-h$ error terms as random variables from dependent normal and independent noncentral t-student distributions,

Method	LASSO		Sparse LTS		ALTSCM	
	Effective genes	Estimation	Effective genes	Estimation	Effective genes	Estimation
	Intercept	0.6712	*Intercept*	0.3093	*Intercept*	–
	ARGF_at	-0.1911	*CHED_at*	-0.0895	*ABH_at*	0.0124
	DNAJ_at	-0.1364	*CSBA_at*	0.0546	*ALD_at*	0.0124
	GAPB_at	0.0380	*CTAA_at*	0.0336	*AMYC_at*	0.0205
	LYSC_at	-0.2977	*DEAD_at*	-0.1260	*ARGB_at*	-0.0132
	PKSA_at	0.0236	*MREBH_at*	-0.0819	*ARGF_at*	-0.0177
	PRIA_at	0.0982	*SPOIISB_at*	0.0248	*ARGG_at*	-0.0133
	SPOIIAA_at	0.0224	*THIK_at*	-0.2799	*ARGH_at*	-0.0160
	SPOVAA_at	0.2652	*XKDI_at*	0.0001	*BIOB_at*	-0.0133
	THIK_at	-0.0051	*XKDS_at*	0.0988	*CARA_at*	-0.0140
	XHLB_at	0.1608	*YCDH_at*	-0.0018	*CARB_at*	-0.0145
	YACN_at	-0.0404	*YCGM_at*	-0.0257	*GAPB_at*	0.0130
	YBFI_at	0.1329	*YCGP_at*	-0.0168	*GSIB_at*	-0.0129
	YCDH_at	-0.0067	*YCKJ_at*	-0.0761	*NDK_at*	-0.0129
	YCGO_at	-0.0057	*YDAR_at*	-0.0175	*PAND_at*	-0.0128
	YCKE_at	0.0092	*YDBM_at*	-0.3054	*PBUX_at*	0.0140
	YCLB_at	0.1994	*YFLL_at*	-0.0542	*PHRI_r_at*	-0.0128
	YCLF_at	-0.0533	*YHCB_at*	-0.1021	*PTSG_at*	-0.0120
	YDDH_at	-0.0176	*YHCL_at*	-0.0850	*SIGY_at*	-0.0155
	YDDK_at	-0.1142	*YHDO_at*	-0.0762	*SSPA_at*	-0.0122
	YEBC_at	-0.5347	*YJBJ_at*	-0.1005	*YBGB_at*	-0.0139
	YFHE_r_at	0.1451	*YJCJ_at*	0.2446	*YCDH_at*	-0.0307
	YFII_at	0.0100	*YKUH_at*	0.2855	*YCGM_at*	-0.0183
	YFIO_at	0.1588	*YORB_i_at*	0.0519	*YCGN_at*	-0.0206
	YFIR_at	0.0441	*YQET_at*	-0.0217	*YCGO_at*	-0.0207
	YHDS_r_at	0.1452	*YRZI_r_at*	0.0109	*YCIC_at*	-0.0229
	YKBA_at	0.1108	*YTGB_at*	-0.0203	*YDDH_at*	-0.0131
	YKVJ_at	0.0237	*YUIA_at*	0.0113	*YFMH_r_at*	-0.0164
	YLXW_at	0.0731	*YVBY_at*	0.0420	*YHDS_r_at*	0.0166
	YMFE_at	0.0183	*YWQD_at*	-0.0098	*YHFH_r_at*	0.0178
	YOAB_at	-0.8123	*YXEL_at*	-0.0442	*YHZA_at*	-0.0396
	YPGA_at	-0.0102	*YXLD_at*	-0.0389	*YOEB$_a$t*	-0.0122
	YQJT_at	0.0415	*YXLE_at*	-0.1714	*YOPS$_a$t*	0.0149
	YQJU_at	0.2320			*YOQX_i_at*	0.0155
	YRVJ_at	-0.0547			*YPUD_at*	0.0349
	YTGB_at	-0.0390			*YPUF_at*	0.0248
	YUID_at	0.0134			*YPUG_at*	0.0208
	YURQ_at	0.0245			*YRPE_at*	-0.0150
	YXLD_at	-0.2005			*YRZI_r_at*	0.0173
	YXLE_at	-0.1068			*YTGA_at*	-0.0168
	YYBG_at	-0.0781			*YTGB_at*	-0.0191
	YYDA_at	-0.1042			*YTGC_at*	-0.0129
					YTGD_at	-0.0198
					YTIA_at	-0.0290
					YUSA_at	-0.0126
					YXLC_at	-0.0204
					YXLD_at	-0.0270
					YXLE_at	-0.0270
					YXLF_at	-0.0194
					YXLG_at	-0.0216
RSS	594.0313		25.6582		**0.1487**	
R^2	0.6519		0.9093		**0.9458**	

Table 1: Results of the proposed estimators for the riboflavin data set

respectively. Such data generating method makes the outliers to appear on one side of the true regression hyperplane, corrupting the nonrobust estimators by tending them to the outliers.

In Table 2, we have reported the results. To save space, the estimations of nonzero coefficients are just reported in Table 2. Also, the outliers are shown in Figure 4. As seen, ALTSCM is again preferable to the other methods.

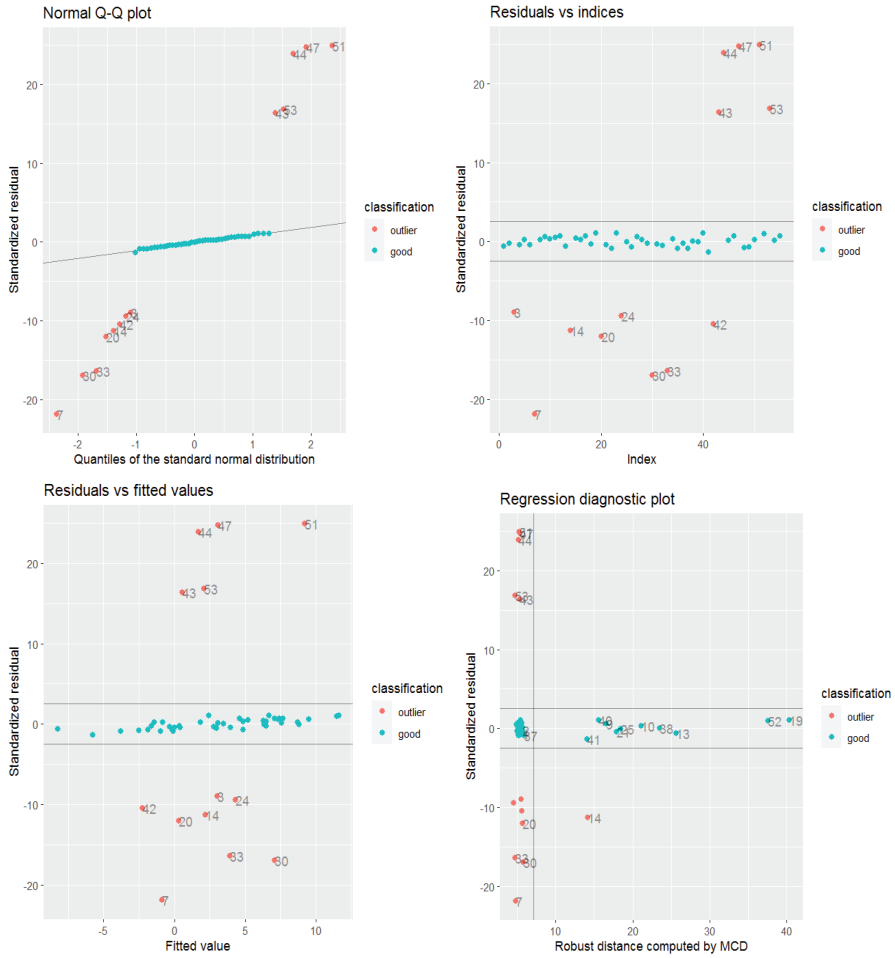


Figure 4: Outlier detection plots for the simulated data set.

Method Parameters	LASSO	Sparse LTS	ALTSCM
β_1	0.00000	0.00000	-1.44790
β_2	0.00000	0.00000	1.98652
β_3	2.08386	2.69022	2.48500
β_4	3.83184	3.47647	4.01102
β_5	-3.01472	-2.03737	-2.99656
β_6	2.27159	2.87485	4.94529
RSS	4203.47	2850.05	**4.71450**
R^2	0.14287	0.41885	**0.93325**

Table 2: Results of the proposed estimators for the simulated data set

6 Conclusions

We dealt with a computationally low cost nonlinear mixed-integer optimization model to combat both the outliers and multicollinearity effects in the high dimensional regression. Suggesting an approximation for the condition number, our model also has a simple (LTS-based) structure. We used the simulated annealing algorithm to solve the model effectively. Computational tests showed that the given estimator (ALTSCM) is practically promising.

References

1. Alfons, A., Croux, C. and Gelper, S., *Sparse least trimmed squares regression for analyzing high dimensional large data set,* Ann. Appl. Stat. 7 (2013), 226–248.

2. Amini M. and Roozbeh, M., *Optimal partial ridge estimation in restricted semiparametric regression models,* J. Multivar. Anal. 136 (2015), 26–40.

3. Arashi M., Roozbeh, M., Hamzah, N.A. and Gasparini, M., *Ridge regression and its applications in genetic studies,* PLoS ONE 16(4) (2021), e0245376.

4. Bertsimas, D. and Tsitsiklis, J.N., *Introduction to linear optimization,* Athena Scientific, Massachusetts, 1997.

5. Buhlmann, P., Kalisch, M. and Meier, L., *High-dimensional statistics with a view towards applications in biology,* Annu. Rev. Stat. Appl. 1 (2014), 255–278.

6. Efron, B. and Hastie, T., *Computer age statistical inference,* Cambridge University Press, Cambridge, 2017.

7. Roozbeh, M., Babaie-Kafaki, S. and Arashi, M., *A class of biased estimators based on QR decomposition*, Linear Algebra Appl. 508 (2016), 190–205.

8. Roozbeh, M., Babaie-Kafaki, S. and Naeimi Sadigh, A., *A heuristic approach to combat multicollinearity in least trimmed squares regression analysis,* Appl. Math. Model. 57 (2018), 105–120.

9. Rousseeuw, P.J. and Leroy, A.M., *Robust regression and outlier detection,* John Wiley and Sons, New York, 1987.

10. Sheather, S.J., *A modern approach to regression with R,* Springer, New York, 2009.

11. Tibshirani, R., *Regression shrinkage and selection via the LASSO,* J. R. Stat. Soc. Ser. B, 58 (1996), 267–288.

12. Watkins, D.S., *Fundamentals of matrix computations,* John Wiley and Sons, New York, 2002.

How to cite this article
Roozbeh, M., Babaie-Kafaki, S., Maanavi, M. A heuristic algorithm to combat outliers and multicollinearity in regression model analysis. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 173-186. doi: 10.22067/ijnao.2021.68160.1008

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 187-200
DOI:10.22067/ijnao.2021.71321.1048
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Heuristic solutions for interval-valued games

R.K. Gupta* and D. Khan

Abstract

When we design the payoff matrix of a game on the basis of the available information, then rarely the information is free from impreciseness, and as a result, the payoffs of the payoff matrix have a certain amount of ambiguity associated with them. In this work, we have developed a heuristic technique to solve two persons $m \times n$ zero-sum games ($m > 2, n > 2$), with interval-valued payoffs and interval-valued objectives. Thus the game has been formulated by representing the impreciseness of the payoffs with interval numbers. To solve the game, a real coded genetic algorithm with interval fitness function, tournament selection, uniform crossover, and uniform mutation has been developed. Finally, our proposed technique has been demonstrated with a few examples and sensitivity analyses with respect to the genetic algorithm parameters have been done graphically to study the stability of our algorithm.

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

Keywords: Two persons zero sum game; Interval-valued payoffs; Genetic Algorithm; Order relations.

1 Introduction

In this paper, an effort has been made to solve two persons $m \times n$ zero-sum games ($m < 2, n < 2$), with interval-valued payoffs and interval-valued ob-

*Corresponding author
Received 5 July 2021; revised 3 October 2021; accepted 6 October 2021
Ranjan Kumar Gupta
Assistant Professor in Management, Department of Commerce and Management, West Bengal State University, Barasat, W.B., India. e-mail: rangupt@rediffmail.com

Debdip Khan
Faculty, Department of Business Administration, Burdwan Raj College, Burdwan, W.B., India. e-mail: debdip.khan@gmail.com

jectives. A payoff matrix with interval-valued payoffs has been considered. Normally, when we design the payoff matrix of a game on the basis of the available information, then rarely the information is free from impreciseness/vagueness, and as a result, the payoffs of the payoff matrix have a certain amount of ambiguity associated with them [8]. Due to this reason, the payoff received by a player as the result of interaction between any of his strategies with any strategy of his competitor is assumed to be interval-valued. Out of the several types of researches done on games, a large fraction has been based on two-person games like Blackwell's work considering vector payoffs [4], Zeleny's investigation on games with multiple payoffs, and so on [32]. Nishizaki and Sakawa [25] worked on two persons zero sum games with multiple goals. Since the end of the last century, especially in the last two decades, researchers have given more emphasis on multiobjective games, considering impreciseness in payoffs and goals. In the majority of these researches, the impreciseness has been represented and dealt with the help of the fuzzy approach. Researchers like Aubin [1, 2], Butnariu [5, 6], Campos [7], Nishizaki and Sakawa [26], Bector and Chandra [3], Vijay et al. [31], Cunlin and Qiang [11], Dutta and Gupta [12], Gong and Hai [15], Chandra and Aggarwal [10], Li [18, 19], Roy and Mondal [28], Jiang et al. [17], Qiu et al. [27], Madandar et al. [21] have all significantly contributed to the researches on the fuzzy games.

However, in the case of fuzzy approach, the user arbitrarily defines the shapes of fuzzy numbers. It is assumed that the constraints, objectives, and parameters are fuzzy sets and that the membership functions are known to the user/decision-maker. However, sometimes the user or the decision-maker is unable to specify the membership function accurately and hence, has to resort to an arbitrary approach. On the other hand, while dealing with impreciseness using interval numbers, one can be absolutely sure that the interval results will always contain the exact result irrespective of whether its upper boundary and lower boundary are overestimated. No scope for subjectivity is present in this case. Thus in a way, the interval approach has some distinct advantages over the fuzzy approach. Because of this, in this paper, we have dealt with the impreciseness with the help of the interval approach; that is, the imprecise payoffs have been represented by interval numbers. The objective function of the game is also interval-valued. In this paper, a genetic algorithm (GA) based heuristic technique of solving two persons zero sum games with interval-valued payoffs and objectives has been proposed, that is, to solve the game a real coded GA with interval fitness function, tournament selection, uniform crossover, and uniform mutation has been developed. GA is a widely popular heuristic search and optimization technique, first developed by Prof. J. H. Holland, University of Michigan. Currently, there are several textbooks on GAs by authors like Goldberg [14], Michalewicz [24], Sakawa [29], Gen and Cheng [13] and others. While implementing GA to solve the game, for selection operation and also for finding the best chromosome in each generation, we have used the concepts of interval arithmetic

and order relation of interval number. The contributions of researchers like Ishibuchi and Tanaka [16], Chanas and Kuchta [9], Sengupta and Pal [30], Mahato and Bhunia [22] in defining ordered relations of interval-valued numbers are worth mentioning.

2 Concepts of interval numbers and their comparisons

Let $K = [k_L, k_R] = \{x : k_L \leq x \leq k_R, k_L, k_R, x \in R\}$ be an interval-valued number, where k_L and k_R are the left and right boundaries of K, respectively. Also, assume that $K = \langle kC, kr\rangle = \{x : k_C k_r \leq x \leq k_C + k_r, , x \in R\}$, where $k_C = (k_L + k_R)/2$ and $k_r = (k_R - k_L)/2$, are perspectively the center and radius of the interval, and R is the set of real numbers. If there are two closed interval numbers K and $G(= [g_L, g_R])$ and $\Delta \in (+, -, 0, 1)$ is a binary operation on the set of real numbers, then the binary operation on interval K and G is defined by $K\Delta G = \{k\Delta g : k \in K \; and \; g \in G\}$. In the case of division, it is assumed that $0 \notin G$. Thus,

(i) $K + G = [k_L + g_L, k_R + g_R]$,

(ii) $K - G = [k_L - g_R, k_R - g_L]$,

(iii) $\lambda K = \begin{cases} [\lambda k_L, \lambda k_R] \text{ if } \lambda \geq 0, \\ [\lambda k_R, \lambda k_L] \text{ if } \lambda < 0, \end{cases}$ where λ is a real number.

(iv) $K \times G = [k_L, k_R] \times [g_L, g_R]$
$= \begin{cases} [Min(k_L g_L, k_L g_R, k_R g_L, k_R g_R), Max(k_L g_L, k_L g_R, k_R, g_L, k_R g_R)], \\ [k_L g_L, k_R g_R], \text{only if } k_L \geq 0 \text{ and } g_L \geq 0. \end{cases}$

Considering the optimistic decision making for maximization problems, the order relation $\geq_{o\max}$ between the intervals K and G is defined as

(v) $K \geq_{o\max} G$ if and only if $k_L \geq g_L$,

(vi) $K > o\max G$ if and only if $K \geq_{o\max} G$ and $K \neq G$.
Considering the pessimistic decision-making for minimization problems, the order relation $>_{p\max}$ between the intervals $K = [k_L, k_R] = \langle k_C, k_r\rangle$ and $G = [g_L, g_R] = \langle g_c, g_r\rangle$ may be defined as

(vii) $K >_{p\max} G$ if and only if $k_C > g_C$, when intervals K and G are either disjoint or partially overlapping.

(viii) $K >_{p\max} G$ if and only if $k_C \geq g_C$ and $k_r < g_r$ when interval K is contained in interval G.

3 Methodology and formulation

A popular and simple case of game theory is two persons zero sum game in which it is assumed that two players are involved, each having a finite number of strategies, and the algebraic sum of the gains and losses of those two players is equal to zero, that is, due to the interaction of any pair of strategies of the competitive players (say A and B), the amount received by one player is exactly equal to the losses of the other player. This can be illustrated with the following the $m \times n$ matrix:

$$M = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

where $M \in \mathbb{R}^{m \times n}$ is an $m \times n$ real payoff matrix of player A. Here $\mathbb{R}^U$ is the U-dimensional Euclidean space and a_{ij} is the payoff of player A, when player A plays the strategy i and player B plays the strategy j.
A mix strategy of player A is given by the vector x in R_+^m, that is, nonnegative orthant of R^m, such that

$$\underset{\sim}{x}^t e_m = 1 \ where \ e_m = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{m \times 1} \quad \text{and} \quad \underset{\sim}{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}_{m \times 1} \geq \underset{\sim}{0}.$$

Thus if S^m is the strategy space of player A, then $S^m = \{\underset{\sim}{x} \in R_+^m, \underset{\sim}{x}^t \ e_m = 1\}$. Similarly, the strategy space of player B is given by $S^n = \{\underset{\sim}{y} \in R_+^n, \underset{\sim}{y}^t \ e_n = 1\}$.
Here, the vector $\underset{\sim}{y}$ denotes a mix strategy of player B such that $\underset{\sim}{y}^t \ e_n = 1$.
If player A plays the mix strategy $\underset{\sim}{x}$ and B plays the mix strategy $\underset{\sim}{y}$, then the expected payoff of player A is given by $x^t M y = \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_i y_j$.
The player A's mix strategy $\underset{\sim}{x}^{\#}$ is said to be his optimal strategy if $\underset{\sim}{x}^t \ M \ \underset{\sim}{y}^{\#} \leq \underset{\sim}{x}^{\#} \ M \ \underset{\sim}{y}^{\#}$ for all $\underset{\sim}{x} \in S^m$, where $\underset{\sim}{y}^{\#}$ is B optimal strategy.
Similarly, the player B's strategy $\underset{\sim}{y}^{\#}$ is said to be optimal strategy if

$$\underset{\sim}{x}^{\#^t} \ M \ \underset{\sim}{y} \geq \underset{\sim}{x}^{\#^t} \ M \ \underset{\sim}{y}^{\#} \qquad \text{for all } \underset{\sim}{y} \in S^n.$$

Thus player A's objective is to determine the optimum values of x_i's in such a manner that it can maxi-min its expected payoff for any values of the elements of the vector y that player B chooses.
Hence from player A's point of view, the game can be expressed by the

following problem:

$$Max\ Z(\underset{\sim}{x}) = \underset{j=1}{\overset{n}{Min}} \sum_{i=1}^{m} a_{ij}x_i \tag{1}$$

subject to $\underset{\sim}{x}^t\ e_m = 1$

$$\underset{\sim}{x}^t \geq \underset{\sim}{0}.$$

From B's side, we have

$$Min\ U(\underset{\sim}{y}) = \underset{i=1}{\overset{m}{Max}} \sum_{j=1}^{n} a_{ij}y_j \tag{2}$$

subject to $\underset{\sim}{y}^t e_n = 1$.

However, as discussed in the introduction section in real-life situation, the payoffs are generally imprecise, and here we have represented them by interval-valued numbers, that is, $a_{ij} = [a_{ij_L}, a_{ij_R}]$.

Thus

$$\sum_{i=1}^{m} a_{ij}x_i = \sum_{i=1}^{m} [a_{ij_L}, a_{ij_R}]x_i = \left[\sum_{i=1}^{m} a_{ij_L}x_i, \sum_{i=1}^{m} a_{ij_R}x_i\right].$$

Hence (1) becomes

$$Max\ Z(\underset{\sim}{x}) = \underset{j=1}{\overset{n}{Min}} \left[\sum_{i=1}^{m} a_{ij_L}x_i, \sum_{i=1}^{m} a_{ij_R}x_i\right],$$

subject to $\underset{\sim}{x}^t e_m = 1$

$$\underset{\sim}{x}^t \geq \underset{\sim}{0}.$$

From B's point of view, the problem is

$$Min\ U(\underset{\sim}{y}) = \underset{i=1}{\overset{m}{Max}} \left[\sum_{j=1}^{m} a_{ij_L}y_j, \sum_{j=1}^{m} a_{ij_R}x_j\right]$$

subject to $\underset{\sim}{y}^t e_n = 1$

$$\underset{\sim}{y}^t \geq \underset{\sim}{0}.$$

4 Solution procedure

We have developed a real-coded GA for solving the interval-valued game. The main algorithm and the broad working principle of GA are widely pop-

ular and are available in several books and journal papers. Hence, we do not present it here anymore. However, the finer details of its basic components like "representation of chromosome", "initialization of population", "evaluation function", "Selection process", and "Genetic operators (crossover and mutation)" are explained below in brief. Furthermore, here we explain the GA developed for solving the problem from player A's side. For solving the problem from player B's side, an almost similar approach is adopted.

In the problem formulated by us, from player A's side, there are "m" (≥ 2) continuous decision variables (each representing the probabilities with which each of the "m" strategies is played by player A). Hence the ith chromosome is represented by a real row matrix $X_i(A) = [X_{i1}, X_{i2}, \ldots, X_{im},]$, where $X_{i1}, X_{i2}, \ldots, X_{im}$ (such that $\sum_{j=1}^{m} X_{ij} = 1$ & $0 \leq X_{ij} \leq 1$ for all $j = 1, 2, \ldots m,$) represent the decision variables, $x_1, x_2, \ldots, x_m$, respectively, of the problem.

After representing the chromosomes, the population size (popsize) numbers of chromosomes have been initialized.

In this problem, for each chromosome, the first component (gene), that is, X_{i1}, is initialized by randomly generating a real number between 0 and 1. Then, X_{i2} is initialized by randomly generating a real number between 0 and $(1 - X_{i1})$. Similarly, X_{i3} is initialized by randomly generating a real number between 0 and $1 - (X_{i1} + X_{i2})$, and so on, until the value of the last component is taken as $X_{im} = \{1 - \sum_{j=1}^{m-1} X_{ij}\}$. For each component of each chromosome, random numbers have been selected by using uniform distribution.

In the evaluation function, the fitness value for each chromosome (i.e., a potential solution) is calculated. In our work, the fitness value for each chromosome is considered to be the value of the objective function corresponding to it. Using the selection operation, the below-average solutions are eliminated from the population for the next generation. In this work, the tournament selection scheme of size two with replacement has been implemented. In this selection scheme, at first, two chromosomes are randomly selected. Then out of these two chromosomes, the better chromosome (i.e., one having the better fitness value) is finally selected for the next generation. The selection of a better chromosome is made on the basis of definitions (*vii*) and (*viii*) (discussed under section 2) of order relations between two interval numbers as the objective function of optimization problem be interval valued. Once the selection process is complete, the surviving chromosomes become eligible to take part in crossover operation, in which at the time two parent chromosomes get involved to generate offspring. These offspring possess the features of both the parent chromosomes. Here we have denoted the probability of crossover by pobcross. In this work, the crossover operation (as shown in Figures 1 and 2) is done in the following manner:

At first, *pobcross* popsizeis* is found and its integral value is stored in variable V.

(i) Then two chromosomes $X_i(A)$ and $X_q(A)$ are randomly selected from the population for crossover. For creating the rth $(r = 1, 2, \ldots, m)$ component $X_{ir}^{/}$ and $X_{qr}^{/}$ of the two offspring, the following procedure is adopted.

(ii) A real number "h_r" is randomly generated between 0 and $|X_{ir} - X_{qr}|$.

(iii) If $X_{ir} > X_{qr}$, then $X_{ir}^{/} = X_{ir} - h_r$ and $X_{qr}^{/} = X_{qr} + h_r$.

Otherwise, if $X_{ir} < X_{qr}$, then $X_{ir}^{/} = X_{ir} + h_r$ and $X_{qr}^{/} = X_{qr} - h_r$.
The above steps [(i) –(iii)] are repeated for $V/2$ times. Let the parent chromosomes be $X_i(A)$ and $X_q(A)$, as shown below in Figure 1: Now if it is assumed

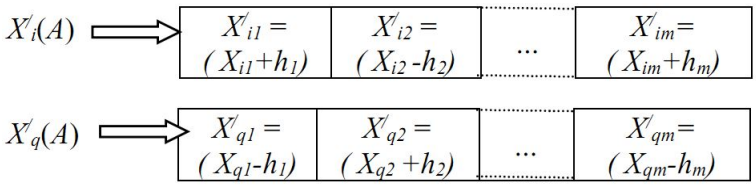


Figure 1: The parent chromosomes

that $X_{i1} < X_{q1}$, $X_{i2} > X_{q2}$, and $X_{im} < X_{qm}$, then the child chromosomes $X_1^{/}(A)$ and $X_q^{/}(A)$ will be given by (as shown below in Figure 2): By apply-

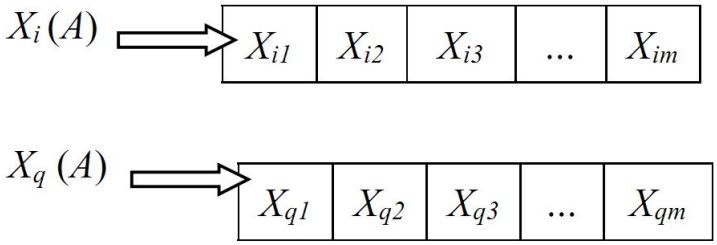


Figure 2: The child chromosomes

ing the mutation operation to a single chromosome, random variations have been injected into the population. The objective of mutation is to push the population slightly towards a better path. Here uniform mutation has been used and the probability of mutation has been denoted by pobmute.
If the elements (genes) X_{iw} and X_{it} of chromosome X_i are selected for mutation, then $|X_{iw} - X_{it}| = b(say)$ is first calculated. Then the modified value of X_{iw} and X_{it} are given by

$$X'_{iw} = \begin{cases} X_{iw} + \Delta(b), & \text{if } X_{it} > X_{iw}, \\ X_{iw} - \Delta(b), & \text{if } X_{it} < X_{iw}, \end{cases} \text{and } X'_{it} = \begin{cases} X_{it} + \Delta(b), & \text{if } X_{iw} > X_{it}, \\ X_{it} - \Delta(b), & \text{if } X_{iw} < X_{it}, \end{cases}$$

where $w \in \{1, 2, \ldots, m\}$, $t \in \{1, 2, \ldots, m\}$, and $\Delta(b) = a$ is a random real number between $[0, b]$.

5 Numerical example

We have considered three examples and solved them with the proposed GA. Except for Example 3, in the other two examples, the payoffs are considered as interval-valued. However, in Example 3, they are considered as fixed (by taking identical values for the left boundaries and right boundaries of the intervals).The values of the parameters considered in these examples are all feasible, although they have not been selected from any case study. In each of the examples, 20 independent runs have been performed by the proposed GA, of which the best value of the game has been taken and the corresponding mix strategies of both the players are determined and displayed in Tables 1–6. The following values of GA parameters are used in this work: $popsize = 100$, $pobcross = 0.95$, $pobmute = 0.15$, $maxgen = 1000$.

Example 1.

$$M_1 = \begin{bmatrix} [2,7,3,3] & [0,8] & [-3,-1] \\ [-5,-1] & [0,0] & [0,2] \\ [-1.5,-0.5] & [-6,-2] & [0.25,3.75] \end{bmatrix}$$

Mixed Strategy of player A i.e. $\underset{\sim}{x}$	Value of the game in interval form [Obj L, Obj R]	Expected value of the game, i.e., center value of [Obj L, Obj R]
$\underset{\sim}{x} = \begin{bmatrix} 0.403846 \\ 0.230769 \\ 0.365385 \end{bmatrix}$	$[-1.120192, 1.427885]$ $[-2.192308, 2.500000]$ $[-0.611539, 0.919231]$	0.153846

Table 1: Solution for player A

Mixed strategy of player B, i.e., $\underset{\sim}{y}$	Value of the game in interval form [Obj L, Obj R]	Expected value of the game, i.e., center value of [Obj L, Obj R]
$\underset{\sim}{y} = \begin{bmatrix} 0.153846 \\ 0.230769 \\ 0.615385 \end{bmatrix}$	$[-1.1738462, 1.430769]$ $[-1.769231, 1.461538]$ $[-0.076923, 0.769231]$	-0.153846

Table 2: Solution for player B

Example 2.

$$M_2 = \begin{bmatrix} [1,5] & [-2,6] & [7,9] \\ [3,7] & [7,7] & [3,5] \\ [4,8] & [0,1] & [3,3] \end{bmatrix}$$

Mixed strategy of player A, i.e., $\underset{\sim}{x}$	Value of the game in interval form [Obj L, Obj R]	Expected value of the game, i.e., center value of [Obj L, Obj R]
$\underset{\sim}{x} = \begin{bmatrix} 0.225490 \\ 0.598039 \\ 0.176471 \end{bmatrix}$	$[2.725490, 6.725490]$ $[3.901961, 5.549020]$	4.725490

Table 3: Solution for player A

Mixed strategy of player B, i.e., $\underset{\sim}{y}$	Value of the game in interval form [Obj L, Obj R]	Expected value of the game, i.e., center value of [Obj L, Obj R]
$\underset{\sim}{y} = \begin{bmatrix} 0.607843 \\ 0.039216 \\ 0.352941 \end{bmatrix}$	$[-6.450980, -3.000000]$ $[-6.294117, -3.156863]$	-4.725490

Table 4: Solution for player B

Example 3.

$$M_3 = \begin{bmatrix} [5,5] & [7,7] & [4,4] & [10,10] \\ [4,4] & [3,3] & [7,7] & [2,2] \\ [7,7] & [2,2] & [5,5] & [6,6] \end{bmatrix}$$

Mixed strategy of player A, i.e., $\underset{\sim}{x}$	Value of the game in interval from [Obj L, Obj R]	Expected value of the game, i.e., center value of [Obj L, Obj R]
$\underset{\sim}{x} = \begin{bmatrix} 0.548387 \\ 0.290323 \\ 0.161290 \end{bmatrix}$	$[5.032258, 5.032258]$	5.032258

Table 5: Solution for player A

Mixed strategy of player B, i.e., $\underset{\sim}{y}$	Value of the game in interval from [Obj L, Obj R]	Expected value of the game, i.e., center value of [Obj L, Obj R]
$\underset{\sim}{y} = \begin{bmatrix} 0.354839 \\ 0.225806 \\ 0.413955 \\ 0.000000 \end{bmatrix}$	$[-5.032258, -5.032258]$	-5.032258

Table 6: Solution for player B

6 Sensitivity analysis

The outcome of this work is heavily dependent on the stability, convergence, and efficiency of the algorithm proposed by us. Hence, to study this stability, by considering Example 1, the sensitivity of the "expected value of the game" (from player A's point of view) have been analyzed graphically with respect to GA parameters like pobcross, pobmute, maxgen and pobsize separately,

keeping the other parameters at their original values. Figures 3–6 reveal that the result of Example 1, obtained by the GA proposed by us, is stable over a large range of the GA parameters mentioned above.

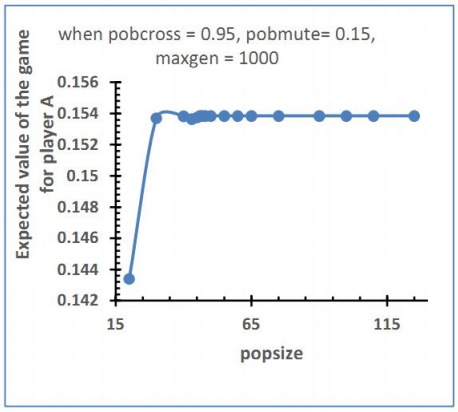


Figure 3: Expected value of the game for player A vs popsize

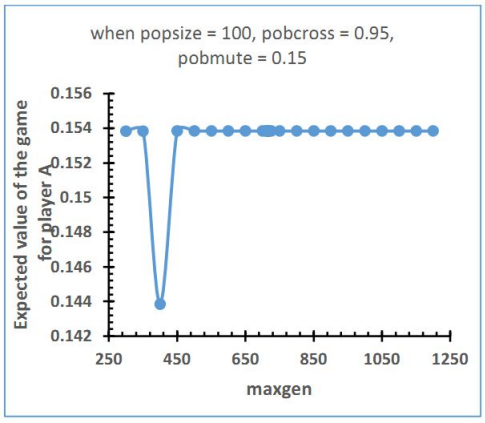


Figure 4: Expected value of the game for player A vs maxgen

Figure 3 shows that when *pobcrossis* fixed at 0.95, *pobmuteat* 0.15 and *maxgenat* 1000, then the expected value of the game is fairly stable when the *popsizeis* equal to or above 43. Similarly, it is evident from Figure 4 that when *pobcross, pobmute* and *pobsize* are kept at their original values, then once the value of *maxgen* goes above 452 mark. Moreover, the expected value of the game becomes perfectly horizontal and thus confirms that it is perfectly stable. Similarly, from Figures 5 and 6, it is clear that the expected value of the game is stable when the value of *pobcrossis* greater or equal to 0.72 and the value of *pobmuteis* greater or equal to 0.1.

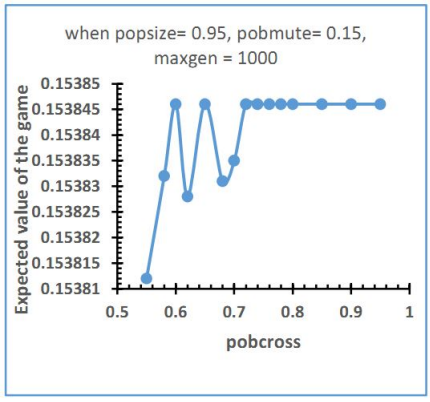


Figure 5: Expected value of the game for player A vs pobcross

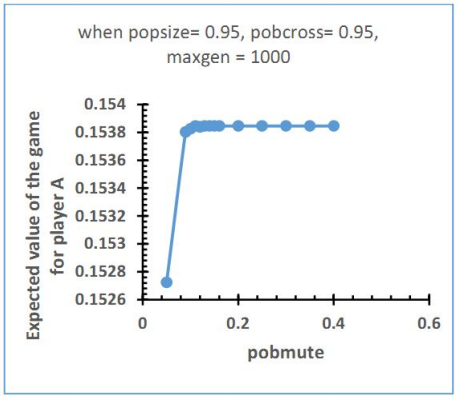


Figure 6: Expected value of the game for player A vs pobmute

7 Conclusion

In this paper, a heuristic technique has been developed to solve two persons $m \times n$ zero sum games ($m > 2, n > 2$), with interval-valued payoffs and interval-valued objectives. It is a well-known fact that during the design phase of the payoff matrix of a game rarely the available information is free from impreciseness. Because of this impreciseness, the payoffs of the payoff matrix have a certain degree of ambiguity present in them. To address this impreciseness-driven ambiguity in payoffs and goals, since the end of the last century, several types of research have been done in two persons zero sum games with the help of fuzzy approach. However, as discussed in the "Introduction" section of this paper, the interval approach has some distinct advantages over the fuzzy approach. While dealing with impreciseness using the interval approach, one can be absolutely sure that the interval result

will always contain the exact result irrespective of whether its upper boundary and lower boundary are overestimated. Hence, in this work, we have addressed the said impreciseness with the help of interval-valued numbers and thus formulated the game with interval-valued payoffs and objectives. To solve the game, a real coded GA with interval fitness function, tournament selection, uniform crossover, and uniform mutation has been developed. Finally, a few numerical examples have been solved with the help of the developed GA and to study the stability of the GA, sensitivity analyses with respect to different GA parameters have been done and shown graphically.

References

1. Aubin, J.P. *Mathematical Methods of Game and Economic Theory*,7. North-Holland Publishing Co., Amsterdam-New York, 1979.

2. Aubin, J. P. *Cooperative fuzzy game*, Math. Oper. Res. 6 (1981), no. 1, 1–13.

3. Bector, C.R. and Chandra, S. *Fuzzy Mathematical Programming and Fuzzy Matrix Game*, Springer-Verlag, Berlin, Heidelberg, 2005.

4. Blackwell, D. *An analog of the minimax theorem for vector payoffs,* Pacific J. Math. 6 (1956), 1–8.

5. Butnariu, D. *Fuzzy games: a description of the concept*, Fuzzy Sets and Systems 1 (1978), no. 3, 181–192.

6. Butnariu, D. *Stability and Shapley value for an n-persons fuzzy game,* Fuzzy Sets and Systems 4 (1980), no. 1, 63–72.

7. Campos, L. *Fuzzy linear programming models to solve fuzzy matrix games*, Fuzzy Sets and Systems 32(3) (1989), 275–289.

8. Cevikel, A.C. and Ahlatçıoğlu, M. *Solutions for fuzzy matrix games*, Comput. Math. Appl. 60(3) (2010), 399–410.

9. Chanas, S. and Kuchta, D. *Multiobjective programming in the optimization of interval objective functions-A generalized approach*, Eur. J. Oper. Res. 94 (1996) 594–598.

10. Chandra, S. and Aggarwal, A. *On solving matrix games with pay-offs of triangular fuzzy numbers: Certain observations and generalizations*, Eur. J. Oper. Res. 246(2015) 575–581.

11. Cunlin, L. and Qiang, Z. *Nash equilibrium strategy for fuzzy non-cooperative games*, Fuzzy Sets and Systems 176 (2011), 46–55.

12. Dutta, B. and Gupta, S.K. *On Nash equilibrium strategy of two-person zero-sum games with trapezoidal fuzzy payoffs*, Fuzzy Inf. Eng. 6 (2014), no. 3, 299–314.

13. Gen, M. and Cheng, R. *Genetic algorithms and engineering optimization*, John Wiley & Sons Inc., 2000.

14. Goldberg, D.E. *Genetic algorithms: Search, optimization and machine learning, reading*, MA: Addison Wesley, 1989.

15. Gong, Z. and Hai, S. *The interval-valued trapezoidal approximation of interval-valued fuzzy numbers and its application in fuzzy risk analysis*, J. Appl. Math. (2014) 1–22.

16. Ishibuchi, H. and Tanaka, H. *Multiobjective programming in optimization of the interval objective function*, Eur. J. Oper. Res. 48 (1990) 219–225.

17. Jiang, W., Xie, C., Luo, Y. and Tang, Y. *Ranking Z-numbers with an improved ranking method for generalized fuzzy numbers*, J. Intell. Fuzzy Syst. 32(2017) 1931–1943.

18. Li, D.F. *An effective methodology for solving matrix games with fuzzy payoffs*, IEEE Transaction 43 (2013) 610–621.

19. Li, D.F. *Linear programming models and methods of matrix games with payoffs of triangular fuzzy numbers. Studies in Fuzziness and Soft Computing*, 328. Springer, Heidelberg, 2016.

20. LotfiKatooli, L. and Shahsavand, A. *A reliable approach for terminating the GA optimization method*, Iranian Journal of Numerical Analysis and Optimization, 7(1), (2017), 83–105.

21. Madandar, F., Haghayeghi, S. and Vaezpour, S.M. *Characterization of Nash equilibrium strategy for heptagonal fuzzy games*, Int. J. Anal. Appl., 16 (3) (2018) 353–367.

22. Mahato, S.K. and Bhunia, A.K. *Interval-arithmetic-oriented interval computing technique for global optimization*, AMRX Appl. Math. Res. Express 2006, Art. ID 69642, 19 pp.

23. Mazraeh, H.D. and Pourgholi, R. *An efficient hybrid algorithm based on genetic algorithm (GA) and Nelder Mead (NM) for solving nonlinear inverse parabolic problem,* Iranian Journal of Numerical Analysis and Optimization 8 (2018), 119–140.

24. Michalewicz, Z. *Genetic algorithms + data structure= evaluation programs*, Berlin: Springer Verlag, 1996.

25. Nishizaki, I. and Sakawa, M. *Two-person zero-sum games with multiple fuzzy goals*, J. Fuzzy Theory Systems 4 (3) (1992), 289–300.

26. Nishizaki, I. and Sakawa, M. *Fuzzy and multiobjective games for conflict resolution*, Studies in Fuzziness and Soft Computing, 64. Physica-Verlag, Heidelberg, 2001.

27. Qiu, D., Xing, Y. and Chen, S. *Solving fuzzy matrix games through a ranking value function method*, Journal of Mathematics and Computer Science (JMCS) 18 (2018) 175–183.

28. Roy, S.K. and Mondal, S.N. *An approach to solve fuzzy interval valued matrix game*, Int. J. Oper. Res. 26(3) (2016), 253–267.

29. Sakawa, M. *Genetic algorithms and fuzzy multiobjective optimization*, Operations Research/Computer Science Interfaces Series, 14. Kluwer Academic Publishers, Boston, MA, 2002.

30. Sengupta, A. and Pal, T.K. *Theory and methodology on comparing interval numbers*, Eur. J. Oper. Res.127 (2000) 28–43.

31. Vijay, V., Chandra, S. and Bector, C.R. *Matrix games with fuzzy goals and fuzzy payoffs*, Omega: The International Journal of Management 33 (2005) 425–429.

32. Zeleny, M. *Games with multiple payoffs*, Internat. J. Game Theory 4 (1975), no. 4, 179–191.

How to cite this article
Gupta, R., Khan, D. Heuristic solutions for interval valued games. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1):187 -200. doi: 10.22067/ijnao.2021.71321.1048

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 201-227
DOI:10.22067/ijnao.2021.68926.1016
https://ijnao.um.ac.ir/

Research Article

Using Mott polynomials operational matrices to optimize multi-dimensional fractional optimal control problems

S.A. Alavi*, A.R. Haghighi, A. Yari and F. Soltanian

Abstract

We offer a method for solving the fractional optimal control problems of multi-dimensional. We obtain a fractional derivative and multiplication operational matrix for Mott polynomials (M-polynomials). In the proposed method, the Caputo sense of the fractional derivative is applied on dynamical system. The main feature of this method is to reduce the problem into a system of algebraic equations in order to simplify it. We also show that by increasing the approximation points, the responses converge to the real answer. When the degree of fractional derivative approaches to 1, then the obtained solution approaches to the classical solution as well.

AMS subject classifications (2020): 49MXX, 34H05.

Keywords: Mott polynomials; Caputo derivative; fractional optimal control problems; operational matrix.

---

*Corresponding author
Received 16 February 2021; revised 24 November 2021; accepted 9 December 2021
Seyyed Ali Alavi
Department of Mathematics, Payame Noor University, Tehran, Iran.
mail: ali336alavi@gmail.com

Ahmad Reza Haghighi
Department of Mathematics, Technical and Vocational University, Tehran, Iran.
mail: ah.haghighi@gmail.com

Ayatollah Yari
Department of Mathematics, Payame Noor University, PO BOX 19395-3697, Tehran, Iran. mail:a_yary@yahoo.com

Fahimeh Soltanian
Department of Mathematics, Payame Noor University, PO BOX 19395-3697, Tehran, Iran. mail:f_soltanian@pnu.ac.ir

1 Introduction

Over the last years, many researchers have developed fractional calculus and its applications in physics, chemistry, engineering, and so on; see [2, 7, 8, 12]. A fractional dynamic system (FDs) is a system whose dynamics are expressed by fractional differential equations, and a fractional optimal control problem (FOCP) is an optimal control problem for an FDs [4, 26]. The theory of optimal control is an area in mathematics that has been developing for years, but the theory of fractional optimal control is a new area. Several authors studied this field [1, 3, 9, 22]. Some applications of FOCPs are given in [10, 19]. In most papers, one-dimensional FOCPs are considered where the problem is only with one state, one control variable, and one fractional differential equation [14, 15, 28]. In recent years, the use of different methods for solving optimal control problems has been considered by some researchers. A new numerical approach for solving FOCPs, including state and control inequality constraints using new biorthogonal multiwavelets, was made by Ashpazzadeh, Lakestani, and Yildirim [6]. In this paper, we consider FOCPs in the Caputo sense with multi-dimensional variables. In this method, the state and the control vectors do not necessarily have only one dimension.

An FOCP can be defined with respect to different definitions of fractional derivatives. Most important types of fractional derivatives are the Riemann–Liouville and the Caputo. In this paper, we consider the multi-dimensional FOCP as follows:
Determinate states $X(t) \in \mathbb{R}^n$ and controls $U(t) \in \mathbb{R}^k$, which

$$\text{Min } J(t, X(t), U(t)) = \int_0^1 f(t, X(t), U(t))dt, \tag{1}$$

subject to

$$D^\alpha x_i(t) = g_i(t, X(t), U(t)), \qquad 0 < \alpha \leq 1, 0 < t \leq 1, i = 1, \ldots, n, \tag{2}$$

and satisfy the initial condition

$$X(0) = X_0, \tag{3}$$

where

$$X(t) = [x_1(t), \ldots, x_n(t)]^T, \qquad U(t) = [u_1(t), \ldots, u_k(t)]^T, \tag{4}$$

$X_0 = [x_{0,1}(t), \ldots, x_{0,n}(t)]^T$, and also $f, g_i : [0, 1] \times \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$ are polynomial functions. The above problem reduces to a standard optimal control problem, when $\alpha = 1$.

The main purpose is to generalize the Mott operational matrix to fractional calculus. In the present work, we use Mott polynomials (MPs) for

solving FOCPs. The method consists of expanding the solution by MPs with the unknown coefficients. The properties of MPs are used to evaluate the unknown coefficients and find an approximate solution for $X(t)$ and $U(t)$ in the problem. Also, illustrative examples are included to demonstrate the applicability of the new approach.

This article is organized as follows. Section 2 presents some of the preliminaries in fractional calculus. Section 3 describes the MPs and function approximation. We make an operational matrix for fractional integration, derivative, and multiplication by MPs in Section 4. In Section 5, we apply the MPs to solve multi-dimensional FOCPs. In Section 6, the convergence of the proposed method is discussed. In Section 7, the numerical examples are simulated to demonstrate the high performance of the proposed method. Finally, Section 8 concludes our work.

2 Some preliminaries in fractional calculus

This section provides some basic definitions of the fractional calculus.

Definition 1. We define from [11] $C_\mu = \{f(t) : f(t) > 0 \text{ for } t > 0, f(t) = t^p f_1(t) \text{ where } p > \mu, f_1 \in C[0,\infty)\}$ and $C_\mu^n = \{f(t) : f^{(n)}(t) \in C_\mu\}$, where $n \in N$ and $\mu \in \mathbb{R}$.

Definition 2. The Riemann–Liouville fractional integral operator of order $\alpha \geq 0$ of a function $f \in C_\mu$, $\mu \geq 1$, is defined as [11]

$$ {}_0I_t^\alpha f(t) = \begin{cases} \dfrac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} f(s) ds, & \alpha > 0, t > 0, \\ f(t), & \alpha = 0, \end{cases} \tag{5} $$

and for $n-1 < \alpha \leq n, n \in N, t > 0, f \in C_{-1}^n$, the fractional derivative of $f(t)$ in the Caputo sense is defined as

$$ \begin{aligned} {}_0^c D_t^\alpha f(t) &= I^{n-\alpha} D^n f(t) \\ &= \begin{cases} \dfrac{1}{\Gamma(n-\alpha)} \int_0^t (t-s)^{n-\alpha-1} \dfrac{d^n}{ds^n} f(s) ds, & n-1 < \alpha < n, n \in N, \\ f^{(n)}(t), & \alpha = n, n \in N, \end{cases} \end{aligned} \tag{6} $$

where D^n is the nth-order derivative.

Definition 3. For $f \in C_\mu, \mu \geq -1, \alpha, \beta \geq 0$, we have [27]

$$ {}_0I_t^\alpha t^\gamma = \frac{\Gamma(\gamma+1)}{\Gamma(\gamma+\alpha+1)} t^{\gamma+\alpha}, \tag{7} $$

and for $n-1 < \alpha \leq n, n \in N, t > 0, f \in C_{\mu}^{n}, \mu \geq -1$, also, we have the following properties:

1. ${}_{0}^{c}D^{\alpha} I^{\alpha} f(t) = f(t)$,

2. $I^{\alpha}\ {}_{0}^{c}D^{\alpha} f(t) = f(t) - \sum_{j=0}^{n-1} f^{(t)}\left(0^{+}\right) \frac{x^{k}}{k!}$,

3. ${}_{0}^{c}D^{\beta} f(t) = I^{\alpha-\beta} D^{\alpha} f(t)$.

3 MPs

The MP $s_n(t)$ is defined by (see [21, 23, 25])

$$s_n(t) = (-1)^n \left(\frac{t}{2}\right)^n (n-1)! \sum_{k=0}^{h\left(\frac{n}{2}\right)} \binom{n}{k} (-1)^k \frac{t^{-2k}}{(n-2k-1)!}, \tag{8}$$

where

$$h\left(\frac{n}{2}\right) = \begin{cases} \dfrac{n}{2} & \text{if } n \text{ is even,} \\ \dfrac{n}{2} - \dfrac{1}{2} & \text{if } n \text{ is odd.} \end{cases}$$

From the above definition, it is obvious that $s_0(t) = 1 \neq 0$. For this reason $s_n(t)$ is a Sheffer set [29]. The first few MPs are

$$\begin{aligned} s_0(t) &= 1, \\ s_1(t) &= -\frac{1}{2}t, \\ s_2(t) &= \frac{1}{4}t^2, \\ s_3(t) &= \frac{3}{4}t - \frac{1}{8}t^3, \\ s_4(t) &= -\frac{3}{2}t^2 + \frac{1}{16}t^4, \\ s_5(t) &= -\frac{15}{2}t + \frac{15}{8}t^3 - \frac{1}{32}t^5. \end{aligned}$$

Now we define the vector A_r, for $r = 0, 1, \ldots, n$. If r is odd, then for $i = 0, 1, \ldots, n$,

$$A_r = b_{i_r}, \tag{9}$$

$$b_{i_r} = \begin{cases} 0, & i = 2l, l \in N \cup \{0\}, \\ c_{i_r} \text{ otherwise}, \end{cases} \tag{10}$$

$$c_{i_r} = \left(\frac{-1}{2}\right)^r (r-1)! \frac{\binom{r}{k_i}(-1)^{k_i}}{(r-2k_i-1)!},$$

$$k_i = \beta_j, \beta_j - 1, \ldots, 0,$$

$$\beta_j = \frac{r-1}{2} - j, \quad j = 0, 1, \ldots, \frac{r-1}{2}. \tag{11}$$

If r is even, then for $i = 0, 1, \ldots, n$,

$$A_r = b'_{i_r}, \tag{12}$$

$$b'_{i_r} = \begin{cases} 0 & i = 2l+1, l \in N \cup \{0\}, \\ c'_{i_r} \text{ otherwise}, \end{cases} \tag{13}$$

$$c'_{i_r} = \left(\frac{-1}{2}\right)^r (r-1)! \frac{\binom{r}{k_i}(-1)^{k'_i}}{(r-2k'_i-1)!},$$

$$k'_i = \beta'_j, \beta'_j - 1, \ldots, 0,$$

$$\beta'_j = \frac{r}{2} - j, \quad j = 0, 1, \ldots, \frac{r}{2}. \tag{14}$$

Hence, $S_r(t) = A_r T_r(t)$ for $(r = 0, 1, \ldots, n)$, here $T_r(t) = [1, t, \ldots, t^r]^T$. Then, we define an $(r+1) \times (r+1)$ lower triangular matrix A such that $A = [A_0, A_1, \ldots, A_r]^T$ and $A_i (i = 0, 1, \ldots, n)$ is a row vector of order r.
As a result,

$$\varphi_n(t) = A T_n(t), \tag{15}$$

where

$$\varphi_n(t) = \left[s_0(t), s_1(t), \ldots, s_n(t)\right]^T. \tag{16}$$

3.1 Function approximation

We recall here a theorem that was stated and proved in [20]. Suppose that $H = L^2[0,1]$ is a Hilbert space and that $\{s_0, s_1, \ldots, s_n\}$ is the MPs of degree n on the interval $[0,1]$. We define $Y = \text{Span}\{s_0, s_1, \ldots, s_n\}$. Let f be an arbitrary element in H. Since Y is a finite-dimensional subspace of the space H, the function f has the best unique approximation on Y like $f_n \in Y$, that is, there exists $f_n \in Y$ such that

$$\|f - f_n\|_2 \leq \|f - y\|_2, \qquad \text{for all } y \in Y, \tag{17}$$

where $\|f\|_2 = \sqrt{\langle f, f\rangle}$ and $\langle \cdot, \cdot \rangle$ denotes the inner product. Since $f_n \in Y$, therefore f_n is a linear combination of the spanning basis of Y; that is, there exist $n+1$ coefficients

$$C = [c_0, c_1, \ldots, c_n] \in R \tag{18}$$

such that

$$f(t) \simeq f_n(t) = \sum_{j=0}^{n} c_j s_j(t) = C^T \varphi_n(t), \tag{19}$$

where

$$\|f - f_n\|_2 \to \min . \tag{20}$$

Then C can be obtained by

$$C = Q^{-1} \langle f(t), \varphi_n(t) \rangle, \tag{21}$$

where

$$Q = \langle \varphi_n(t), \varphi_n(t) \rangle = \int_0^1 \varphi_n(t) \varphi_n(t)^T dt. \tag{22}$$

Theorem 1. [20] Let X be an inner product space and let $M \neq \emptyset$ be a convex subset that is complete in the metric induced by the inner product. Then for every given $x \in X$, there exists unique $y \in M$ such that

$$\delta = \inf_{\tilde{y} \in M} \|x - \tilde{y}\| = \|x - y\|. \tag{23}$$

4 Mott operational matrices

4.1 Mott operational matrix of the fractional integration

In this section, we describe the MPs operational matrix of fractional integration of the vector φ_n. The operational matrix can be approximated as

$$I^{\alpha} \varphi_n(t) \simeq P \varphi_n(t), \tag{24}$$

where P is the $(n+1)\times(n+1)$ Riemann–Liouville fractional operational matrix of integration. We construct P as follows:

$$\begin{aligned} I^{\alpha}s_i(t) &= (-1)^i\left(\frac{1}{2}\right)^i(i-1)!\sum_{k=0}^{h\left(\frac{i}{2}\right)}\binom{i}{k}(-1)^k\frac{I^{\alpha}t^{i-2k}}{(i-2k-1)!} \\ &= (-1)^i\left(\frac{1}{2}\right)^i(i-1)!\sum_{k=0}^{h\left(\frac{i}{2}\right)}\frac{\binom{i}{k}(-1)^k}{(i-2k-1)!}t^{i-2k+\alpha}\frac{\Gamma(i-2k+1)}{\Gamma(i-2k+\alpha+1)}. \end{aligned} \tag{25}$$

Now we approximate $t^{i-2k+\alpha}$ by $n+1$ terms of the Mott basis

$$t^{i-2k+\alpha}\simeq\sum_{j=0}^{n}b_j s_j(t), \tag{26}$$

where

$$\begin{aligned} b_j &= Q_j^{-1}\int_0^1 t^{i-2k+\alpha}s_j(t)dt \\ &= (-1)^j\left(\frac{1}{2}\right)^j(j-1)!\sum_{L=0}^{h\left(\frac{j}{2}\right)}\frac{\binom{j}{L}(-1)^L}{(j-2L-1)!}\times\frac{1}{i-2k+\alpha+j-2L+1}. \end{aligned} \tag{27}$$

Therefore we have

$$I^{\alpha}s_i(t)\simeq\sum_{j=0}^{n}B_{ij}s_j(t), \tag{28}$$

where

$$\begin{aligned} B_{ij} =&(-1)^{i+j}\left(\frac{1}{2}\right)^{i+j}(i-1)!(j-1)! \\ &\sum_{k=0}^{h\left(\frac{i}{2}\right)}\sum_{L=0}^{h\left(\frac{j}{2}\right)}\binom{j}{L}\binom{i}{k}(-1)^{k+L}\frac{1}{(i-2k-1)!(j-2L-1)!} \\ &\times\frac{1}{i+j-2L-2k+\alpha+1}, \end{aligned} \tag{29}$$

and

$$Q_j=\left\langle s_j(t),s_j(t)\right\rangle=\int_0^1 s_j(t)s_j(t)^T dt. \tag{30}$$

Finally, we obtain

$$P = \begin{bmatrix} B_{00} & \dots & B_{0n} \\ \vdots & \ddots & \vdots \\ B_{n0} & \dots & B_{nn} \end{bmatrix}, \tag{31}$$

where P is called the MPs operational matrix of fractional integration.

4.2 Mott operational matrix of the fractional derivative

In this section, we describe the MPs operational matrix of fractional derivative of the vector $\varphi_n(t)$. The operational matrix can be approximated as

$${}_0^c D_t^\alpha \varphi_n(t) \simeq T\varphi_n(t), \tag{32}$$

where T is the $(n+1)\times(n+1)$ operational matrix of the fractional derivative. We construct T as follows:

$$\begin{aligned} {}_0^c D_t^\alpha \varphi_n(t) &= \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-s)^{n-\alpha-1} \frac{d^n}{ds^n} \varphi_n(s) ds \\ &= \frac{1}{\Gamma(n-\alpha)} t^{n-\alpha-1} * \varphi_n^{(n)}(t), \end{aligned}$$

where $*$ denotes the convolution product and

$$\varphi_n^{(n)}(t) = D^n \varphi_n(t).$$

Therefore we have

$$\begin{aligned} D^\alpha \varphi_n(t) &= \frac{1}{\Gamma(n-\alpha)} t^{n-\alpha-1} * D^n \varphi_n(t) \\ &= \frac{D^n}{\Gamma(n-\alpha)} t^{n-\alpha-1} * \varphi_n(t) = \frac{D^n}{\Gamma(n-\alpha)} t^{n-\alpha-1} * AT_n(t) \\ &= \frac{AD^n}{\Gamma(n-\alpha)} t^{n-\alpha-1} * [1, t, \dots, t^n] \\ &= \frac{AD^n}{\Gamma(n-\alpha)} [t^{n-\alpha-1} * 1, t^{n-\alpha-1} * t, \dots, t^{n-\alpha-1} * t^n] \\ &= AD^n [D^\alpha 1, D^\alpha t, \dots, D^\alpha t^n] \\ &= AD^n \left[\frac{0!}{\Gamma(1-\alpha)} t^{-\alpha}, \frac{1!}{\Gamma(2-\alpha)} t^{1-\alpha}, \dots, \frac{n!}{\Gamma(n-\alpha+1)} t^{n-\alpha} \right] \\ &= AKD^n \bar{T}_n, \end{aligned}$$

where K is an $(n+1)\times(n+1)$ matrix where the entries are given by

$$K_{i,j}=\begin{cases}\dfrac{i!}{\Gamma(i+1-\alpha)}, & i=j,\\ 0, & i\neq j,\end{cases}\qquad i,j=0,1,\ldots,n,$$

and

$$\bar{T}_n=[t^{-\alpha},t^{1-\alpha},\ldots,t^{n-\alpha}]^T.$$

Now we approximate $t^{n-\alpha}$ by $n+1$ terms of the Mott basis

$$t^{n-\alpha}\simeq E^T\varphi_n(t),$$

where

$$\begin{aligned}E&=Q^{-1}\int_0^1 t^{n-\alpha}\varphi_n(t)dt=Q^{-1}\int_0^1 t^{n-\alpha}AT_n(t)dt\\&=Q^{-1}A\int_0^1 t^{n-\alpha}[1,t,\ldots,t^n]dt\\&=Q^{-1}A\left[\frac{1}{n-\alpha+1},\frac{1}{n-\alpha+2},\ldots,\frac{1}{n-\alpha+n}\right]^T,\end{aligned}$$

and

$$\begin{aligned}Q&=\langle\varphi_n,\varphi\rangle=\int_0^1\varphi_n(t)\varphi_n(t)^Tdt\\&=\int_0^1 AT_n(t)T_n(t)^TA^Tdt=A\int_0^1 T_n(t)T_ndtA^T=AHA^T,\end{aligned}$$

where H is the well-known Hilbert matrix

$$H=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1}\\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+2}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \frac{1}{n+1} & \frac{1}{n+2} & \frac{1}{n+1} & \cdots & \frac{1}{2n+1}\end{bmatrix}.$$

Finally, we obtain

$${}_0^cD_t^\alpha\varphi_n(t)\simeq T\varphi_n(t),$$

where

$$T=AKD^nE. \tag{33}$$

Moreover, T is called the MPs operational matrix of fractional derivative.

4.3 The operational matrix of multiplication

The following property of the product of two Mott function vectors will be also used:

$$C^T\varphi_n(t)\varphi_n(t)^T \simeq \varphi_n(t)^T\tilde{C}^T, \tag{34}$$

where $\tilde{C}$ is the $(n+1)\times(n+1)$ multiplication operational matrix. To illustrate the calculation procedure, we let

$$C^T\varphi_n(t)\varphi_n(t)^T = \big[c_0(t), c_1(t), \ldots, c_n(t)\big].$$

Now we approximate $C^T\varphi_n(t)\varphi_n(t)^T$ as follows:

$$c_i(t) \simeq \sum_{j=0}^{n} \tilde{c}_{ij}s_j(t) = \tilde{C}_i^T\varphi_n(t), \tag{35}$$

where

$$\tilde{C}_i = [\tilde{c}_{i0}, \tilde{c}_{i1}, \ldots, \tilde{c}_{in}]^T. \tag{36}$$

Using (21), we obtain

$$c_i^k = \left\langle \sum_{j=0}^{n} \tilde{c}_{ij}s_j(t), s_k(t) \right\rangle = \sum_{j=0}^{n} \tilde{c}_{ij}d_{jk}(t), \qquad j,k = 0,\ldots,n, \tag{37}$$

where $c_i^k = \big\langle c_i(t), s_k(t)\big\rangle$ and

$$d_{jk} = \big\langle s_j(t), s_k(t)\big\rangle. \tag{38}$$

Therefore

$$C_i^T = \tilde{C}_i^T D, \tag{39}$$

and

$$C_i = [c_i^0, c_i^1, \ldots, c_i^n]^T, \tag{40}$$

where $D = [d_{jk}]$ is a matrix of order $(n+1)\times(n+1)$ given by (38). Also, $\tilde{C}_i^T$ in (39) is given by

$$\tilde{C}_i^T = C_i^T D^{-1}. \tag{41}$$

Hence, we get the operational matrix of multiplication as

$$\tilde{C}_0 = [\tilde{c}_{ij}]_{0\le i,j\le n}. \tag{42}$$

5 MPs for solving multi-dimensional FOCPs

Using Theorem 1, we can approximate the state functions $x_i(t)$ and the control functions $u_j(t)$ as

$$x_i(t) \approx C_i^T \varphi_n(t), \qquad i = 1, \dots, n, \tag{43}$$

$$u_j(t) \approx B_j^T \varphi_n(t), \qquad j = 1, \dots, k, \tag{44}$$

where $C_i, B_j \in R^{(m\times 1)\times 1}$. By using (32) and (43), we can write

$$D^\alpha x_i(t) \approx C_i^T D_\alpha \varphi_n(t), \qquad i = 1, \dots, n.$$

As a result, problem (1)–(4) reduces to

$$\text{Minimize} \int_0^1 f(t, C_1^T \varphi_n(t), \dots, C_n^T \varphi_n(t), B_1^T \varphi_n(t), \dots, B_k^T \varphi_n(t))dt,$$

subject to

$$C_i^T T \varphi_n(t) = g_i\big(t, C_1^T \varphi_n(t), \dots, C_n^T \varphi_n(t), B_1^T \varphi_n(t), \dots, B_k^T \varphi_n(t)\big),$$

with the initial conditions

$$C_i^T \varphi_n(0) = x_{0,i}, \qquad i = 1, \dots, n. \tag{45}$$

Since functions f, g_i are polynomials, we get the following approximations:

$$\int_0^1 f(t, X(t), U(t))dt \approx F(C_1, \dots, C_n, B_1, \dots, B_k), \tag{46}$$

$$g_i(t, X(t), U(t)) \approx G_i(C_1, \dots, C_n, B_1, \dots, B_k)\varphi_n(t), \qquad i = 1, \dots, n, \tag{47}$$

where $F, G_i : R^{(m+1)\times n} \times R^{(m+1)\times k} \to R^{1\times(m+1)}$. For each $i = 1, \dots, n$, we can generate algebraic equations from (47) as follows:

$$\tilde{G}_{i,j} = \int_0^1 (C_i^T T - G_i(C_1, \ldots, C_n, B_1, \ldots, B_k)\varphi_n(t)B_{jn}(t))dt = 0, \quad j = 0, \ldots, n-1,$$

and from (45), we set $\tilde{G}_{i,n} = C_i^T \varphi_n(0) - x_{0,i}$.

Finally, the FOCP (1)–(4) has been reduced to a parameter optimization problem, which can be stated as follows:

Find C_i and B_j that

$$\text{Min} \quad F(C_1, \ldots, C_n, B_1, \ldots, B_k)$$

$$s.t. \quad \tilde{G}_{i,j}(C_1, \ldots, C_n, B_1, \ldots, B_k) = 0, \quad i = 1, \ldots, n, j = 0, \ldots, k.$$

We define the Lagrange function for the above problem as

$$L(C_1, \ldots, C_n, B_1, \ldots, B_k, \tilde{\lambda}_1, \ldots, \tilde{\lambda}_n) = F(C_1, \ldots, C_n, B_1, \ldots, B_k) + \sum_{i=1}^{n}\sum_{j=0}^{k} \tilde{\lambda}_{i,j}\tilde{G}_{i,j}(C_1, \ldots, C_n, B_1, \ldots, B_k),$$

and consider the necessary conditions for the extremum and obtain the corresponding system of algebraic equations

$$\frac{\partial L}{\partial C_i} = 0, \quad \frac{\partial L}{\partial \tilde{\lambda}_i} = 0, \quad i = 1, \ldots, n,$$
$$\frac{\partial L}{\partial B_j} = 0, \quad j = 1, \ldots, k.$$

These equations can be solved for C_i, B_j, and $\tilde{\lambda}_i$ by Newton's iterative method. Then, we get the approximate value of the state functions $x_i(t)$ and the control functions $u_j(t)$ from (43) and (44), respectively.

6 Error estimation and convergence analysis

In the following theorem, the error estimation for the approximated functions will be expressed in terms of Gram determinant. For any given elements $x_1, x_2, \ldots, x_n$ in a Hilbert space H, the Gram determinant of these elements is defined as follows [20]:

$$G(x_1, x_2, \ldots, x_n) = \begin{vmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle & \ldots & \langle x_1, x_n \rangle \\ \langle x_2, x_1 \rangle & \langle x_2, x_2 \rangle & \ldots & \langle x_2, x_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n, x_1 \rangle & \langle x_n, x_2 \rangle & \ldots & \langle x_n, x_n \rangle \end{vmatrix}. \tag{48}$$

Theorem 2. Suppose that H is a Hilbert space and that Y is a closed subspace of H such that $\dim Y < \infty$ and $y_1, y_2, \ldots, y_n$ is any basis for Y. Let x be an arbitrary element in H and let y_0 be the unique best approximation to x out of Y. Then from [20]

$$\|x - y_0\|_2^2 = \frac{G(x, y_1, y_2, \ldots, y_n)}{G(y_1, y_2, \ldots, y_n)}, \tag{49}$$

where

$$G(x, y_1, y_2, \ldots, y_n) = \begin{vmatrix} \langle x, x \rangle & \langle x, y_1 \rangle & \ldots & \langle x, y_n \rangle \\ \langle y_1, x \rangle & \langle y_1, y_1 \rangle & \ldots & \langle y_1, y_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle y_n, x \rangle & \langle y_n, y_1 \rangle & \ldots & \langle y_n, y_n \rangle \end{vmatrix}. \tag{50}$$

The following theorems illustrate that by increasing the number of MPs the error tends to zero.

Theorem 3. Let f be an arbitrary element in H. Then the function f has the best unique approximation on Y like $f_n \in Y$; that is,

$$\text{there exists } f_n \in Y \text{ s.t. } \quad \text{for all } y \in Y : \|f - f_n\|_2 \leq \|f - y\|_2, \tag{51}$$

where $\|f\|_2 = \sqrt{\langle f, f \rangle}$ and $\langle \cdot, \cdot \rangle$ denotes the inner product. Since $f_n \in Y$, therefore f_n is a linear combination of the spanning basis of Y; that is, there are $n + 1$ coefficients

$$C = [c_0, c_1, \ldots, c_n] \in R, \tag{52}$$

such that

$$f(t) \simeq f_n(t) = \sum_{j=0}^{n} c_j s_j(t) = C^T \varphi_n(t), \tag{53}$$

where

$$\|f - f_n\|_2 \to \min . \tag{54}$$

Then C can be obtained by

$$C = Q^{-1}\langle f(t), \varphi_n(t)\rangle, \tag{55}$$

where

$$Q = \langle \varphi_n(t), \varphi_n(t)\rangle = \int_0^1 \varphi_n(t)\varphi_n(t)^T dt. \tag{56}$$

Theorem 4. Suppose that $f(t) \in L^2[0,1]$ is approximated by $f_n(t)$ as

$$f_n(t) = \sum_{i=0}^{n} c_i \beta_i(t) = C^T \varphi_n(t), \tag{57}$$

where C and $\varphi_n(t)$ are defined respectively in (55) and (16). Then

$$\lim_{m\to\infty} \|f(t) - f_n(t)\|_{L^2[0,1]} = 0. \tag{58}$$

The error vector eI^α of the operational matrix is given by

$$eI^\alpha = [eI_0^\alpha, eI_1^\alpha, \ldots, eI_n^\alpha]^T = P\varphi_n(t) - I^\alpha \varphi_n(t). \tag{59}$$

From (49) and Theorem 2, we have

$$\left\| t^{i-2k+\alpha} - \sum_{j=0}^{n} b_j s_j \right\|_2 = \left(\frac{G(t^{i-2k+\alpha}, s_i(t), s_1(t), \ldots, s_n(t))}{G(s_0(t), s_1(t), \ldots, s_n(t))} \right). \tag{60}$$

Hence, according to (59) and (28), we have

$$\begin{aligned} \|eI_i^\alpha\|_2 &= \left| I^\alpha s_i(t) - \sum_{j=0}^{n} B_{ij} s_j(t) \right|_2 \\ &\le \left[\left(-\frac{1}{2}\right)^{i+j} (i-1)!(j-1)! \right] \\ &\quad \times \sum_{k=0}^{h(\frac{i}{2})} \sum_{L=0}^{h(\frac{j}{2})} \binom{j}{L} \binom{i}{k} (-1)^{k+L} \frac{1}{(i-2k-1)!(j-2L-1)!} \\ &\quad \times \frac{1}{i+j-2L-2k+\alpha+1} \left| t^{i-2k+\alpha} - \sum_{j=0}^{n} b_j s_j \right|_2 \end{aligned}$$

$$\leq \left[\left(-\frac{1}{2}\right)^{i+j}(i-1)!(j-1)!\right] \times \sum_{k=0}^{h\left(\frac{i}{2}\right)}\sum_{L=0}^{h\left(\frac{j}{2}\right)}\binom{j}{L}\binom{i}{k}(-1)^{k+L}\frac{1}{(i-2k-1)!(j-2L-1)!} \times \frac{1}{i+j-2L-2k+\alpha+1} \times \left(\frac{G(t^{i-2k+\alpha}, s_0(t), s_1(t), \ldots, s_n(t))}{G(s_0(t), s_1(t), \ldots, s_n(t))}\right)^{\frac{1}{2}}. \quad (61)$$

By considering Theorem 4 and using (61), one can conclude that by increasing the number of the Mott bases the vector eI^{α} tends to zero.

7 Numerical examples

To demonstrate the applicability of the numerical scheme, we apply the present method for the following illustrative examples.

Example 1. [16] Consider the following two-dimensional FOCP:

$$\begin{aligned} \text{Min} \quad & J[u,x] = \frac{1}{2}\int_0^1 (x^2(t)+u^2(t))dt, \\ \text{s.t.} \quad & D^{\alpha}x(t) = -x(t)+u(t), \\ & x(0) = 1. \end{aligned}$$

Our aim is to find the pair $(x(t), u(t))$, which minimizes the performance index J. The exact optimal solution of this problem for $\alpha = 1$ is as follows:

$$\begin{aligned} x(t) &= \cosh(\sqrt{2}t) + \beta\sinh(\sqrt{2}t), \\ u(t) &= (1+\sqrt{2}\beta)\cosh(\sqrt{2}t) + (\sqrt{2}+\beta)\sinh(\sqrt{2}t), \\ \beta &= -\frac{\cosh(\sqrt{2}) + \sqrt{2}\sinh(\sqrt{2})}{\sqrt{2}\cosh(\sqrt{2}) + \sinh(\sqrt{2})}. \end{aligned}$$

In Tables 1 and 2, the absolute errors of our proposed method and the method presented in [16] for $\alpha = 1$ are given. We see that the absolute errors of our approach are less than that of the method presented in [16]. Moreover, in Table 3, we show the comparison of the approximate optimal value of objective functional for $\alpha = 1$. It can be seen that the approximate value J tends to $J^* = 0.192909298093179135 6$.

Table 1: Comparison of the absolute errors of approximate optimal state for Example 1

t	Present Method	Method in [16]
0.1	6.23e-9	2.11e-5
0.2	4.60e-10	9.71e-6
0.3	8.94e-9	4.08e-7
0.4	1.01e-9	5.76e-7
0.5	9.33e-9	5.66e-6
0.6	3.89e-10	9.25e-6
0.7	8.67e-9	8.35e-6
0.8	9.96e-10	4.34e-6
0.9	5.83e-9	2.59-6

Table 2: Comparison of the absolute errors of approximate optimal control for Example 1

t	Present Method	Method in [16]
0.1	1.1e-7	6.74e-6
0.2	1.57e-7	3.17e-6
0.3	3.49e-9	5.92e-7
0.4	1.41e-7	7.10e-7
0.5	3.84e-9	2.01e-6
0.6	1.38e-7	2.71e-6
0.7	9.80e-9	2.11e-6
0.8	1.50e-7	8.59e-7
0.9	1.08e-7	8.93-8

Table 3: Comparison of the approximate optimal value of J for Example 1

n	Present Method		Method in [16]	
	J	error of J	J[16]	error of J [16]
7	0.1929092980931	7.6e-15	0.192909308499	1.04e-8
8	0.1929092980931	7.6e-15	0.192909298458	3.64e-10
9	0.1929092980931	4.5e-18	0.192909298107	1.41e-11

Example 2. [13] Consider the following two-dimensional FOCP:

$$\begin{aligned} \text{Min} \quad & J[u,x] = \frac{1}{2}\int_0^1 (x_1^2(t) + x_2^2 + u^2(t))dt, \\ s.t. \quad & \mathrm{D}^\alpha x_1(t) = -x_1(t) + x_2(t) + u(t), \\ & \mathrm{D}^\alpha x_2(t) = -2x_2(t), \\ & x_1(0) = x_2(0) = 1. \end{aligned}$$

At $\alpha = 1$, the exact solutions are

$$x_1(t) = 0.018352e^{\sqrt{2}t} + 2.48165e^{-\sqrt{2}t} - \frac{3e^{-2t}}{2},$$
$$x_2(t) = e^{-2t},$$
$$u(t) = 0.044305e^{\sqrt{2}t} - 1.0279322e^{-\sqrt{2}t} + \frac{e^{-2t}}{2}$$

In Tables 4–6, the absolute errors when $\alpha = 1$ and $n = 3, 4, 6$ are demonstrated, and in Figure 1 (a-c), the results for $\alpha = 0.9$ and $n = 3, 4, 6$ are plotted. In Table 7, the comparison of our numerical results for the minimum values of J with different values of α at $n = 6, 7, 8$ with the results obtained in [24] is tabulated. In Figure 2 (a-c), we show that when α tends to 1, the approximate solutions tend to the exact solutions. In [13], a reasonable result was achieved with a large number of approximations ($n = 64, 128$), while in the present work, we achieve a satisfactory result with at most six elements of the Mott basis, which demonstrates the efficiency of the new method even when the number of approximations is not so great.

Table 4: Absolute errors of $x_1(t)$ for $\alpha = 1$ at various choices of n for Example 2

	n		
t	3	4	6
0.1	6.2546×10^{-4}	7.4441×10^{-4}	1.5002×10^{-5}
0.2	4.5464×10^{-3}	9.0972×10^{-4}	8.3446×10^{-6}
0.3	3.9041×10^{-3}	1.4174×10^{-5}	1.0492×10^{-5}
0.4	1.1462×10^{-3}	7.5444×10^{-4}	5.1504×10^{-6}
0.5	1.9425×10^{-3}	8.7140×10^{-4}	1.2149×10^{-5}
0.6	4.0841×10^{-3}	3.1893×10^{-4}	1.0835×10^{-6}
0.7	4.3836×10^{-3}	5.3794×10^{-4}	1.2403×10^{-5}
0.8	2.2303×10^{-3}	1.0500×10^{-3}	5.0792×10^{-6}
0.9	2.7809×10^{-3}	3.6221×10^{-4}	1.6075×10^{-5}

Table 5: Absolute errors of $x_2(t)$ for $\alpha = 1$ at various choices of n for Example 2

	n		
t	3	4	6
0.1	1.8817×10^{-4}	1.0437×10^{-3}	1.3953×10^{-5}
0.2	1.0761×10^{-2}	1.4093×10^{-3}	7.5730×10^{-6}
0.3	1.0648×10^{-2}	1.3573×10^{-4}	9.8425×10^{-6}
0.4	4.7251×10^{-3}	1.0583×10^{-3}	4.5687×10^{-6}
0.5	3.0146×10^{-3}	1.3381×10^{-3}	1.1291×10^{-5}
0.6	9.3023×10^{-3}	5.9271×10^{-4}	1.2293×10^{-6}
0.7	1.1462×10^{-2}	6.9627×10^{-4}	1.1414×10^{-5}
0.8	7.3019×10^{-3}	1.5621×10^{-3}	4.8958×10^{-6}
0.9	4.9714×10^{-3}	6.4096×10^{-4}	1.4835×10^{-5}

Table 6: Absolute errors of $u(t)$ for $\alpha = 1$ at various choices of n for Example 2

	n		
t	3	4	6
0.1	6.6070×10^{-4}	2.3079×10^{-4}	5.0375×10^{-6}
0.2	1.7968×10^{-4}	3.1360×10^{-4}	2.7248×10^{-6}
0.3	7.8105×10^{-4}	3.1989×10^{-5}	3.5467×10^{-6}
0.4	9.3073×10^{-4}	2.3164×10^{-4}	1.6400×10^{-6}
0.5	6.1527×10^{-4}	2.9416×10^{-4}	4.0622×10^{-6}
0.6	1.9916×10^{-5}	1.3128×10^{-4}	4.4423×10^{-7}
0.7	7.0047×10^{-4}	1.5236×10^{-4}	4.1084×10^{-6}
0.8	1.0445×10^{-3}	3.4459×10^{-4}	1.7637×10^{-6}
0.9	5.7713×10^{-4}	1.4011×10^{-4}	5.3528×10^{-6}

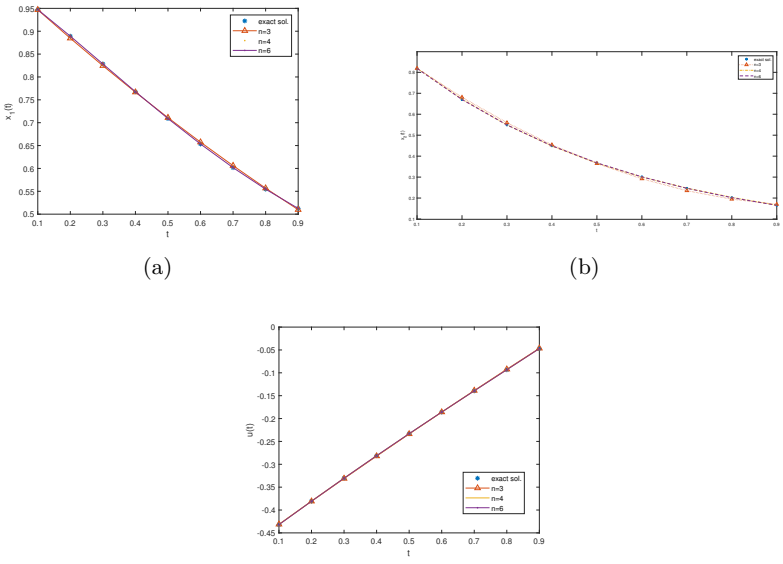
Table 7: Values of J when α approaches to 1, for Example 2

	PresentMethod			*Method of* [29]		
α	$n = 6$	$n = 7$	$n = 8$	$n = 6$	$n = 7$	$n = 8$
0.6	0.329 09	0.329 09	0.329 10			
0.7	0.351 62	0.351 62	0.351 62			
0.8	0.376 27	0.376 27	0.376 27	0.378 33	0.377 56	0.377 17
0.9	0.403 08	0.403 08	0.403 08	0.403 98	0.403 66	0.403 46
0.99	0.429 00	0.429 00	0.429 00	0.429 09	0.429 05	0.429 04
0.9999	0.431 95	0.431 95	0.431 95	0.431 96	0.431 96	0.431 96
1	0.431 98	0.431 98	0.431 98	0.431 99	0.431 98	0.431 98

Example 3. [1, 5, 22] Consider the following FOCP:

$$\begin{aligned} \text{Min} \quad & J[u,x] = \frac{1}{2}\int_0^1 (x^2(t) + u^2(t))dt, \\ s.t. \quad & \mathrm{D}^\alpha x(t) = tx(t) + u(t), \quad 0 \le \alpha \le 1, \\ & x(0) = 1. \end{aligned}$$

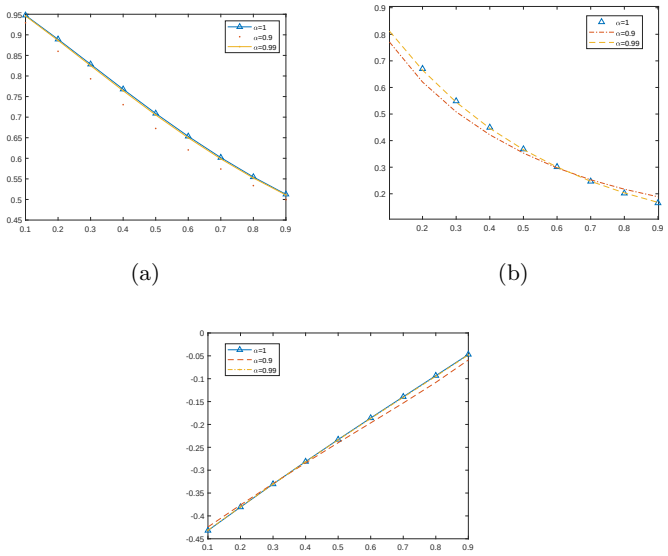
This problem has the exact solution $J = 0.4842676962287272$. In Figure 3 (a,b), the obtained results of the variables $x(t)$ and $u(t)$ are plotted for different values of α. In Figure 4 (a,b), the comparison between the exact solutions and the proposed method is plotted for $n = 8$ and $\alpha = 1$. In Table 8, the comparison of our numerical results for the minimum values of J for $\alpha = 1, 0.99$ at $n = 5$ with the results obtained in [18, 24] is tabulated. Obviously, our estimated results are in good agreement with them.



(a) (b)

(c)

Figure 1: The behavior of the approximate solutions of Example 2 for $n = 3, 4, 6$ and $\alpha = 0.9$, with exact solution. $(a)x_1(t)$, $(b)x_2(t)$, $(c)u(t)$.



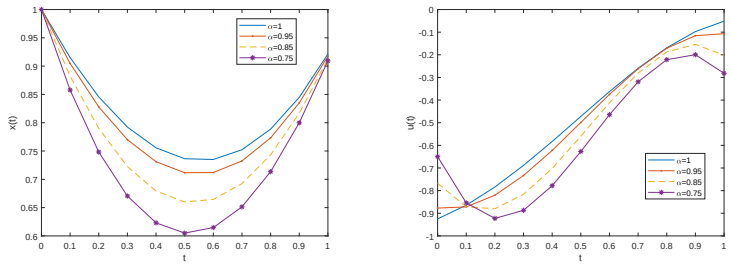
(a) (b)

(c)

Figure 2: The behavior of the approximate solutions of Example 2 for $n = 6$ and $\alpha = 0.9, 0.99, 1$. $(a)x_1(t)$, $(b)x_2(t)$, $(c)u(t)$.

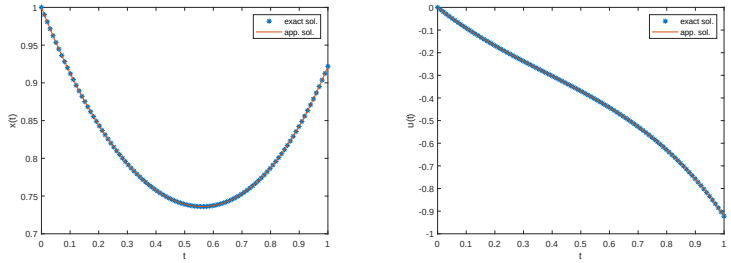
Table 8: The estimated values of J for different values of α and $n = 5$ for Example 3

α	Present Method	Method of [24]	Method of [18]
1	0.484267	0.484268	0.484268
0.99	0.483461	0.483468	0.483463
0.9	0.475874	0.476067	0.475883



(a) (b)

Figure 3: The behavior of the approximate solutions of Example 3 for $n = 8$ and $\alpha = 0.75, 0.85, 0.95, 1$, with exact solution. $(a)x(t)$, $(b)u(t)$.



(a) (b)

Figure 4: Comparison between the exact solutions and the proposed method of Example 3 for $n = 8$ and $\alpha = 1$. $(a)x(t)$, $(b)u(t)$.

Example 4. [30] In this example, the vibration of a spring-mass-damper system subjected to an external force is considered. In particular, the response to harmonic excitations, impulses, and step forcing functions is examined. In many environments, rotating machinery, motors, and so on cause periodic motions of structures to induce vibrations into other mechanical devices and structures nearby. On summing the forces, the equation for the forced vibration of the system in Figure 5 is obtained. It is common to approximate the driving forces $F(t)$

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t),$$

where m, c, and k are fixed numbers. Also $F(t)$ represents the control force derived from the action of an actuator force represented by $F(t) = bu(t)$, where b is a fixed number. The linear regulator problem has a specific application in the vibration suppression, and the performance index for this problem is defined as

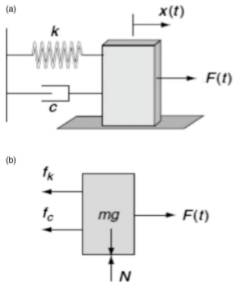


Figure 5: (a) Schematic of the forced mass–damper system assuming no friction on the surface and (b) free body diagram of the system of part (a) for Example 4

$$J = \frac{1}{2}\int_{t_0}^{t_f} (x^2(t) + au^2(t))dt,$$

where t_0 and t_f are initial and final times, respectively. We introduce the usual state variable notation

$$\begin{cases} x_1 = x, \\ x_2 = {}_0^c D_t^\alpha x(t). \end{cases}$$

Then, the equation of motion in a order can be written as

$$\begin{cases} {}_0^c D_t^\alpha x_1(t) = x_2, \\ {}_0^c D_t^\alpha x_2(t) = -\dfrac{k}{m}x_1 - \dfrac{c}{m}x_2 + \dfrac{b}{m}u. \end{cases}$$

By selecting $t_0 = 0$, $t_f = 1$, $c = 2$, and $m = a = b = k = 1$, the boundary conditions for this example are considered as

$$\begin{cases} x(0) = {}_0^c D_t^\alpha x(t)(0), \\ x(1) = {}_0^c D_t^\alpha x(t)(1), \end{cases}$$

or

$$\begin{cases} x_1(0) = x_2(0) = 1, \\ x_1(1) = x_2(1) = 0. \end{cases}$$

For $\alpha = 1$, we obtain the following exact solutions:

$$\begin{aligned} x_{1e}(t) =& [\frac{1393}{95}e^{-\frac{167}{152}t} + \frac{676}{219}e^{\frac{167}{152}t}](\sin\frac{76}{167}t) \\ &+ [\frac{1574}{423}e^{-\frac{167}{152}t} - \frac{1151}{423}e^{\frac{167}{152}t}](\cos\frac{76}{167}t), \\ x_{2e}(t) =& [\frac{2838}{613}e^{\frac{167}{152}t} - \frac{12872}{723}e^{-\frac{167}{152}t}](\sin\frac{76}{167}t) \\ &+ [\frac{1021}{395}e^{-\frac{167}{152}t} - \frac{626}{395}e^{\frac{167}{152}t}](\cos\frac{76}{167}t), \\ u_e(t) =& [\frac{16157}{890}e^{\frac{167}{152}t} - \frac{1134}{443}e^{-\frac{167}{152}t}](\sin\frac{76}{167}t) \\ &- [\frac{2547}{461}e^{\frac{167}{152}t} + \frac{2591}{1263}e^{-\frac{167}{152}t}](\cos\frac{76}{167}t), \end{aligned}$$

and $J_e = 13.00484741498823$, where $x_{1e}(t) = x(t)$. In Table 9, the results of J for different values of α and n are listed. It is seen that with the increase in the number of the Mott basis, the approximate value of J converges to the exact solution. Also Tables 10–12 demonstrate the approximation of $x_1(t)$, $x_2(t)$, and $u(t)$ for different values of n with $\alpha = 1$. Figure 6 (a-d) illustrates the behavior of state variables $x_1(t), x_2(t)$, control variable $u(t)$ and performance index J, respectively, for $n = 7$ and for different values of α with the exact solutions. Table 13 demonstrates the approximation of J, for $n = 8$ and different values of α. Also in Table 14, the absolute errors of $J, x_1(t), x_2(t)$, and $u(t)$ for Example 4 are calculated at various choices of n and $\alpha = 1$.

Table 9: Approximate solutions of J for different values of α and n for Example 4

	α		
n	1	0.99	0.9
3	**13.0**1904761904762	12.644 81	10.531 56
5	**13.00484**78586403	12.638 42	10.209 67
7	**13.00484741498**875	12.649 19	10.210 82
9	**13.00484741498822**	12.654 24	10.220 66
11	**13.00484741498823**	12.656 99	10.227 68

H]

Table 10: Absolute errors of $x_1(t)$ for $\alpha = 1$ at various choices of n for Example 4

	n		
t	5	7	11
0	0	0	0
0.1	8.0624×10^{-6}	2.8936×10^{-9}	3.3240×10^{-13}
0.2	8.9073×10^{-6}	3.8441×10^{-9}	2.0139×10^{-13}
0.3	1.5140×10^{-6}	5.1600×10^{-9}	5.0016×10^{-13}
0.4	1.1564×10^{-5}	2.6349×10^{-9}	8.2645×10^{-13}
0.5	1.2533×10^{-5}	7.1014×10^{-9}	3.1086×10^{-15}
0.6	4.7572×10^{-6}	1.6704×10^{-9}	8.1934×10^{-13}
0.7	4.5973×10^{-6}	5.2971×10^{-9}	4.8572×10^{-13}
0.8	7.9685×10^{-6}	3.1233×10^{-9}	1.9929×10^{-13}
0.9	3.9680×10^{-6}	2.7298×10^{-9}	3.1830×10^{-13}
1	1.3322×10^{-15}	1.2212×10^{-15}	4.4409×10^{-16}

Table 11: Absolute errors of $x_2(t)$ for $\alpha = 1$ at various choices of n for Example 4

	n		
t	5	7	11
0	0	0	0
0.1	7.8816×10^{-5}	2.6708×10^{-8}	1.0362×10^{-11}
0.2	6.2887×10^{-5}	6.7790×10^{-8}	1.4781×10^{-11}
0.3	1.2314×10^{-4}	4.6618×10^{-8}	1.1712×10^{-11}
0.4	6.2434×10^{-5}	8.3958×10^{-8}	5.3897×10^{-12}
0.5	4.2027×10^{-5}	7.2681×10^{-9}	1.6518×10^{-11}
0.6	1.003×10^{-4}	8.4910×10^{-8}	5.3266×10^{-12}
0.7	7.2908×10^{-5}	3.2740×10^{-8}	1.1480×10^{-11}
0.8	8.3314×10^{-6}	6.6651×10^{-8}	1.4331×10^{-11}
0.9	5.8537×10^{-5}	1.7511×10^{-8}	9.9403×10^{-12}
1	7.7715×10^{-16}	1.2212×10^{-15}	7.7716×10^{-16}

Table 12: Absolute errors of $u(t)$ for $\alpha = 1$ at various choices of n for Example 4

	n		
t	5	7	11
0	3.9149×10^{-3}	4.0395×10^{-6}	1.2538×10^{-9}
0.1	9.5758×10^{-3}	1.5795×10^{-6}	2.1342×10^{-10}
0.2	1.3556×10^{-3}	6.0445×10^{-7}	2.8410×10^{-11}
0.3	1.7251×10^{-4}	1.1567×10^{-6}	2.5282×10^{-10}
0.4	8.5641×10^{-4}	2.3114×10^{-7}	2.8751×10^{-10}
0.5	1.0077×10^{-3}	1.1500×10^{-6}	3.2260×10^{-11}
0.6	3.5723×10^{-4}	3.5077×10^{-7}	3.0539×10^{-10}
0.7	5.0337×10^{-4}	1.0067×10^{-6}	2.0282×10^{-10}
0.8	8.2804×10^{-4}	6.34×10^{-7}	8.3564×10^{-11}
0.9	1.5161×10^{-4}	1.3112×10^{-6}	2.4373×10^{-10}
1	1.1831×10^{-3}	3.307×10^{-6}	1.1912×10^{-9}

Table 13: Approximation values of J for $n = 8$ and different values of α

	$\alpha = 1$
n	
1	13.004 847 414 988 234 117 130 714 182 4
0.99	12.676 065 068 414 313 965 482 218 517 6
0.9	10.416 927 546 051 989 730 195 819 221 4
0.8	9.004 530 163 410 743 299 622 066 577 65
0.7	8.363 433 733 210 117 712 725 797 350 86

Table 14: Absolute errors of J, x_1, x_2, u for $\alpha = 1$ and different values of n

	error			
n	error of J	error of x_1	error of x_2	error of u
11	7.6085×10^{-16}	4.846×10^{-13}	1.04×10^{-11}	2.792×10^{-10}
9	7.5908×10^{-15}	3.974×10^{-12}	7.744×10^{-11}	1.9×10^{-9}
8	3.8474×10^{-15}	2.729×10^{-10}	4.563×10^{-9}	9.6×10^{-8}
7	5.2381×10^{-12}	3.908×10^{-9}	5.603×10^{-8}	1.024×10^{-6}
5	4.4365×10^{-7}	7.474×10^{-6}	7.234×10^{-5}	9.419×10^{-4}

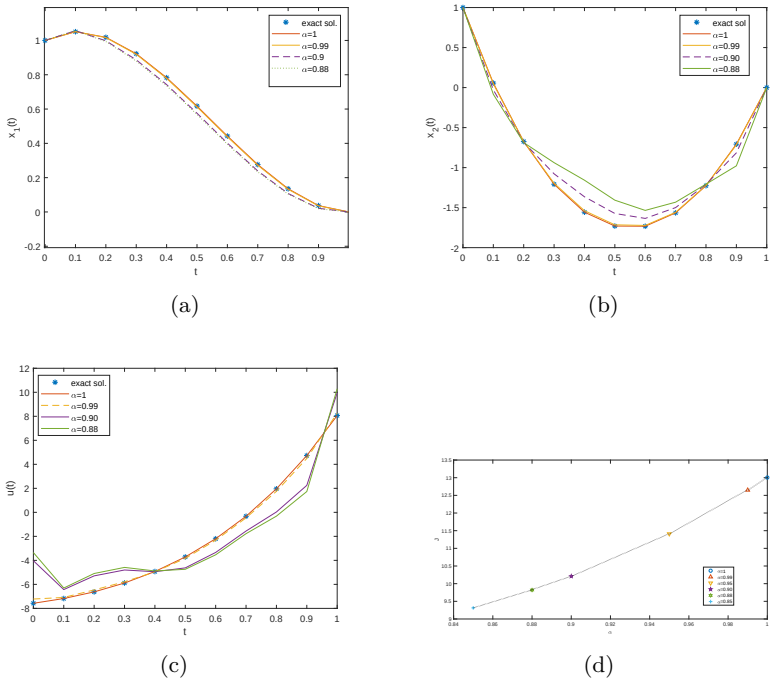


Figure 6: The behavior of the approximate solutions of Example 4 for $n = 7$ and $\alpha = 0.88, 0.90, 0.99, 1$, with exact solution. $(a)x_1(t)$, $(b)x_2(t)$, $(c)u(t)$, $(d)J$.

8 Conclusions

In this paper, a new numerical method has been derived to find the approximate solutions of the multi-dimensional FOCPs; this numerical method uses MPs. The Mott fractional integration matrix reduced the FOCP into an equivalent integral problem. By using the Mott fractional derivative and multiplication matrix, we can transform the equivalent functional integral equation problem into an algebraic system of equations, where this new problem is most easy to solve. Some examples are presented to demonstrate the validity and applicability of the new method. MATLAB (2018) is used for computations in this study.

References

1. Agrawal, O.P. *On a general formulation for the numerical solution of optimal control problems,* Int. J. Control 50 (1989) 627–638.

2. Agrawal O.P. *A general formulation and solution scheme for fractional optimal control problem*, Nonlinear Dynam, 38 (2004) 323–337.

3. Agrawal, O.P. and Baleanu, D. *A Hamiltonian formulation and a direct numerical scheme for fractional optimal control problems,* J. Vib. Control. 13 (2007) 1269–1281.

4. Ahmadi Darani, M. and Saadatmandi, A. *The operational matrix of fractional derivative of the fractional-order Chebyshev functions and its applications,* Comput. Methods Differ. Equ. 5(1) (2017), 67–87.

5. Akbarian, T. and Keyanpour, M. *A new approach to the numerical solution of fractional order optimal control problems,* Appl. Appl. Math. 8(2) (2013), 523–534.

6. Ashpazzadeh, E., Lakestani, M. and Yildirim, A. *Biorthogonal multiwavelets on the interval for solving multidimensional fractional optimal control problems with inequality constraint,* Optimal Control Appl. Methods 41(5) (2020), 1477–1494.

7. Atanacković, T.M., Pilipović, S., Stanković, B., and Zorica, D. *Fractional calculus with applications in mechanics. Wave propagation, impact and variational principles. Mechanical Engineering and Solid Mechanics Series,* ISTE, London; John Wiley & Sons, Inc., Hoboken, NJ, 2014.

8. Bagley, RL and Torvik PJ (1983) A theoretical basis for the application of fractional calculus to viscoelasticity, J. Rheol., 27, 201–210.

9. Baleanu, D., Defterli, O. and Agrawal, O.P. *A central difference numerical scheme for fractional optimal control problems,* J. Vib. Control. 15(4) (2009) 583–597.

10. Bohannan, G.W. *Analog fractional order controller in temperature and motor control applications,* J. Vib. Control. 14(9-10) (2008) 1487–1498.

11. Caputo, M. *Linear models of dissipation whose Q is almost frequency independent II,* Reprinted from Geophys. J. R. Astr. Soc. 13 (1967), no. 5, 529–539. Fract. Calc. Appl. Anal. 11 (2008), no. 1, 4–14.

12. Chow, T.S. *Fractional dynamics of interfaces between soft-nanoparticles and rough substrates,* Phys. Lett. A, 342, (2005) 148–155.

13. Defterli, O. *A numerical scheme for two-dimensional optimal control problems with memory effect,* Comput. Math. Appl. 59(5) (2010) 1630–1636.

14. Garrappa, R. *Numerical solution of fractional differential equations: A survey and a software tutorial,* https://www.mdpi.com/2227-7390/6/2/16

15. Ghomanjani, F. *A numerical technique for solving fractional optimal control problems and fractional Riccati differential equation,* J. Egyptian Math. Soc. 24(4) (2016), 638–643.

16. Habibli, M. and Noori Skandari, M.H. *Fractional Chebyshev pseudospectral method for fractional optimal control problems,* Optimal Control Appl. Methods 40(3) (2019), 558–572.

17. Hoda F. Ahmed *A numerical technique for solving multidimensional fractional optimal control problems,* Journal of Taibah University for Science, 12(5), (2018) 494–505.

18. Keshavarz, E., Ordokhani, Y. and Razzaghi, M. *A numerical solution for fractional optimal control problems via Bernoulli polynomials,* 22(18) (2016), 3889–3903.

19. Jesus, I.S. and Machado, J.T. *Fractional control of heat diffusion systems,* Nonlinear Dyn. 54 (2008) 263–282.

20. Kreyszig, E. *Introductory functional analysis with applications,* New York: John Wiley and sons. Inc, 1978.

21. Kruchinin, D.V. *Explicit formulas for some generalized Mott polynomials,* Advanced Studies in Contemporary Mathematics, 24 (3) (2014) 327–332.

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

23. Maheswaran, A. and Elango, *Characterization of delta operator for Euler, Bernoulli of second kind and moot polynomials,* Int. J. Pure Appl. Math. 109 (2) 2016, (2016) 371–384.

24. Moghaddam, M.A., Yousef, E. and Lakestani, M. *Solving fractional optimal control problems using Genocchi polynomials,* Comput. Methods Differ. Equ. 9(1) (2021), 79–93.

25. Mott, N.F. *The polarization of electrons by double scattering*, Proc. R. Soc. Lond. A 135, (1932) 429–458.

26. Nemati. A., Yousefi, S.A. *A numerical method for solving fractional optimal control problems using Ritz method*, J. Comput. Nonlinear Dyn. 11(5) 051015-1 (2016) 7 pp.

27. Oldham, K.B. and Spanier, J. *The fractional calculus: Theory and application of differentiation and integration to arbitrary order,* Acad. Press, N. York and London, 1974.

28. Ross, B. *A brief history and exposition of the fundamental theory of fractional calculus, in fractional calculus and its Applications,* Lecture Notes in Mathematics, Vol.457, pp. 1–36, Springer, Berlin, 1975.

29. Weisstein, Eric W. "Mott Polynomial." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/MottPolynomial

30. Yari A *Numerical solution for fractional optimal control problems by Hermite polynomials,* J. Vib. Control. 25 (2021) 5–6.

31. Yousefi, S.A., Lotfi, A. and Dehghan, M. *The use of a Legendre multiwavelet collocation method for solving the fractional optimal control problems,* J. Vib. Control. 17(13)(2011) 2059–2065.

How to cite this article
Ali Alavi, S., Haghighi, A., Yari, A., Soltanian, F. Using MOTT polynomials operational matrices to optimize multi-dimensional fractional optimal control problems. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 201-227. doi: 10.22067/ijnao.2021.68926.1016

Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 1, (2022), pp 229-249
DOI:10.22067/ijnao.2021.73189.1070
https://ijnao.um.ac.ir/

Research Article

Legendre wavelet method combined with the Gauss quadrature rule for numerical solution of fractional integro-differential equations

M. Riahi Beni

Abstract

In this paper, we use a novel technique to solve the nonlinear fractional Volterra-Fredholm integro-differential equations (FVFIDEs). To this end, the Legendre wavelets are used in conjunction with the quadrature rule for converting the problem into a linear or nonlinear system of algebraic equations, which can be easily solved by applying mathematical programming techniques. Only a small number of Legendre wavelets are needed to obtain a satisfactory result. Better accuracies are also achieved within the method by increasing the number of polynomials. Furthermore, the existence and uniqueness of the solution are proved by preparing some theorems and lemmas. Also, error estimation and convergence analyses are given for the considered problem and the method. Moreover, some examples are presented and their results are compared to the results of Chebyshev wavelet, Nyström, and Newton–Kantorovitch methods to show the capability and validity of this scheme.

AMS subject classifications (2020): 47G20, 65T60, 34A08.

Keywords: Legendre wavelet; Gaussian quadrature; operational matrix; fractional Volterra-Fredholm integro-differential equations.

1 Introduction

Fractional calculus has been applied in physics in recent years, although it has a long history in mathematics. Most of the physical phenomenon can be mod-

Received 21 October 2021; revised 26 November 2021; accepted 10 December 2021
Mohsen Riahi Beni
Department of Mathematics, Higher Education Complex of Saravan, Saravan, Iran. e-mail: m.riahi@saravan.ac.ir

eled by using fractional calculus [34, 10]. Applications of fractional differential equations and fractional integral equations create a wide area of research for many researchers [3, 4, 12, 11]. It has been applied to model the nonlinear oscillation of earthquakes, fluid-dynamic traffic, frequency-dependent damping behavior of many viscoelastic materials, continuum and statistical mechanics, solid mechanics, economics, signal processing, and control theory. To better analyze these systems, it is required to obtain the solution of these equations. However, mostly, finding the analytical solution of these equations cannot be possible, so considering more accurate numerical solutions can be helpful. In the literature, there are various techniques for solving fractional ordinary differential equations (FODEs), fractional partial differential equations (FPDEs), fractional integro-differential equations (FIDEs), and dynamic systems with fractional derivatives, such as analytical and semi-analytical methods (homotopy analysis method, Adomian's decomposition method, etc.) [1, 13, 14, 15, 17, 39, 35] and numerical methods (finite difference schemes, collocation method, Tau method, etc.) [6, 36, 16, 19, 30, 5, 22].

We can find some other famous methods for the numerical solutions of these kinds of equations. For example, spline collocation [29], analytical Lie group approach [28], fractional differential transform [25], least-squares [20], rationalized Haar function [27], exp-function method [8], and many others. In [32], a novel Legendre wavelet Petrov–Galerkin method was presented for fractional Volterra integro-differential equations. The Chebyshev wavelet method [23] has been used to nonlinear fractional Volterra-Fredholm integro-differential equations (FVFIDEs) with mixed boundary conditions. Approximate solutions of Volterra-Fredholm integro-differential equations of fractional order based on the Sinc-collocation method have been discussed in [2]. In [24], the Nyström and Newton–Kantorovitch methods were described for solving FVFIDEs with mixed boundary conditions. Systems of integro-differential equations [37] have been solved by using the spectral second kind Chebyshev wavelets scheme. Also, nonlinear Volterra integro-differential equations of fractional order have been solved numerically using the Euler wavelet method [38]. The reliable modification of the Laplace Adomian decomposition method [9] has been applied to solve nonlinear Volterra-Fredholm integral equations. Haar wavelet bases have been employed for solving the integro-differential equation [7]. In [18], hybrid Bernstein block-pulse functions have been used for solving systems of fractional integro-differential equations.

Orthogonal polynomials and functions apply to different problems because of their appropriate attributes. These functions and methods such as collocation, Galerkin, and Tau, are applied to reduce the solutions of different problems to the solutions of a system of algebraic equations. In this work, the Legendre wavelets are implemented to obtain an approximate solution for nonlinear FVFIDEs. We consider the fractional FVFIDEs in a Banach space as follows:

$$\left({}_0^C D_x^\alpha y\right)(x) + Vy(x) + Fy(x) = g(x), \quad m-1 < \alpha \leq m, \quad m \in \mathbb{N}, \tag{1}$$

where ${}_0^C D_x^\alpha$ is in the Caputo sense fractional derivative, $g(x)$ is a given continuous function, $x \in J = [a, b]$, $y(x)$ is unknown function, and

$$Vy(x) = \int_a^x k_1(x,\xi) N_1(y(\xi))\, d\xi, \quad Fy(x) = \int_a^b k_2(x,\xi) N_2(y(\xi))\, d\xi,$$

in which $k_i : J \times J \to \mathbb{R}$, $i = 1, 2$, are continuous functions and $N_i : J \times \mathbb{R} \to \mathbb{R}$, $i = 1, 2$ are Lipschitz nonlinear continuous functions. Equation (1) is subject to the following mixed boundary conditions:

$$\sum_{j=1}^{m} \left[\lambda_{ij} y^{(j-1)}(a) + \eta_{ij} y^{(j-1)}(b)\right] = \gamma_i, \quad i = 1, 2, \ldots, m. \tag{2}$$

The main characteristic of this technique is that it reduces these problems to a system of algebraic equations. This approach is based on converting the FVFIDEs into mixed Volterra-Fredholm integral equations through integration, approximating various terms involved in the equation by truncated Legendre wavelet series and using the operational matrices to eliminate the integral, derivation, and product operations.

The rest of the paper is organized as follows: In section 2, some essential mathematical preliminaries and definitions of fractional calculus are introduced. In section 3, the properties of Legendre wavelets and function approximations are discussed. In the next section, the Gauss quadrature Legendre wavelet method (GQLWM) is constructed for solving FVFIDEs. In section 5, we study the convergence and error analysis of our algorithm. The existence and uniqueness of the solution are investigated in section 6. In section 7, the proposed method is applied on two examples to demonstrate the efficiency and accuracy of the present method. At last, a brief conclusion is given in section 8.

2 Basic definitions and notations of the fractional calculus

In this section, some definitions and properties of the fractional calculus, which will be used, are presented. For more details, see [21, 26, 31, 33].

Definition 1. The Gamma function is intrinsically tied with fractional calculus. The definition of the gamma function is given by

$$\Gamma(\alpha) = \int_0^\infty e^{-\xi} \xi^{\alpha-1} d\xi, \quad \mathrm{Re}(\alpha) > 0.$$

Definition 2. A real function $g(x)$, $x > 0$, is said to be in the space C_θ, $\theta \in \mathbb{R}$ if there exists a real number $p > \theta$ such that $g(x) = x^p g_1(x)$, where $g_1(x) \in C[0, \infty)$ and it is said to be in the space C_θ^k, $k \in \mathbb{N} \bigcup \{0\}$ if and only if $g^{(k)} \in C_\theta$.

Definition 3. The Riemann-Liouville fractional integral operator of order $\alpha > 0$ of a function $g \in C_\theta$, $\theta \geq -1$, is defined as

$$ {}_0I_x^\alpha g(x) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-\xi)^{\alpha-1} g(\xi) d\xi. $$

Some properties of the Riemann-Liouville fractional integral are as follows:

1. ${}_0I_x^0 g(x) = g(x)$,
2. $\left({}_0I_x^\alpha {}_0I_x^\beta g\right)(x) = \left({}_0I_x^\beta {}_0I_x^\alpha g\right)(x) = \left({}_0I_x^{\alpha+\beta} g\right)(x)$,
3. ${}_0I_x^\alpha (x-\mu)^\lambda = \frac{\Gamma(\lambda+1)}{\Gamma(\alpha+\lambda+1)}(x-\mu)^{\alpha+\lambda}, \quad \alpha \geq 0,\ \lambda > -1.$

Similar to the integer-order integration, the Riemann-Liouville fractional integral is a linear operator, which means that

$$ {}_0I_x^\alpha \left(\sum_{i=1}^m c_i f_i(x) \right) = \left(\sum_{i=1}^m c_i \right) {}_0I_x^\alpha f_i(x), $$

where $\{c_i\}_{i=1}^m$ are constants.

Definition 4. The fractional derivative of $g(x)$ in the Caputo sense is defined as

$$ {}_0^C D_x^\alpha g(x) = {}_0I_x^{n-\alpha} g^{(n)}(x), $$

for $n - 1 < \alpha \leq n$, $n \in \mathbb{N}$, $x > 0$, and $g \in C_{-1}^n$.

The relation between the Riemann-Liouville operator and Caputo operator is given by the following lemma.

Lemma 1. If $n - 1 < \alpha \leq n$, $n \in \mathbb{N}$, then ${}_0^C D_x^\alpha {}_0I_x^{n-\alpha} g(x) = g(x)$ and

$$ {}_0I_x^{n-\alpha} {}_0^C D_x^\alpha g(x) = g(x) - \sum_{k=0}^{n-1} g^{(k)}(0^+) \frac{x^k}{k!}, \quad x > 0. $$

3 Legendre wavelets and functions approximation

3.1 Properties of Legendre wavelets

Wavelets are a family of functions constructed from the dilation and translation of a single function called the mother wavelet. When the dilation

parameter a and the translation parameter b vary continuously, we have the following family of continuous wavelets as

$$\Psi_{a,b}(x) = |a|^{-\frac{1}{2}} \Psi\left(\frac{x}{a} - \frac{b}{a}\right), \;\; a, b \in \mathbb{R}, \;\; a \neq 0.$$

If we restrict the parameters a and b to discrete values as $a = a_0^{-k}$, $b = nb_0a_0^{-k}$, $a_0 > 0$, $b_0 > 0$ and n and k are positive integers, then the following family of discrete wavelets will be obtained

$$\psi_{k,n}(x) = |a_0|^{\frac{k}{2}} \psi(a_0^k x - nb_0),$$

where $\psi_{k,n}(x)$ form a wavelet basis for $L^2(\mathbb{R})$. In particular, with $a_0 = 2$ and $b_0 = 1$, $\psi_{k,n}(x)$ form an orthonormal basis.
Legendre wavelets $\psi_{n,m}(x) = \psi(k, \hat{n}, m, x)$ have four arguments; $\hat{n} = 2n - 1$ for $n = 1, 2, \ldots, 2^{k-1}$, $k \in \mathbb{Z}^+$, m as the order of Legendre polynomials and x, which is the normalized time. They are defined on the interval $[0, 1)$ by

$$\psi_{n,m}(x) = \psi(k, \hat{n}, m, x) = \begin{cases} 2^{\frac{k-1}{2}}(2m+1)^{\frac{1}{2}} L_m^*(x), & \frac{\hat{n}-1}{2^k} \leq x < \frac{\hat{n}+1}{2^k}, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $m = 0, 1, \ldots, M-1$, $n = 1, 2, \ldots, 2^{k-1}$, the dilation and translation parameters are $a = 2^{-k}$ and $b = \hat{n}2^{-k}$, respectively. Moreover, $L_m^*(x) = L_m(2^k x - \hat{n})$ are shifted Legendre polynomials on interval $[0, 1)$, which are orthogonal with respect to the weight function $w(x) = 1$. Also, L_m's can be determined by the following recurrence formulas:

$$L_0(x) = 1, \;\; L_1(x) = x,$$
$$L_{m+1}(x) = \left(\frac{2m+1}{m+1}\right) x L_m(x) - \left(\frac{m}{m+1}\right) L_{m-1}(x), \;\; m = 1, 2, \ldots.$$

3.2 Function approximation by Legendre wavelets

A function $y(x)$, which is square integrable in $[0, 1)$, may be expressed in terms of the Legendre wavelet as

$$y(x) = \sum_{n \in \mathbb{N}} \sum_{m \in \mathbb{N} \bigcup \{0\}} y_{n,m} \psi_{n,m}(x), \tag{4}$$

where $y_{n,m} = \langle y(x), \psi_{n,m}(x) \rangle$ and $< \cdot, \cdot >$ denotes the inner product. If the infinite series in (4) is truncated, then one obtains

$$y(x) \simeq \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} y_{n,m}\psi_{n,m}(x) = Y^T\Psi(x), \tag{5}$$

where the coefficient vector Y and the Legendre wavelet function vector $\Psi(x)$ are given by

$$Y = [y_{1,0}, y_{1,1}, \ldots, y_{1,M-1}, y_{2,0}, y_{2,1}, \ldots, y_{2,M-1}, \ldots, y_{2^{k-1},0}, y_{2^{k-1},1}, \ldots, y_{2^{k-1},M-1}]^T, \tag{6}$$

$$\Psi(x) = [\psi_{1,0}(x), \psi_{1,1}(x), \ldots, \psi_{1,M-1}(x), \ldots, \psi_{2^{k-1},0}(x), \psi_{2^{k-1},1}(x), \ldots, \psi_{2^{k-1},M-1}(x)]^T. \tag{7}$$

Similarly, we can approximate the functions $k_i(x,\xi) \in L^2([0,1]\times[0,1])$ as

$$k_i(x,\xi) = \Psi(x)^T K_i \Psi(\xi), \tag{8}$$

where K_i, $i = 1, 2$ are $2^{k-1}M \times 2^{k-1}M$ matrices.

4 GQLWM for solving FVFIDEs

Let us consider the nonlinear FVFIDE with mixed boundary conditions given by (1)–(2). To approximate the functions $y(x)$ and $g(x)$ by $y(x) = Y^T\Psi(x)$ and $g(x) = G^T\Psi(x)$, assume that

$$N_1(y(\xi)) = u(\xi), \;\; N_2(y(\xi)) = v(\xi), \tag{9}$$

where $u(\xi)$ and $v(\xi)$ are as follows:

$$u(\xi) = U^T\Psi(\xi), \;\; v(\xi) = V^T\Psi(\xi), \tag{10}$$

in which U^T and V^T are defined similarly as in (6). Applying the operator ${}_0I_x^\alpha$ on both sides of (1), results in

$$\begin{aligned} y(x) - \sum_{k=0}^{n-1} y^{(k)}(0^+)\frac{x^k}{k!} =& \frac{1}{\Gamma(\alpha)}\int_0^x (x-\tau)^{\alpha-1} g(\tau)d\tau \\ &- \frac{1}{\Gamma(\alpha)}\int_0^x (x-\tau)^{\alpha-1}\int_0^\tau k_1(\tau,\xi)u(\xi)d\tau d\xi \\ &- \frac{1}{\Gamma(\alpha)}\int_0^x (x-\tau)^{\alpha-1}\int_0^1 k_2(\tau,\xi)v(\xi)d\tau d\xi. \end{aligned}$$

Replacing the exact solution $y(x)$ by $Y^T\Psi(x)$ and using their approximations by (5), (8)–(10), one gets

$$
\begin{aligned}
Y^T\Psi(x)-&\sum_{k=0}^{n-1} Y^T\Psi^{(k)}(0^+)\frac{x^k}{k!}\\
&=\frac{1}{\Gamma(\alpha)}\int_0^x (x-\tau)^{\alpha-1}G^T\Psi(\tau)d\tau\\
&\quad-\frac{1}{\Gamma(\alpha)}\int_0^x (x-\tau)^{\alpha-1}\int_0^\tau \Psi(\tau)^T K_1\Psi(\xi)U^T\Psi(\xi)d\tau d\xi\\
&\quad-\frac{1}{\Gamma(\alpha)}\int_0^x (x-\tau)^{\alpha-1}\int_0^1 \Psi(\tau)^T K_2\Psi(\xi)V^T\Psi(\xi)d\tau d\xi. \qquad (11)
\end{aligned}
$$

Collocating (11) in $M2^{k-1}$ Legendre wavelet-Gauss collocation points x_j, $j = 1, 2, \ldots, M2^{k-1}$, on interval $[0, 1]$, we arrive at the following system:

$$
\begin{aligned}
Y^T\Psi(x_j)-&\sum_{k=0}^{n-1} Y^T\Psi^{(k)}(0^+)\frac{x_j^k}{k!}\\
&=\frac{1}{\Gamma(\alpha)}\int_0^{x_j} (x_j-\tau)^{\alpha-1}G^T\Psi(\tau)d\tau\\
&\quad-\frac{1}{\Gamma(\alpha)}\int_0^{x_j} (x_j-\tau)^{\alpha-1}\int_0^\tau \Psi(\tau)^T K_1\Psi(\xi)U^T\Psi(\xi)d\tau d\xi\\
&\quad-\frac{1}{\Gamma(\alpha)}\int_0^{x_j} (x_j-\tau)^{\alpha-1}\int_0^1 \Psi(\tau)^T K_2\Psi(\xi)V^T\Psi(\xi)d\tau d\xi. \qquad (12)
\end{aligned}
$$

In a similar way, the mixed boundary conditions (2) are approximated as follows:

$$
\sum_{j=1}^{m}\Big[\lambda_{ij}Y^T\Psi^{(j-1)}(0)+\eta_{ij}Y^T\Psi^{(j-1)}(1)\Big]=\gamma_i,\quad i=1,2,\ldots,m. \qquad (13)
$$

Now, we use the quadrature rule to approximate the integral involved in this equation, which has zero error integration for polynomial integrands of degree less than or equal to $1 + M2^k$ with $M2^{k-1}$ Legendre-Gauss nodes. To do this, $M2^{k-1}$ intervals $[0, x_j]$ are transferred to a fixed interval $[-1, 1]$ through the transformations $\tau = (x_j/2\ )(s + 1)$. Applying the Gaussian quadrature, system (12) is converted to

$$
\begin{aligned}
Y^T\Psi(x_j)-&\sum_{k=0}^{n-1} Y^T\Psi^{(k)}(0^+)\frac{x_j^k}{k!}\\
&=\frac{1}{\Gamma(\alpha)}\int_0^{x_j} (x_j-\tau)^{\alpha-1}G^T\Psi(\tau)d\tau\\
&\quad-\frac{1}{\Gamma(\alpha)}\frac{x_j}{2}\sum_{l=1}^{M2^{k-1}} w_l\Big(\frac{x_j}{2}(1-s_l)\Big)^{\alpha-1}\int_0^\sigma [\Psi(\sigma)]^T K_1\Psi(\xi)U^T\Psi(\xi)d\xi\\
&\quad-\frac{1}{\Gamma(\alpha)}\frac{x_j}{2}\sum_{l=1}^{M2^{k-1}} w_l\Big(\frac{x_j}{2}(1-s_l)\Big)^{\alpha-1}\int_0^1 [\Psi(\sigma)]^T K_2\Psi(\xi)V^T\Psi(\xi)d\xi, \qquad (14)
\end{aligned}
$$

where $\sigma = \frac{x_j}{2}(s_l+1))$ and w_l for $l = 1, 2, \ldots, M2^{k-1}$ are the weight functions on $[-1, 1]$. Taking into account (9)–(10) and using collocation points x_j, $j = 1, 2, \ldots, M2^{k-1}$, one obtains

$$N_1(Y^T\Psi(x_j)) = U^T\Psi(x_j), \;\; N_2(Y^T\Psi(x_j)) = V^T\Psi(x_j). \tag{15}$$

Combining (13)–(15), the main problem is reduced to a system of nonlinear algebraic equations. By solving this system, the unknown values Y^T will be obtained, and by relation (5), the approximate solution of system (1) will be determined.

5 Convergence and error analysis

In this section, we will obtain an estimation for the error bound of our numerical method. Also, its convergence analysis will be discussed. For this purpose, assume that $L_m^*(x) = L_m(2x-1)$ and that $\Lambda(x) = [L_0^*(x), L_1^*(x), \ldots, L_N^*(x)]^T$. So, the function $y(x) \in L^2[0, 1]$ can be expressed in terms of the Legendre polynomials basis $\Lambda(x)$ as

$$y(x) = \sum_{n=0}^{N} l_n L_n^*(x) = L^T\Lambda(x),$$

where $L = [l_0, l_1, \ldots, l_N]^T$.

Lemma 2. Suppose that $y \in C^{m+1}[0, 1]$ and that $S = \text{span}\{L_0^*, L_1^*, \ldots, L_N^*\}$ is a vector space. If $L^T\Lambda(x)$ is the best approximation of $y(x)$ out of S, then the error bound of presented method is as follows:

$$\left\|y - L^T\Lambda\right\|_2 \leq \frac{\sqrt{2\chi^{2m+3}}M_{m+1}}{(m+1)!\sqrt{2m+3}},$$

in which $M_{m+1} = \max\left\{\left|f^{m+1}(x)\right| : \;\; 0 \leq x \leq 1\right\}$ and $\chi = \max\{1-\xi, \xi\}$.

Proof. The Taylor polynomials, implies that

$$Ty(x) = y_0(\xi) + (x-\xi)y'(\xi) + \frac{(x-\xi)^2}{2!}y''(\xi) + \cdots + \frac{(x-\xi)^m}{m!}y^{(m)}(\xi),$$

where $\xi \in (0, 1)$. Therefore, there exists $\lambda \in (0, 1)$ such that

$$|y(x) - Ty(x)| \leq \left|\frac{(x-\xi)^{m+1}}{(m+1)!}y^{(m+1)}(\lambda)\right|.$$

Since $L^T\Lambda(x)$ is the best approximation of $y(x)$, one obtains

$$\begin{aligned}\left\|y-L^T\Lambda\right\|_2^2 \le \|y-Ty\|_2^2 &= \int_0^1 |y(x)-Ty(x)|^2dx\\ &\le \int_0^1 \left|\frac{(x-\xi)^{m+1}}{(m+1)!}y^{(m+1)}(\lambda)\right|^2 dx\\ &\le \frac{M_{m+1}^2}{[(m+1)!]^2}\int_0^1 (x-\xi)^{2m+2}dx \le \frac{2\chi^{2m+3}M_{m+1}^2}{(2m+3)[(m+1)!]^2},\end{aligned}$$

and the proof is completed. □

Theorem 1. Assume that $y \in C^{m+1}[0,1]$ and that its approximation by the Legendre wavelets is $\tilde{y}(x) = Y^T\Psi(x)$. Then, its mean error bound is as follows:

$$\|y-\tilde{y}\|_2 \le \frac{2^{(m+1)(1-k)}\sqrt{2}M_{m+1}}{(m+1)!\sqrt{2m+3}}.$$

Proof. Dividing $[0,1]$ to 2^{k-1} subintervals $I_{k,n} = [(n-1)/2^{k-1}\ , n/2^{k-1}\ ]$, $n = 1,2,\ldots,2^{k-1}$ with the restriction that $\tilde{y}$ is a polynomial of degree less than $m+1$ and also using Lemma 2, we get

$$\begin{aligned}\|y-\tilde{y}\|_2^2 &\le \int_0^1 |y(x)-\tilde{y}(x)|^2dx\\ &= \sum_{n=1}^{2^{k-1}}\int_{\frac{n-1}{2^{k-1}}}^{\frac{n}{2^{k-1}}} |y(x)-\tilde{y}(x)|^2dx\\ &\le \sum_{n=1}^{2^{k-1}}\left[\frac{\sqrt{2}\bar{M}_n 2^{\frac{(1-k)(2m+3)}{2}}}{(m+1)!\sqrt{2m+3}}\right]^2 = \frac{2^{(1-k)(2m+3)+1}}{[(m+1)!]^2(2m+3)}\sum_{n=1}^{2^{k-1}}\bar{M}_n^2\\ &\le \frac{2^{(1-k)(2m+2)+1}M_{m+1}^2}{[(m+1)!]^2(2m+3)},\end{aligned}$$

where $\bar{M}_n = \max\left\{\left|y^{(m+1)}(x)\right|,\ \ x \in I_{k,n}\right\}$, which completes the proof. □

Lemma 3. Suppose that for $y(x) \in [0,1)$, there exists $\beta_y \in \mathbb{R}$ such that $|y(x)| \le \beta_y$. Then, the sum of Legendre coefficients of $y(x)$ defined in (5) is absolutely convergent if $|y_{n,m}| \le \sqrt{2^{1-k}}\beta_y$.

Proof. The function $y(x) \in L^2[0,1]$ can be expressed as the Legendre wavelet basis as defined in (5). For $m \ge 0$, we have

$$|y_{n,m}| = | < y, \psi_{n,m} > | = \left| \int_0^1 y(x)\psi_{n,m}(x)dx \right| \leq \int_0^1 |y(x)||\psi_{n,m}(x)|dx$$
$$\leq \beta_y \int_0^1 |\psi_{n,m}(x)|dx = \beta_y \int_{I_{n,k}} |\psi_{n,m}(x)|dx$$
$$= \beta_y 2^{\frac{k-1}{2}} (2m+1)^{\frac{1}{2}} \int_{I_{n,k}} |L_m(2^k x - 2n + 1)|\, dx.$$

Setting $s = 2^k x - 2n + 1$, one obtains

$$|y_{n,m}| \leq \beta_y 2^{\frac{-k-1}{2}} (2m+1)^{\frac{1}{2}} \int_{-1}^1 |L_m(s)|ds.$$

The Hölder's inequality implies that

$$|y_{n,m}| \leq \beta_y 2^{\frac{-k-1}{2}} (2m+1)^{\frac{1}{2}} \frac{2}{\sqrt{2m+1}} = \sqrt{2^{1-k}}\beta_y.$$

This means that the series $\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} y_{n,m}$ is absolutely convergent as $k \to \infty$. □

Theorem 2. The series solution $\tilde{y}(x) = \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} y_{n,m}\psi_{n,m}(x)$ defined by (5) is convergent with respect to L^2-norm on $[0,1]$ if the sum of absolute values of the Legendre coefficients $\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} |y_{nm}|$ for continuous function $y(x)$ forms a convergent series.

Proof. Let $L^2(\mathbb{R})$ be a Hilbert space and let $\psi_{n,m}$ defined in (3) form an orthogonal basis with $\psi_{n,m}(x) = \Delta_j(x)$ and $\delta_j =< \tilde{y}(x), \Delta_j(x) >$. Define a sequence of partial sums $\{S_n\}$ as

$$S_n(x) = \sum_{j=0}^{n} \delta_j \Delta_j(x).$$

Now, for every $\epsilon > 0$, there exists $N_\epsilon > 0$ such that for every $n > m > N_\epsilon$,

$$\|S_n(x) - S_m(x)\|_2^2 = \int_0^1 \left| \sum_{l=m+1}^{n} \delta_l \Delta_l(x) \right|^2 dx \leq \sum_{l=m+1}^{n} |\delta_l|^2 \int_0^1 |\Delta_l(x)|^2 dx$$
$$= \sum_{l=m+1}^{n} |\delta_l|^2.$$

Using Lemma 3, $\sum_{l=0}^{\infty} |\delta_l|^2$ is absolutely convergent. Hence, according to the Cauchy criterion, we have

$$\sum_{l=m+1}^{n} |\delta_l|^2 < \epsilon^2.$$

So, $\|S_n(x) - S_m(x)\|_2 \leq \sqrt{\epsilon^2} = \epsilon$. Thus, the sequence of partial sum of the series is a Cauchy sequence. Therefore, it convergent with respect to L^2-norm and this completes the proof. □

6 Existence and uniqueness

In order to study the uniqueness of the solution, we consider the FVFIDEs (1) in an operator form as

$$\left({}_0^C D_x^\alpha y\right)(x) = g(x) - K_1 N_1 y - K_2 N_2 y, \tag{16}$$

where

$$K_1 N_1 y = \int_a^x k_1(x,\xi) N_1(y(\xi))\, d\xi, \quad K_2 N_2 y = \int_a^b k_2(x,\xi) N_2(y(\xi))\, d\xi.$$

Applying the operator ${}_0I_x^\alpha$ on both sides of (16), one obtains

$$y(x) = f(x) + {}_0I_x^\alpha \left[g(x) - K_1 N_1 y - K_2 N_2 y\right],$$

where $f(x) = \sum_{k=0}^{n-1} y^{(k)}(0^+)x^k/k!$. This equation can be written as $\mathcal{T}y = y$, where $\mathcal{T}$ is as follows:

$$\mathcal{T}y(x) = f(x) + {}_0I_x^\alpha \left[g(x) - K_1 N_1 y - K_2 N_2 y\right].$$

Let $(C[0,1], ||\infty||)$ be a Banach space of all continuous functions with the norm $||h||_\infty = \max_x |h(x)|$. Also, the operators N_1 and N_2 satisfy the Lipschitz condition on $[0,1]$ with Lipschitz constants J_1 and J_2 as

$$|N_1\tilde{y}(x) - N_1 y(x)| \leq J_1 |\tilde{y}(x) - y(x)|, \quad |N_2\tilde{y}(x) - N_2 y(x)| \leq J_2 |\tilde{y}(x) - y(x)|.$$

In the following theorem with these assumptions, we show that the FVFIDEs (1) have a unique solution.

Theorem 3. Suppose that the nonlinear operators N_1 and N_2 satisfy the following relation

$$J_1 \|N_1\|_\infty + J_2 \|N_2\|_\infty < \Gamma(\alpha + 1).$$

Then the FVFIDE (1) has a unique solution.

Proof. Let $\mathcal{T} : C[0,1] \to C[0,1]$ be defined as

$$\mathcal{T}y(x) = f(x) + \frac{1}{\Gamma(\alpha)} \int_0^x (x-\xi)^{\alpha-1} \left[g(\xi) - K_1 N_1 y(\xi) - K_2 N_2 y(\xi) \right] d\xi.$$

Suppose that $\tilde{y}, y \in C[0,1]$. Then for every positive x, we have

$$\begin{aligned}
&\mathcal{T}\tilde{y}(x) - \mathcal{T}y(x) \\
&= \frac{1}{\Gamma(\alpha)} \int_0^x (x-\xi)^{\alpha-1} \left[K_1 N_1 y(\xi) - K_1 N_1 \tilde{y}(\xi) + K_2 N_2 y(\xi) - K_2 N_2 \tilde{y}(\xi) \right] d\xi.
\end{aligned}$$

Therefore, one obtain

$$\begin{aligned}
&|\mathcal{T}\tilde{y}(x) - \mathcal{T}y(x)| \\
&\leq \frac{1}{\Gamma(\alpha)} \int_0^x |x-\xi|^{\alpha-1} \left[|K_1| \, |N_1 y(\xi) - N_1 \tilde{y}(\xi)| + |K_2| \, |N_2 y(\xi) - N_2 \tilde{y}(\xi)| \right] d\xi \\
&\leq \frac{1}{\Gamma(\alpha)} \int_0^x |x-\xi|^{\alpha-1} \left[|K_1| \, J_1 \, |\tilde{y}(\xi) - y(\xi)| + |K_2| \, J_2 \, |\tilde{y}(\xi) - y(\xi)| \right] d\xi \\
&\leq \frac{1}{\Gamma(\alpha)} \int_0^x |x-\xi|^{\alpha-1} \left[\|K_1\|_\infty J_1 + \|K_2\|_\infty J_2 \right] \|\tilde{y}(\xi) - y(\xi)\|_\infty \, d\xi \\
&\leq \frac{|x|^\alpha}{\Gamma(\alpha+1)} \left(\|K_1\|_\infty J_1 + \|K_2\|_\infty J_2 \right) \|\tilde{y}(\xi) - y(\xi)\|_\infty \\
&\leq \frac{1}{\Gamma(\alpha+1)} \left(\|K_1\|_\infty J_1 + \|K_2\|_\infty J_2 \right) \|\tilde{y}(\xi) - y(\xi)\|_\infty.
\end{aligned}$$

Hence,

$$\|\mathcal{T}\tilde{y}(x) - \mathcal{T}y(x)\|_\infty \leq L \|\tilde{y}(\xi) - y(\xi)\|_\infty,$$

where $L = [1/\Gamma(\alpha+1)\,] \left(\|K_1\|_\infty J_1 + \|K_2\|_\infty J_2 \right)$. Since $L < 1$, by the contraction mapping theorem, this problem has a unique solution in $C[0,1]$. □

7 Numerical examples

In this section, to demonstrate the capability and accuracy of our method, we apply it to some examples and compare the quality of the computed solutions with some other well-known methods. In order to demonstrate the error, we will calculate the maximum absolute error (MAE) by the following formula:

$$MAE = \max_{0 \leq x < 1} \left\{ |y(x) - \tilde{y}(x)| \right\}.$$

Also for comparison, in Example 2, the root-mean-square error (RMSE) is calculated. This error is as follows:

$$\text{RMSE} = \sqrt{\int_0^1 \left[y(x) - \tilde{y}(x) \right]^2 dx}.$$

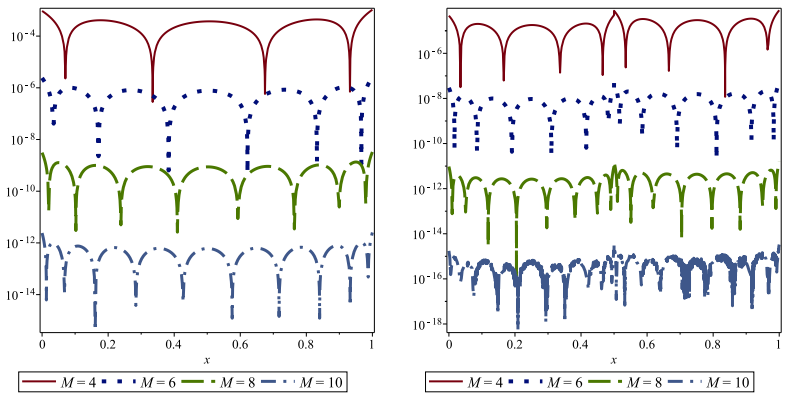
Example 1. Consider the following nonlinear FVFIDEs:

$$\left({}_0^C D_x^{\frac{\sqrt{7}}{2}} y\right)(x) - \int_0^x \frac{1+2\xi}{1+y(\xi)} d\xi - \int_0^1 (1+2\xi) y(\xi) d\xi = g(x),$$

subject to the boundary conditions $y(0) = 0$ and $y(1) = 2$ in which

$$g(x) = \frac{2x^{2-\frac{\sqrt{7}}{2}}}{\Gamma\left(3-\frac{\sqrt{7}}{2}\right)} - \ln(x^2+x+1) + 1 - e^2.$$

The exact solution of this problem is $y(x) = x^2 + x$. Numerical results for Example 1 are indicated in Table 1, which show that the results of our method are better than the results obtained in [24, 23] by the Chebyshev wavelet method (CWM), Nyström and Newton–Kantorovitch methods, respectively. Table 2 analyzes the exact solution y and the approximate solutions of this algorithm by CWM and GQLWM. One can see that the present method has excellent accuracy with respect to CWM. Table 3 shows the maximum absolute error between the exact and approximate solutions for various choices of M and k. The outcomes reveal that the results of GQLWM by using only a small number of bases are very promising and superior to CWM and Nyström and Newton–Kantorovitch methods. Furthermore, we compare the absolute error functions for different values of k and M in Figure 1. These results confirm that with increasing the amounts of k and M, the error will be decreased.



(a) $k = 1$ (b) $k = 2$

Figure 1: Error history of the presented method for various values for M and k of Example 1.

Table 1: Comparison between our method, CWM [23] and method in [24] for Example 1.

	x	0.1	0.3	0.5	0.7	0.9
Error [24]	$k=3,\ n=160$	$7.81E-06$	$1.55E-05$	$1.65E-05$	$1.27E-05$	$5.07E-06$
	$k=3,\ n=320$	$1.92E-06$	$3.84E-06$	$4.10E-06$	$3.15E-06$	$1.25E-06$
	$k=3,\ n=640$	$4.65E-07$	$9.31E-07$	$9.92E-07$	$7.65E-07$	$3.04E-07$
Error [23] for $k=1,\ M=19$		$3.47E-12$	$6.08E-12$	$6.21E-12$	$4.68E-12$	$1.84E-12$
GQLWM	$k=2,\quad M=4$	$2.93-06$	$1.01E-05$	$1.99E-06$	$1.08E-05$	$4.27E-06$
	$k=3,\quad M=5$	$8.17E-09$	$5.11E-10$	$5.05E-11$	$6.78E-10$	$9.48E-09$
	$k=4,\quad M=6$	$1.69E-23$	$4.22E-20$	$9.91E-21$	$7.06E-20$	$4.45E-23$
	$k=1,\ M=10$	$1.01E-12$	$4.67E-13$	$1.00E-12$	$4.71E-13$	$1.02E-12$
	$k=1,\ M=19$	$1.16E-28$	$2.11E-30$	$8.86E-29$	$3.35E-30$	$1.30E-28$

Table 2: The numerical solution of Example 1.

x	y(exact)	y(CWM)	y(GQLWM)
0.0	0.00	0.00000000000000000001	0.00000000000000000000
0.1	0.11	0.11000000000347509219	0.10999999999999999999
0.2	0.24	0.24000000000518991700	0.23999999999999999999
0.3	0.39	0.39000000000608906284	0.38999999999999999999
0.4	0.56	0.56000000000639391977	0.55999999999999999999
0.5	0.75	0.75000000000621724965	0.74999999999999999999
0.6	0.96	0.96000000000562994250	0.95999999999999999999
0.7	1.19	1.19000000000468149700	1.18999999999999999999
0.8	1.44	1.44000000000340880150	1.43999999999999999999
0.9	1.77	1.71000000000184060200	1.70999999999999999999
1	2.00	1.99999999999999999990	2.00000000000000000000

Table 3: Maximum absolute error between the exact and approximate solutions for various choices of M and k for Example 1.

	M	4	7	10	13	16	19
MAE	$k=1$	$4.41E-04$	$4.06E-08$	$1.03E-12$	$4.02E-16$	$3.01E-23$	$1.57E-28$
	$k=2$	$7.68E-05$	$9.41E-10$	$3.05E-15$	$3.96E-21$	$2.55E-27$	$1.61E-34$
	$k=3$	$5.46E-06$	$8.23E-12$	$3.39E-18$	$5.49E-25$	$4.42E-32$	$2.02E-39$

Example 2. Consider the following nonlinear fractional integro-differential equation:

$$\left({}_0^C D_x^{\alpha} y\right)(x) = g(x) + \int_0^1 x\xi[y(\xi)]^2 d\xi, \quad y(0) = 0,$$

where $0 < \alpha \leq 1$ and $g(x)$ is as follows:

Case 1. [23] $g(x) = \frac{4}{3}\Gamma\left(\frac{3}{4}\right)^{-1} x^{\frac{3}{4}} - \frac{1}{4}x$ with $\alpha = \frac{1}{4}$ and exact the solution is $y(x) = x$.

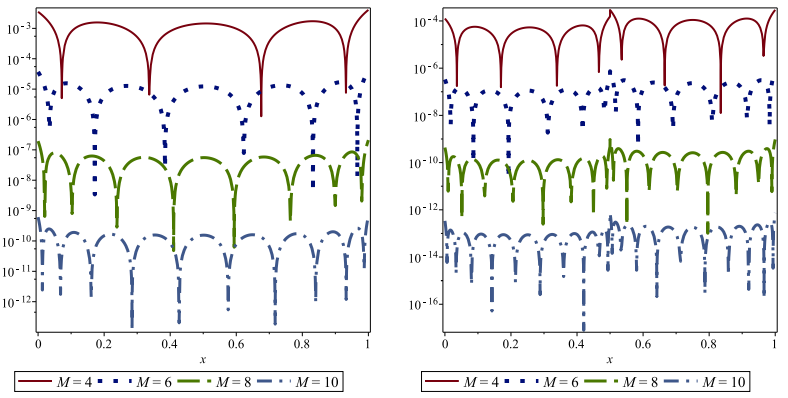
Case 2. [23] $g(x) = \frac{64}{15}\Gamma\left(\frac{3}{4}\right)\frac{\sqrt{2}}{\pi}x^{\frac{9}{4}} - \frac{1}{8}x$ with $\alpha = \frac{3}{4}$ and the exact solution is $y(x) = x^3$.

Case 3. [40] $g(x) = 1 - \frac{1}{4}x$ with $\alpha = 1$ and the exact solution is $y(x) = x$.

We apply our method to this example with various values of M and k, and the results are presented in Tables 4 and 5. The maximum absolute error of $y(x)$ by GQLWM for $k = 1$ with different values of M has been compared to the error of CWM [23] for cases 1 and 2. These results demonstrate the validity and capability of GQLWM with respect to CWM. The graphs of absolute error functions for different values of M and k with $\alpha = 1/4$, $\alpha = 1$, and $\alpha = 3/4$ are given in Figures 2, 3, and 4, respectively. These figures reveal that the error will be decreased when M is increased. Table 5 compares the RMSE of GQLWM with the results in [40] for different values of k and M when $\alpha = 1$. This table reveals that, for a certain value of k, as M increases, the accuracy increases. Also, for a certain value of M, as k increases, the accuracy increases as well. Therefore, GQLWM for solving this problem is very effective and more accurate with respect to CWM and the method in [40].

Table 5: RMSE for some k and M of GQLWM and the second kind Chebyshev wavelet method [40] for Example 2 (case 3).

	Method in [40]	GQLWM			
M	2	2	6	10	14
$k = 3$	$2.970E-07$	$2.348E-09$	$4.236E-11$	$1.003E-12$	$1.320E-13$
$k = 4$	$1.861E-08$	$7.391E-10$	$9.863E-13$	$4.002E-15$	$7.5691E-17$
$k = 5$	$1.164E-09$	$8.541E-12$	$8.786E-16$	$9.251E-18$	$2.015E-20$

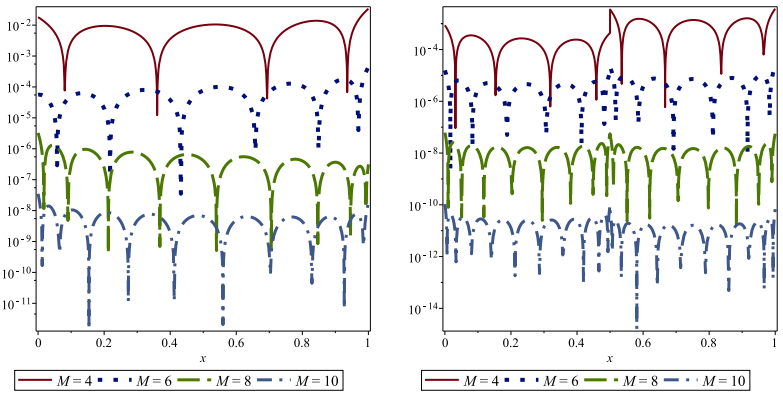


(a) $k = 1$ (b) $k = 2$

Figure 2: Error history of the presented method for various values for M and k of Example 2 with $\alpha = \frac{1}{4}$.

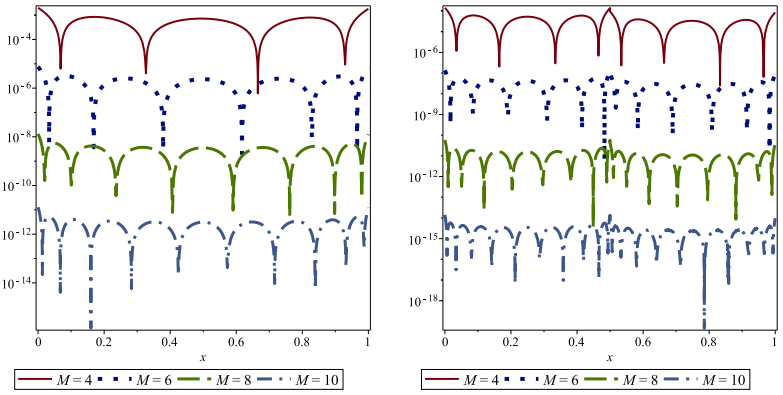
Table 4: Comparison between our method and CWM [23] for Example 2 (cases 1 and 2) when $k = 1$.

α	$\alpha = \frac{3}{4}$				$\alpha = \frac{1}{4}$			
	Error[23]	GQLWM			Error[23]	GQLWM		
x	$M = 19$	$M = 13$	$M = 16$	$M = 19$	$M = 19$	$M = 13$	$M = 16$	$M = 19$
0.0	$1.68E-20$	0	0	0	$7.92E-21$	0	0	0
0.1	$9.06E-13$	$1.75E-13$	$1.85E-16$	$1.27E-20$	$6.57E-18$	$6.53E-16$	$4.00E-19$	$2.51E-24$
0.2	$3.99E-12$	$4.05E-12$	$7.53E-17$	$3.92E-23$	$3.62E-18$	$5.89E-15$	$1.63E-19$	$2.91E-25$
0.3	$1.03E-11$	$3.76E-12$	$1.47E-16$	$1.49E-20$	$2.65E-18$	$5.44E-15$	$3.80E-19$	$6.18E-24$
0.4	$2.10E-11$	$2.21E-12$	$1.53E-16$	$1.06E-20$	$2.11E-18$	$3.41E-15$	$4.19E-19$	$4.08E-24$
0.5	$3.73E-11$	$8.35E-14$	$1.48E-16$	$1.83E-22$	$1.85E-18$	$4.07E-16$	$4.25E-19$	$2.29E-25$
0.6	$6.05E-11$	$2.38E-12$	$1.42E-16$	$1.04E-20$	$1.65E-18$	$2.61E-15$	$4.28E-19$	$4.33E-24$
0.7	$9.16E-11$	$3.92E-12$	$1.23E-16$	$1.51E-20$	$1.49E-18$	$4.63E-15$	$3.94E-19$	$5.92E-24$
0.8	$1.31E-10$	$4.23E-12$	$1.48E-16$	$3.71E-22$	$1.40E-18$	7	$1.81E-19$	$1.23E-25$
0.9	$1.82E-10$	$3.11E-13$	$1.67E-17$	$1.32E-20$	$1.07E-18$	$4.98E-15$	$5.10E-19$	$4.92E-24$
1	$2.45E-10$	0	0	0	$5.42E-17$	0	0	0



(a) $k = 1$ (b) $k = 2$

Figure 3: Error history of the presented method for various values for M and k of Example 2 with $\alpha = \frac{3}{4}$.



(a) $k = 1$ (b) $k = 2$

Figure 4: Error history of the presented method for various values for M and k of Example 2 with $\alpha = 1$.

8 Conclusion

In this paper, the GQLWM, which is an efficient technique for the numerical solution of nonlinear FVFIDEs, has been proposed. Using this method, the system of nonlinear FVFIDEs was reduced to a system of algebraic equations. The numerical solution of the resulted system was approximated by

the Gauss quadrature formula with respect to the Legendre weight function. By solving this nonlinear system, the numerical solution was obtained. Moreover, the convergence, error analysis, existence, and uniqueness of the proposed method were discussed. It was evident that, for a certain value of k, as M increases, the accuracy of the GQLWM is increased. Also, for a certain value of M, as k increases, the accuracy is increased, as well. The method was applied to two examples, and the obtained results were compared to some other well-known methods. This comparison showed that the GQLWM is a suitable and powerful technique for solving the nonlinear FVFIDEs. In the end, we note that the method can be easily extended and applied to multi-dimensional integral equations or systems of FVFIDEs easily with some modifications. We also believe that it shall not be difficult to extend this approach to nonlinear equations of general form, which will be the subject of future researches.

Acknowledgements

The author thanks the editor and the reviewers for their constructive comments and suggestions to improve the quality of the article.

References

1. Abbasbandy, S., Hashemi, M., Hashim, I.*On convergence of homotopy analysis method and its application to fractional integro-differential equations*, Quaest. Math. 36(1), (2013) 93–105.

2. Alkan, S., Hatipoglu, V.F.*Approximate solutions of Volterra-fredholm integro-differential equations of fractional order*, Tbilisi Math. Journal 10(2), (2017) 1–13.

3. Amin, R., Alshahrani, B., Mahmoud, M., Abdel-Aty, A.H., Shah, K., Deebani, W.*Haar wavelet method for solution of distributed order time-fractional differential equations*, Alex. Eng. J. 60(3), (2021) 3295–3303.

4. Amin, R., Shah, K., Asif, M., Khan, I.*A computational algorithm for the numerical solution of fractional order delay differential equations*, Appl. Math. Comput. 402, (2021) 125863.

5. Amin, R., Shah, K., Asif, M., Khan, I., Ullah, F.*An efficient algorithm for numerical solution of fractional integro-differential equations via Haar wavelet*, J. Comput. Appl. Math. 381, (2021) 113028.

6. Bhrawy, A., Zaky, M., Van Gorder, R.A.*A space-time Legendre spectral tau method for the two-sided space-time Caputo fractional diffusion-wave equation*, Numer. Algorithms 71(1) (2016), 151–180.

7. Erfanian, M., Gachpazan, M., Beiglo, H.*A new sequential approach for solving the integro-differential equation via Haar wavelet bases*, Comput. Math. Math. Phys. 57(2), (2017) 297–305.

8. Guner, O., Bekir, A.*Exp-function method for nonlinear fractional differential equations*, Nonlinear Sci. Lett. A 8, (2017) 41–49 .

9. Hamoud, A., Ghadle, K.*The reliable modified of Laplace Adomian decomposition method to solve nonlinear interval Volterra-Fredholm integral equations*, Korean J. Math. 25(3) (2017), 323–334.

10. Hashemi, M., Ashpazzadeh, E., Moharrami, M., Lakestani, M.*Fractional order Alpert multiwavelets for discretizing delay fractional differential equation of pantograph type*, Appl. Numer. Math. 170 (2021), 1–13.

11. Hashemi, M.S., Baleanu, D.*Lie symmetry analysis of fractional differential equations*, CRC Press (2020)

12. Hashemi, M.S., Darvishi, E., Inc, M.*A geometric numerical integration method for solving the Volterra integro-differential equations*, Int. J. Comput. Math. 95(8), (2018) 1654–1665.

13. He, S., Sun, K., Wang, H.*Dynamics of the fractional-order Lorenz system based on Adomian decomposition method and its DSP implementation*, IEEE/CAA Journal of Automatica Sinica (2016) doi: 10.1109/JAS.2016.7510133.

14. Heris, J.M.*Solving the integro-differential equations using the modified Laplace Adomian decomposition method*, J. Math. Ext. 6 (2012).

15. Hesameddini, E., Rahimi, A., Asadollahifard, E.*On the convergence of a new reliable algorithm for solving multi-order fractional differential equations*, Commun. Nonlinear Sci. Numer. Simul. 34, 154–164 (2016)

16. Hesameddini, E., Riahi, M.*Bernoulli Galerkin matrix method and its convergence analysis for solving system of Volterra-Fredholm integro-differential equations*, Iran. J. Sci. Technol. Trans. A Sci. 43(3) (2019), 1203–1214.

17. Hesameddini, E., Riahi, M., Latifizadeh, H.*A coupling method of homotopy technique and Laplace transform for nonlinear fractional differential equations*, International Journal of Advances in Applied Sciences 1(4), (2012) 159–170.

18. Hesameddini, E., Shahbazi, M.*Hybrid Bernstein block-pulse functions for solving system of fractional integro-differential equations*, Int. J. Comput. Math. 95(11) (2018), , 2287–2307.

19. Liu, Z., Cheng, A., Li, X.*A second-order finite difference scheme for quasilinear time fractional parabolic equation based on new fractional derivative*, Int. J. Comput. Math. 95(2) (2018), 396–411.

20. Mahdy, A.M., Mohamed, E.M.*Numerical studies for solving system of linear fractional integro-differential equations by using least squares method and shifted Chebyshev polynomials*, Fluid Mechanics: Open Access. (2016) 03. 10.4172/2476-2296.1000142.

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

22. Modanli, M., Akgül, A.*On solutions of fractional order telegraph partial differential equation by Crank-Nicholson finite difference method*, Appl. Math. Nonlinear Sci. 5(1), (2020) 163–170.

23. Mohyud-Din, S.T., Khan, H., Arif, M., Rafiq, M.*Chebyshev wavelet method to nonlinear fractional Volterra–Fredholm integro-differential equations with mixed boundary conditions*, Adv. Mech. Eng. 9(3), (2017) 1687814017694802.

24. Nazari S.D., Jahanshahi, M.*Numerical solution of nonlinear fractional Volterra-Fredholm integro-differential equations with mixed boundary conditions*, International Journal of Industrial Mathematics 7(1), (2015) 63–69.

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

26. Oldham, K., Spanier, J.*The fractional calculus theory and applications of differentiation and integration to arbitrary order*, vol. 111. Elsevier 1974.

27. Ordokhani, Y., Rahimi, N.*Numerical solution of fractional Volterra integro-differential equations via the rationalized Haar functions*, modern research physics 14(3), (2014) 211–2.

28. Pashayi, S., Hashemi, M.S., Shahmorad, S.*Analytical Lie group approach for solving fractional integro-differential equations*, Commun. Nonlinear Sci. Numer. Simul. 51, 66–77 (2017)

29. Pedas, A., Tamme, E., Vikerpuur, M.*Spline collocation for a class of nonlinear fractional boundary value problems*, AIP Conference Proceedings, vol. 1863, p. 160010. AIP Publishing, 2017.

30. Pirim, N.A., Ayaz, F.*A new technique for solving fractional order systems: Hermite collocation method*, Applied Mathematics 7(18), (2016) 2307.

31. Podlubny, I.*Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, vol. 198. Academic press, 1998.

32. Sahu, P., Ray, S.S.*A novel Legendre wavelet Petrov-Galerkin method for fractional Volterra integro-differential equations*, Comput. Math. with Appl. (2016).

33. Samko, S.G., Kilbas, A.A., Marichev, O.I. *Fractional integrals and derivatives*, Theory and Applications, Gordon and Breach, Yverdon 1993.

34. Shah, K., Khan, Z.A., Ali, A., Amin, R., Khan, H., Khan, A.*Haar wavelet collocation approach for the solution of fractional order Covid-19 model using Caputo derivative*, Alex. Eng. J. 59(5), (2020) 3221–3231.

35. Singh, B.K.*Homotopy perturbation new integral transform method for numeric study of space-and time-fractional* $(n+1)$*-dimensional heat-and wave-like equations*, Waves, Wavelets and Fractals 4(1) (2018)

36. Sun, H., Zhao, X., Sun, Z.z.*The temporal second order difference schemes based on the interpolation approximation for the time multiterm fractional wave equation*, J. Sci. Comput. 78(1), (2019) 467–498.

37. Sweilam, N., Nagy, A., Youssef, I.K., Mokhtar, M.M.*New spectral second kind Chebyshev wavelets scheme for solving systems of integro-differential equations*, Int. J. Appl. Comput. Math. 3(2), (2017) 333–345.

38. Wang, Y., Zhu, L.*Solving nonlinear Volterra integro-differential equations of fractional order by using euler wavelet method*, Adv. Differ. Equ. 2017(1), (2017) 27 pp.

39. Yin, X.B., Kumar, S., Kumar, D.*A modified homotopy analysis method for solution of fractional wave equations*, Adv. Mech. Eng. 7(12), (2015) 1687814015620330.

40. Zhu, L., Fan, Q. *Solving fractional nonlinear Fredholm integro-differential equations by the second kind Chebyshev wavelet*, Commun. Nonlinear Sci. Numer. Simul. 17(6), (2012) 2333–2341.

How to cite this article
Riahi Beni, M. Legendre wavelet method combined with the Gauss quadrature rule for numerical solution of fractional integro-differential equations. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(1): 229-249. doi: 10.22067/ijnao.2021.73189.1070

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] Stoer, J. and Bulirsch, R. *Introduction to Numerical Analysis*, Springer-verlag, New York, 2002.

[2] Brunner, H. *A survey of recent advances in the numerical treatment of Volterra integral and integro-differential equations*, J. Comput. Appl. Math. 8 (1982), 213-229.

Reviewers Acknowledgment

Peer review is the system used to assess the validity, quality and the originality of articles for publication in scholarly journals. We would like to sincerely thank the following reviewers who have taken part in the peer-review process for Iranian Journal of Numerical Analysis and Optimization during the last year.

Abbas, Muhammad
Afsharirad, Maria
Ahmad, Imtiaz
Ali, Mohamed
Alipour, Mohsen
Amini, Keyvan
Amini, Morteza
Amiraliyeva, Ilhame
Arabameri, Maryam
Area, Iván
Arezoomandan, Mahdie
Babaie-Kafaki, Saman
Baghani, Omid
Balakumar, V.
Biazar, Jafar
Burke, Maxim R.
Cakir, Musa
Chyzhykov, Igor
Cui, Jianbo
Dadi, Zohreh
Dadkhah, Mahmood
Darehmiraki, Majid
Darvishi, Mohammdtaghi
Duressa, Gemechis
Ebadi, Moosa
Elzaki, Tarig
Erfanian, Majid
Eslahchi, Mohammad Reza
Farahi, Mohammad Hadi
Fatemi, Masoud
Fazeli, Somayyeh
Gao, Jing
Ghanbari, Reza
Ghane, Fateme Helen
Ghiasi, mojtaba
Ghomanjani, Fatemeh
Ghorbani, Asghar
Ghorbani-Moghadam, Khatere
Ghovatmand, Mehdi
Gupta, Bhupendra
Hadhoud, Adel R.
Hadizadeh Yazdi, Mahmoud
Hamidi, Farhad
Hashemi, M.S.
Hashemi Mehne, Hamed
Hashemizadeh, Elham
Hassanpour, Hassan
Heydari, Mohammad H.
Hout, Karel
Hu, Gang
Iliev, Anton
Isah, Abdulnasir
Izadkhah, Mohammad Mahdi
Jabbari, Hadi
Jafarzadeh, Morteza
Jalilian, Reza
Karimi, S.
Khojasteh Salkuyeh, Davod
Kochetov, Yury
Kumar, Devendra
Mansoori, Amin
Marzban, Hamid Reza
Masjed-Jamei, Mohammad
Masnavi, Houman
Mesforush, Ali
Mirhosseini-Alizamini, Seyed Mehdi
Miri, Seyed Mohsen
Mohammadi, Maryam
Namjoo, Mehran
Nasseri, Hadi
Nazemi, Alireza
Noori Skandari, M. Hadi
Panjeh Ali Beik, Fatemeh
Podila, Pramod Chakravarthy
Ranjbar, Mahdi
Rashidinia, Jalil

Ravi Kanth, Adivi Sri Venkata
Rezaee Pajand, Mohammad
Rivaz, Azim
Roul, Pradip
Saberi-Movahed, Farid
Saboor Bagherzadeh, Amir
Sadeghi Goughery, Hossein
Sahihi, Hussein
Salahi, Maziar
Samei, Mohammad Esmael
Sarmad, majid
Seikh, Mijanur
Shah, Kamal
Shahriari, Mohammad
Shaikh, Ali Akbar
Shirazian, Mohammad
Shirzadi, Mohammad
Shivanian, Elyas
Solaymani Fard, Omid
Soleimani Damaneh, Majid
Soleymani, Fazlollah
Soroushian, Aram
Soujanya, BSL
Tajaddini, Azita
Tohidi, Emran
Toutounian, Faezeh
Tynda, Moslem
Vasundhara, J. Devi
Wang, Hongwei
Yazdani Cherati, Allah Bakhsh
Zaferanieh, Mehdi
Zaky, Mahmoud A.
Zamani, Moslem
Zarrabi, Mohammad Reza
Zeng, Tieyong
Zhang, Huayan

Iranian Journal of Numerical Analysis and Optimization

Former : MRJMS

CONTENTS Vol. 12, No.1, pp 1-250, 2022

The rest of contents is on back of the page

web site : http://ijnao.um.ac.ir

Email : ijnao@um.ac.ir

ISSN-Print : 2423-6977

ISSN-Online : 2423-6969