Iranian Journal of Numerical Analysis and Optimization

Volume 9 , Number 2

Summer 2019

Serial Number: 16

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, by the Ministry of Culture and Islamic Guidance.

Volume 9, Number 2, Summer 2019

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:

- Zentralblatt
- ISC
- SID
- DOAJ

Informs all scholars, researchers, professors, postgraduate students and respected authors that the Iranian Journal of Numerical Analysis and Optimization- IJNAO - licensed on /3/18/548913 dated 15/1/2014 by the Honorable Director General of Policy Research and Planning, Ministry of Science, Research and Technology is a scientific journal. According to the letter No. 92/1536 dated 10/2/2014, the respected supervisor of the Vice President of Research and Technology of the Islamic world Science Citation database, the Iranian Journal of Numerical Analysis and Optimization is also indexed on the ISC database.

Iranian Journal of Numerical Analysis and Optimization

Volume 9, Number 2, Summer 2019

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 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 biannually 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. 9, No. 2, (2019), pp 1–16
DOI:10.22067/ijnao.v9i2.70454

Research Article

A new approximate inverse preconditioner based on the Vaidya's maximum spanning tree for matrix equation $AXB = C$

K. Rezaei, F. Rahbarnia* and F. Toutounian

Abstract

We propose a new preconditioned global conjugate gradient (PGL-CG) method for the solution of matrix equation $AXB = C$, where A and B are sparse Stieltjes matrices. The preconditioner is based on the support graph preconditioners. By using Vaidya's maximum spanning tree preconditioner and BFS algorithm, we present a new algorithm for computing the approximate inverse preconditioners for matrices A and B and constructing a preconditioner for the matrix equation $AXB = C$. This preconditioner does not require solving any linear systems and is highly parallelizable. Numerical experiments are given to show the efficiency of the new algorithm on CPU and GPU for the solution of large sparse matrix equation.

AMS(2010): 65F10.

Keywords: Krylov subspace methods; matrix equation; approximate inverse preconditioner; global conjugate gradient; support graph preconditioner; Vaidya's maximum spanning tree preconditioner.

*Corresponding author
Received 23 January 2018; revised 27 May 2018; accepted 13 August 2018
K. Rezaei
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran e-mail: k.rezai1985@gmail.com

F. Rahbarnia
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran. e-mail: Rahbarnia@ferdowsi.um.ac.ir

F. Toutounian
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran.
The Center of Excellence on Modeling and Control Systems, Ferdowsi University of Mashhad, Iran. e-mail: toutouni@math.um.ac.ir

1 Introduction

The solution of linear systems of equations is at the heart of many computations in science, engineering, and other disciplines; see [2, 8–10] and their references. Hence, many researches have been performed on various types of matrix equations; for example, see [1, 8, 11, 16, 18, 19, 27, 28].

The principal goal of this paper is to use support graph preconditioning techniques to solve the matrix equation

$$AXB = C, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ are large sparse Stieltjes matrices. The linear matrix equation (1) can be written as the following $nm \times nm$ linear system:

$$(B^T \otimes A)vec(X) = vec(C), \tag{2}$$

where $vec(X)$ is the vector of $\mathbb{R}^{nm}$ obtained by stacking the columns of the $n \times m$ matrix X and $\otimes$ denotes the Kronecker product ($A \otimes B = [a_{ij}B]_{ij}$). The CG algorithm [24] can be used to solve the linear system (2). However, for large problems, this approach cannot be applied directly. In addition, the number of iterations of conjugate gradient method for the solution of linear system of equations $Ax = b$ is bounded by the square root of the spectral condition number $\kappa(A)$ of A. The condition number is the ratio of extreme eigenvalues of A, $\kappa(A) = \lambda_{max}(A)/\lambda_{min}(A)$. Preconditioner accelerates the convergence of iterative methods for solving linear systems.

In this paper, we use the preconditioned global conjugate gradient (PGL-CG) method for obtaining the approximate solution of matrix equation (1). The preconditioner is based on the support graph preconditioners. Predecessors of support-graph methods can be found in the work from the late 80s by Notay, Beauwens, and collaborators in which graph-theoretic notions (principally paths) are used in the analysis of preconditioners; see [3,4,21–23]. These insights were extended by Vaydia [26], who described his work in a talk in 1991 but did not publish a paper. Vaidya used support-graph techniques to design a family of preconditioners based on spanning trees in graphs. Later, Gremban, Miller, and Zagha [14, 15] extended the technique and used it to construct another family of preconditioners. In Section 3, we use Vaidya's maximum spanning tree preconditioners of matrices A and B for developing fast and efficient preconditioner to precondition equation (2).

Throughout this paper, all matrices are assumed to be real. For two matrices $X, Y \in \mathbb{R}^{n\times s}$, the inner product $\langle X, Y\rangle_F = (Y^T X)$ is used and the associated norm is the Frobenius norm denoted by $\|.\|_F$.

The rest of the paper is organized as follows. In the next section, we implement the preconditioned global CG method for solving matrix equation (1) and we introduce Vaidya's maximum spanning tree preconditioner. In section 3, we present a new algorithm for computing the inverse of this kind

of preconditioners. In section 4, numerical examples are given to illustrate the efficiency of the proposed preconditioner. Conclusions are summarized in Section 5.

2 Preconditioned GL-CG method for solving the matrix equation $AXB = C$

In this section, we consider the matrix equation $AXB = C$, where A and B are symmetric and positive definite and assume that the preconditioners P_A and P_B are available. The preconditioners P_A and P_B are the matrices that approximate A and B in some sense, respectively. It is assumed that P_A and P_B are also symmetric positive definite. Then, we can precondition system (1) as follows:

$$(P_B \otimes P_A)^{-1}(B \otimes A)vec(X) = (P_B \otimes P_A)^{-1}vec(C), \tag{3}$$

where the preconditioner $(P_B \otimes P_A)$ is a symmetric positive definite matrix. In addition, from the fact that $\|A \otimes B\| = \|A\|\|B\|$ [17], we have

$$\text{cond}((P_B \otimes P_A)^{-1}(B \otimes A)) = \text{cond}(P_B^{-1}B)\text{cond}(P_A^{-1}A).$$

The straightforward application of PCG algorithm [24] to the linear system (3) yields the following preconditioned global CG algorithm for solving the matrix equation (1).

Algorithm 1 PGL-CG for solving AXB=C

1. Compute $R_0 = C - AX_0B, Z_0 = P_A^{-1}R_0P_B^{-1}$ and $P_0 = Z_0$
2. **for** $j = 0, 1, \ldots,$ until convergence **do**
3. $\quad \alpha_j = \frac{<R_j, Z_j>_F}{<AP_jB, P_j>_F}$
4. $\quad X_{j+1} = X_j + \alpha_j P_j$
5. $\quad R_{j+1} = R_j - \alpha_j AP_jB$
6. $\quad Z_{j+1} = P_A^{-1}R_{j+1}P_B^{-1}$
7. $\quad \beta_j = \frac{<R_{j+1}, Z_{j+1}>_F}{<R_j, Z_j>_F}$
8. $\quad P_{j+1} = Z_{j+1} + \beta_j P_j$
9. **end for**

We focus on applying Vaidya's preconditioner of the first class to the matrices A and B for constructing the preconditioners P_A and P_B. In order to explain Vaidya's preconditioner, we first present the following definition from [7].

Definition 1. The underlying graph $G_A = (V_A, E_A)$ of an n-by-n symmetric matrix A is a weighted undirected graph whose vertex set is $V_A = \{1, 2, \ldots, n\}$

and whose edges set is $E_A = \{(i,j) : i \neq j, a_{i,j} \neq 0\}$. The weight of an edge (i,j) is $-a_{i,j}$. The weight of a vertex i is the sum of elements in the row i of A.

Graph preconditioner, introduced by Vaidya [26] in the early nineties, uses maximum-weight spanning tree (MWST) preconditioners to bound the condition number of a preconditioned system. Vaidya's method constructs a preconditioner M whose underlying graph G_M is a subgraph of G_A (graph of A). The graph G_M of preconditioner has the same set of vertices as G_A and a subset of the edges of G_A. Vaidya proposed two classes of preconditioners. The first class of MWST preconditioners guarantees a condition number bound of $O(n^2)$ for any $n \times n$ sparse diagonally dominant symmetric (SDD) matrix; see [7]. The second class of preconditioners is based on MWST augmented with a few extra edges. This class of preconditioners guarantees that the work in the linear solver is bounded by $O(n^{1.75})$ for any sparse diagonally dominant matrix. In this paper, we focus on applying Vaidya's preconditioners of the first class to a subclass of SDD matrices, the class of SDD matrices with nonpositive off-diagonal elements (Stieltjes matrices).

In order to construct the MWST preconditioner P_A for A, we first construct the maximum-weight spanning tree T_A in G_A and then modify the diagonal elements of preconditioner P_A such that A and P_A have the same row sums. In other words, T_A is a connected graph with no cycles (i.e., a spanning tree), and the total weight of its edges is maximal among all spanning trees of G_A. The preconditioner P_A is a diagonally dominant Stieltjes matrix whose underlying graph is $G_{P_A} = T_A$, and whose row sums are identical to those of A.

When the condition number of the matrix $B \otimes A$ is high, it becomes necessary to develop a fast and efficient preconditioner for the iterative solution of (2). In order to precondition the system (2), we first construct Vaidya's preconditioners (maximum-weight spanning tree) P_A and P_B for A and B, respectively, and then we use $P_B \otimes P_A$ as a preconditioner for the matrix $B \otimes A$. The implementation of this preconditioner is based on computation of the inverse matrices P_A^{-1} and P_B^{-1}. In Section 3, we show that, by using the breadth first search (BFS) algorithm [25], we can easily compute these inverse matrices.

3 Computation of inverse of a MWST preconditioner

Let M be a symmetric positive definite matrix whose underlying graph T_M is a tree. In order to compute the inverse of M, we need the following definition.

Definition 2. The elimination matrix $L_{pq}(-\alpha) \in \mathbb{R}^{n\times n}$ with $p \neq q$, is an identity matrix with one nonzero off-diagonal entry in the row p and the column q. Therefore the entries of $L_{pq}(-\alpha)$ are as follows:

$$(L_{pq}(-\alpha))_{(i,j)} = \begin{cases} 1 & \text{if } i = j, \\ -\alpha & \text{if } (i,j) = (p,q), \\ 0 & otherwise. \end{cases}$$

Now we investigate the result of symmetric transformation

$$\overline{M} = L_{pq}(-\alpha) M L_{pq}^T(-\alpha). \tag{4}$$

Now $L_{pq}(-\alpha)M$ changes only the row p of M, while $ML_{pq}^T(-\alpha)$ changes only the column p. Thus, multiplying out equation (4) and using the symmetry of M, we get the explicit formulas

$$\begin{array}{ll} \bar{m}_{pj} = \bar{m}_{jp} = m_{pj} - \alpha m_{qj}, & j \neq p, \\ \bar{m}_{pp} = m_{pp} - 2\alpha m_{pq} + \alpha^2 m_{qq}, & \\ \bar{m}_{ij} = \bar{m}_{ji} = m_{ij} & \text{otherwise.} \end{array} \tag{5}$$

The idea of our method is to try to zero the off-diagonal elements of M by a series of transformations (4) and using the leaves of graph T_M and its subtrees.

Let us assume that the vertex q is a leaf in T_M and its neighbor is the vertex p. In order to zero the off-diagonal m_{pq}, accordingly, to set $\bar{m}_{pq} = 0$, equation (5) gives the following expression for the parameter α :

$$\alpha = \frac{m_{pq}}{m_{qq}}. \tag{6}$$

From (5) and (6), the entries of $\overline{M} = L_{pq}(-\alpha)ML_{pq}^T(-\alpha)$ are as follows:

$$\begin{array}{ll} \bar{m}_{pp} = m_{pp} - 2\alpha m_{pq} + \alpha^2 m_{qq}, & \\ \bar{m}_{pq} = \bar{m}_{qp} = 0, & \\ \bar{m}_{ij} = m_{ij} & \text{otherwise.} \end{array}$$

This process which eliminates the nonzero entries m_{pq} and m_{qp} of M, is equivalent to eliminate the edge (p,q) from the tree T_M. If $\widetilde{M}$ denotes the matrix obtained from the matrix $\overline{M}$ by removing the row q and the column q, then it is trivial that the underlying graph of this submatrix is the induced subtree of T_M on $V(T_M) - \{q\}$. Let $M_1 = M, p_1 = p, q_1 = q, \alpha_1 = \alpha, \overline{M}_1 = L_{pq}(-\alpha)ML_{pq}(-\alpha)^T$, and $M_2 = \widetilde{M}$; then, we can successively transform M to diagonal form by means of transformations of the type (4) in $(n-1)$ steps with the elimination matrices $L_{p_j,q_j}(-\alpha_j)$ and $\alpha_j = (m_{p_j,q_j}/m_{q_j,q_j}), j = 1,2,\ldots,n-1$, which are defined by choosing the edges $(p_j,q_j), j = 1,2,\ldots,n-1$ such that the vertex q_j is a leaf in the subtree T_{M_j}. To achieve this, we need to apply the BFS algorithm (Algorithm 2) to the maximum-weight spanning tree T_M to obtain the vector $V = [j_1, j_2, \ldots, j_n]$, which represents an array of vertices that are traversed

and sorted by the BFS algorithm and $Level(u_j), j = 1, 2, \ldots, n$, which represent the level of traversed vertices $u_j, j = 1, 2, \ldots, n$ in the BFS tree.

By using the array of vertices $V = [j_1, j_2, \ldots, j_n]$, we can diagonalize the matrix M in $n-1$ steps. In step k, $k = 1, 2, \ldots, n-1$, by choosing the vertex $q_k = j_{n-k+1}$ from V and considering its parent $p_k = i_{n-k+1}$ and the edge (p_k, q_k), we define the elimination matrix $L_{p_k,q_k}(-\alpha_k)$ for eliminating the off-diagonal element m_{p_k,q_k}. In Lemma 1, we show that the vertex $q_k = j_{n-k+1}$ at step k is a leaf in the subtree T_{M_k}. In what follows, we show that by using Level (u_j), $j = 1, 2, \ldots, n$, we can reduce the overall time of producing the inverse of M.

Algorithm 2 Breadth first search algorithm as BFS(G,s)

$V = \varnothing$
for each vertex $u \in V(G) - s$ **do**
  $state(u) =$ "$undiscovered$"
  $p(u) = nil$, i.e. no parent is in the BFS tree
end for
$state(s) =$ "$discovered$"
$V = V \cup \{s\}$
$Level(s) = 0$
$p(s) = nil$
$Q = \{s\}$
while $Q \neq \varnothing$ **do**
  $u = dequeue(Q)$
  process vertex u as desired
  for each $v \in Adj(u)$ **do**
    process edge (u, v) as desired
    if $state(v) =$ "$undiscovered$" **then**
      $state(v) =$ "$discovered$"
      $V = V \cup \{v\}$
      $p(v) = u$
      $Level(v) = Level(p(v)) + 1$
      $enqueue(Q, v)$
    end if
    $state(u) =$ "$processed$"
  end for
end while

Lemma 1. *Let $V = \{j_1, j_2, \ldots, j_n\}$ be the set of vertices obtained by the BFS algorithm. If we isolate the vertex j_n and $T_M^{(j_n)}$ denotes the induced subtree of T_M on the vertex set $V(T_M) - \{j_n\}$, then j_{n-1} is a leaf in the induced subtree $T_M^{(j_n)}$.*

Proof. Let i_n be the parent of j_n and $Level(j_n) = l$. According to the BFS algorithm, if $s < t$, then $Level(j_s) \leqslant Level(j_t)$. Suppose that we isolate the

vertex j_n; then we must consider the $Level(j_{n-1})$. If $Level(j_{n-1}) = l$, then it is trivial that the vertex j_{n-1} is a leaf in the induced subtree $T_M^{(j_n)}$. If $Level(j_{n-1}) = l - 1$, then it means that there is no vertex in level l, so the vertex j_{n-1} has no children, according to the BFS algorithm, and it is a leaf in the subtree $T_M^{(j_n)}$. □

Let M be Vaidya's maximum-weight spanning tree preconditioner for the diagonally dominant spd matrix A and let $V = \{j_1, j_2, \ldots, j_n\}$ be the array obtained by the BFS algorithm. We observe that we can diagonalize M by $n-1$ elimination matrices $L_{p_k,q_k}(-\alpha_k)$, $k = 1, 2, \ldots, n-1$, where $q_k = j_{n-k+1}$. So, the elimination process yields the diagonal matrix D as follows:

$$D = L_{n-1}L_{n-2}\ldots L_1 M L_1^T \ldots L_{n-2}^T L_{n-1}^T,$$

where $L_k = L_{p_k,q_k}(-\alpha_k), k = 1, 2, \ldots, n-1$. Therefore, we have

$$M^{-1} = (L_{n-1}L_{n-2}\ldots L_1)^T D^{-1}(L_{n-1}L_{n-2}\ldots L_1).$$

In addition, by supposing that level$(j_n) = l$, we can write

$$M^{-1} = (G_1 G_2 \ldots G_l)^T D^{-1}(G_1 G_2 \ldots G_l),$$

where $G_k = L_{\nu_k+s_k-1}\ldots L_{\nu_k}$ for $k = 1, \ldots, l$, and s_k represents the number of vertices that have the level k in the graph T_M, and the matrices $L_{\nu_k+s_k-1}, \ldots, L_{\nu_k}$ are generated by the vertices $j_{n-(\nu_k+s_k-1)+1}, \ldots, j_{n-\nu_k+1}$, which have level k. In Lemma 2, we show that the nonzero off-diagonal elements of G_k are equal to the nonzero off-diagonal elements of matrices $L_{\nu_k+s_k-1}, \ldots, L_{\nu_k}$.

Lemma 2. *Let $S_k = \{j_{n-(\nu_k+s_k-1)+1}, \ldots, j_{n-\nu_k+1}\}$ be the set of vertices in level k of algorithm BFS and let $G_k = L_{\nu_k+s_k-1}\ldots L_{\nu_k}$, where $L_r = L_{p_r,q_r}(-\alpha_r)$ for $r = \nu_k, \ldots, \nu_{k+s_k-1}$ and p_r is the parent of $q_r = j_{n-r+1}$. Then the entries of G_k are as follows:*

$$G_k(i,j) = \begin{cases} 1 & \text{if } i = j, \\ -\alpha_r = -\frac{m_{p_r,q_r}}{m_{q_r,q_r}} & \text{if } (i,j) = (p_r, q_r),\ q_r = j_{n-r+1} \in S_k, \\ & \quad \text{and } (p_r, q_r) \in E(T_A), \\ 0 & otherwise. \end{cases}$$

Proof. From the definition of elimination matrix $L_r = L_{p_r,q_r}(-\alpha_r)$, we have

$$L_r = L_{p_r,q_r}(-\alpha_r) = I - \alpha_r E_{p_r,q_r},$$

where E_{p_r,q_r} contains only 0s except for 1 in the (p_r, q_r)th position. From the fact that all the vertices in S_k have level k, for all $q_r(= j_{n-r+1}), q_{r'}(= j_{n-r'+1}) \in S_k$, we have

$$E_{p_r,q_r} \times E_{p_{r'},q_{r'}} = 0 \quad \text{for } q_r \neq p_{r'}.$$

So, by the induction on the number of elimination vertices in level k, we can easily show that

$$G_k = I - \sum_{r=\nu_k}^{\nu_k+s_k-1} \alpha_r E_{p_r,q_r},$$

which completes the proof. □

Finally, summarizing the previous results, we describe the tree inverse algorithm for computing the inverse of M as follows:

Algorithm 3 Tree inverse

$input(T_A, root)$
$(v, Level) = BFS(T_A, root)$
$\tilde{M} = I$
$d = Diag(A)$
for $k = l$ to 1 step -1 (l is the number of levels obtained from the BFS algorithm) **do**
  $G = I$
  for all vertices in level k **do**
    j =current vertex
    i =the parent of current vertex
    $\alpha = -\frac{m_{ij}}{m_{jj}}$
    $g_{ij} = \alpha$
    $d_{ii} = d_{ii} - \alpha a_{ij}$
  end for
  $\tilde{M} = G\tilde{M}$
end for
set $P_A^{-1} = \tilde{M}^T D^{-1} \tilde{M}$

4 Numerical experiments

In this section, we compare the experimental results obtained by solving the preconditioned system of equation (1). Four preconditioners will be compared: MWST, AINV (right-looking version) [5], incomplete Cholesky, and RIF [6] preconditioners. In addition, the following approaches are used for applying the MWST preconditioners:

1. We use the matrices P_A^{-1} and P_B^{-1} computed by the tree inverse algorithm (Algorithm 3).

2. We use the Cholesky factorization of the MWST preconditioners P_A and P_B for computing Z_{j+1} in lines 1 and 6 of Algorithm 1.

3. In order to reduce the fill-in, first, we apply the reverse Cuthill–McKee ordering [12,13] to the preconditioners P_A and P_B and then we use the Cholesky factorizations of the resulting matrices.

Finally, for large matrices, we compare the results obtained by the approach 1 on GPU and CPU.

The examples have been coded in MATLAB with double-precision and have been executed on a quad-processor 4.2 GHz i7 computer with 32 GBytes of main memory. In all examples, the initial iteration matrix is zero. We stop the iterations when

$$RError = \frac{\|R_k\|_F}{\|R_0\|_F} \leq \epsilon,$$

where R_k is the residual of the kth iterate and ϵ is a proper stopping tolerance. In all the tables, the CPU time is in second and a dagger (†) indicates that no convergence is achieved after 10000 iterations except for Tables 5 and 6, where the maximum number of iterations is 30000. We also set the stopping tolerance 10^{-9}. The matrix C is chosen such that the exact solution X has the entries $x_{ij} = i * j$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

For the first set of examples, we consider the matrix NOS6 from Harwell–Boeing collection [20] and the matrix ST_n, which is obtained by discretizing the poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \quad in\ \Omega =]0,1[\times]0,1[$$

with the Dirichlet boundary condition on a uniform grid of mesh size $h = \frac{1}{n+1}$ via central differences. These matrices with their properties are presented in Table 1. In this table, cond denotes the condition number of the matrices in 2-norm.

Table 1: First set of test problems information

Test matrix	n	nnz	cond
ST_5	25	105	20.77
ST_{10}	100	460	69.8634
ST_{20}	400	1920	258.4520
ST_{30}	900	4380	564.9227
ST_{40}	1600	7840	989.2690
ST_{50}	2500	12300	1531.5
Nos6	675	3255	8×10^6

In Table 2, we compare the number of iterations (It) and the CPU iteration time (It-time) for the preconditioners: the approximate inverse with drop tolerance equal to 0.1 (AINV) [5] , the incomplete Cholesky factorization

Table 2: Number of iterations and CPU times to converge for the first set of examples

Matrices		$MWST_1$		$MWST_2$		$MWST_3$		AINV		IC(0)		RIF	
A	B	It	It-time	It	It-time	It	It-time	It	It-time	It	It-time	It	It-time
NOS6	ST_5	207	0.16	196	0.20	194	0.11	431	0.24	273	0.20	199	0.29
NOS6	ST_{10}	499	0.97	485	1.86	486	1.63	1110	2.03	575	1.71	405	1.59
NOS6	ST_{20}	1131	16.78	1115	35.49	1115	20.93	2738	34.58	1353	24.61	962	21.27
NOS6	ST_{30}	1817	84.99	1797	162.92	1797	100.26	4936	235.54	2285	136.82	1591	105.34
NOS6	ST_{40}	2489	308.60	2470	512.05	2470	335.12	7771	978.51	3381	463.15	2396	385.51
NOS6	ST_{50}	3167	884.18	3152	1283.7	3148	873.3	†	†	4570	1242.9	3332	1094.4
NOS6	NOS6	725	22.90	694	35.25	694	23.12	2016	38.51	1737	62.65	766	35.05

Table 3: Preconditioning times and total times for the first set of examples

Matrices		$MWST_1$		$MWST_2$		$MWST_3$		AINV		IC(0)		RIF	
A	B	P-time	T-time	P-time	T-time	P-time	T-time	P-time	T-time	P-time	T-time	P-time	T-time
NOS6	ST_5	0.45	0.61	4.61	4.81	0.85	0.96	0.40	0.64	1.08	1.28	0.27	0.56
NOS6	ST_{10}	0.49	1.46	4.70	6.56	0.87	2.50	0.40	2.43	1.11	2.82	0.29	1.82
NOS6	ST_{20}	0.65	17.43	7.25	42.74	1.10	22.03	0.48	35.06	1.44	26.05	0.39	21.66
NOS6	ST_{30}	0.92	85.91	23.91	188.83	2.10	102.36	1.16	236.7	2.97	139.79	0.89	106.23
NOS6	ST_{40}	1.34	309.94	85.63	598.13	4.88	340.00	4.89	983.49	6.92	470.07	3.53	389.04
NOS6	ST_{50}	2.15	886.33	248.50	1532.20	10.35	883.65	17.02	†	15.25	1258.15	10.84	1105.24
NOS6	NOS6	0.80	23.70	9.19	44.44	1.63	24.75	0.78	59.29	2.14	64.79	0.56	35.61

(IC(0)), MWST using the approaches 1–3 ($MWST_1$, $MWST_2$, $MWST_3$, respectively), and the robust incomplete factorization with drop tolerance equal to 0.1 (RIF). The CPU time for computing the preconditioner (P-time) and the total time for computing an approximate solution (T-time) are given in Table 3. Table 2 reveals that the preconditioner $MWST_1$ is faster (in terms It-time) than the other preconditioners (except for $MWST_3$ with A =NOS6 and $B = ST_5, ST_{50}$) and it requires a lower number of iterations than AINV and IC(0) preconditioners. Table 3 shows that the preconditioners $MWST_1$ is faster (in terms T-time) than the other preconditioners (except for $MWST_3$ with A =nos6 and $B = ST_{50}$, and RIF for NOS6 and ST_5). In addition, for large matrices (NOS6 with ST_{40}, and ST_{50}), $MWST_1$ preconditioner is better (in terms of P-time) than the other preconditioners. For small matrices (NOS6 with ST_5, ST_{10}, ST_{20}, and ST_{30}), we observe that the time of constructing the preconditioner RIF is smaller than that of the other preconditioners. For the second set of examples, we define matrices $STM_n = ST_n + DI_n$, where DI_n is a diagonal matrix such that the matrix STM_n has zero row weights (except for one row, where we increase the row sums to obtain a nonsingular matrix). Table 4 represents the properties of these matrices.

Table 4: Second set of test problems information

Test matrix	n	nnz	cond
STM_5	25	105	2.0002×10^6
STM_{10}	100	460	8.0012×10^6
STM_{20}	400	1920	3.2006×10^7
STM_{30}	900	4380	7.2016×10^7
STM_{40}	1600	7840	1.2803×10^8
STM_{50}	2500	12300	2.0005×10^8

The results obtained for these matrices (which have large condition number) are presented in Tables 5 and 6. The results of Table 5 show that $MWST_1$ is the best in terms of iteration time (except for $MWST_3$ with $A = STM_5$, $B = STM_{30}$ and RIF with $A = STM_{20}$, $B = STM_{20}$). From the results of Table 6, we observe that, for large matrices, $MWST_1$ is better (in terms of total time) than the other preconditioners (except for RIF with $A = STM_{10}$, $B = STM_{10}$ and $A = STM_{20}$, $B = STM_{20}$). Finally, we consider the results obtained for the preconditioner $MWST_1$ in terms of CPU time, GPU time, and the number of iterations. We mention that the preconditioner was computed on the CPU. All numerical experiments in this section were computed in double precision with a MATLAB code. We used a Geforce GTX 1070 GPU with 8 GBytes VRAM memory. The results are listed in Tables 7 and 8. The notations CIT (GIT) and CIT-time (GIT-time) represent the number of iterations and CPU iteration time (GPU iteration time) on the CPU (GPU) required for convergence, respectively. These tables show that the number of iterations for the CPU and the GPU are close

Table 5: Number of iterations and CPU times to converge for the second set of examples

Matrices		$MWST_1$		$MWST_2$		$MWST_3$		AINV		IC(0)		RIF	
A	B	It	It-time	It	It-time	It	It-time	It	It-time	It	It-time	It	It-time
STM_5	STM_5	101	0.01	85	0.01	84	0.01	195	0.02	165	0.01	83	0.01
STM_5	STM_{10}	390	0.03	360	0.06	357	0.05	1135	0.11	669	0.10	447	0.08
STM_5	STM_{20}	932	1.06	882	1.52	881	1.09	4745	5.87	2152	3.31	1585	2.28
STM_5	STM_{30}	1491	8.86	1437	9.67	1435	8.50	†	†	4299	25.19	3327	24.73
STM_{10}	STM_{10}	831	0.18	768	0.49	762	0.29	1084	0.27	662	0.31	431	0.18
STM_{10}	STM_{20}	2264	4.72	2187	8.63	2186	5.66	7074	14.41	3149	9.62	2322	6.30
STM_{10}	STM_{30}	3596	34.00	3520	52.34	3514	34.80	†	†	6589	66.48	5141	60.66
STM_{20}	STM_{20}	4568	36.58	4420	89.29	4419	44.67	9185	69.22	3721	41.84	2667	29.07
STM_{20}	STM_{30}	7599	216.78	7517	461.86	7508	253.27	†	†	†	†	8309	333.13

Table 6: Preconditioning times and total times for the second set of examples

Matrices		$MWST_1$		$MWST_2$		$MWST_3$		AINV		IC(0)		RIF	
A	B	P-time	T-time	P-time	T-time	P-time	T-time	P-time	T-time	P-time	T-time	P-time	T-time
STM_5	STM_5	0.04	0.05	0.04	0.05	0.02	0.03	0.02	0.04	0.02	0.03	0.02	0.03
STM_5	STM_{10}	0.07	0.10	0.13	0.19	0.05	0.10	0.02	0.13	0.04	0.14	0.02	0.10
STM_5	STM_{20}	0.22	1.28	2.80	4.32	0.27	1.36	0.10	5.97	0.38	3.69	0.10	2.38
STM_5	STM_{30}	0.47	9.33	20.29	29.96	1.55	10.05	0.82	†	1.88	27.07	0.61	25.34
STM_{10}	STM_{10}	0.10	0.28	0.22	0.71	0.08	0.37	0.02	0.29	0.06	0.37	0.02	0.20
STM_{10}	STM_{20}	0.25	4.97	2.89	11.52	0.30	5.96	0.10	14.51	0.40	10.02	0.10	6.4
STM_{10}	STM_{30}	0.50	34.50	20.38	72.72	1.58	36.38	0.82	†	1.90	68.38	0.61	61.27
STM_{20}	STM_{20}	0.40	36.98	5.56	94.85	0.52	45.19	0.18	70.30	0.74	42.58	0.19	29.26
STM_{20}	STM_{30}	0.65	217.43	23.05	484.91	1.8	255.07	0.90	†	2.24	†	0.69	333.82

together and that the GPU time is very smaller than the CPU time for large matrices. So, we can conclude that $MWST_1$ preconditioner in the GL-CG method offers a great potential in a parallel processing environment.

Table 7: The performance of the preconditioner $MWST_1$ on CPU and GPU for the first set of examples

Matrices		$MWST_1$			
A	B	CIt	CIt-time	GIt	GIt-time
NOS6	ST_5	207	0.16	208	0.56
NOS6	ST_{10}	499	0.97	500	1.39
NOS6	ST_{20}	1131	16.78	1131	11.95
NOS6	ST_{30}	1817	84.99	1815	57.91
NOS6	ST_{40}	2489	308.60	2487	200.15
NOS6	ST_{50}	3167	884.18	3162	591.95
NOS6	NOS6	725	22.90	729	18.10

Table 8: The performance of the preconditioner $MWST_1$ on CPU and GPU for the second set of examples

Matrices		$MWST_1$			
A	B	CIt	CIt-time	GIt	GIt-time
STM_{10}	STM_{10}	831	0.18	833	1.09
STM_{10}	STM_{20}	2264	4.72	2267	4.73
STM_{10}	STM_{30}	3596	34.00	3604	20.35
STM_{10}	STM_{40}	4926	152.78	4931	71.82
STM_{10}	STM_{50}	6274	488.41	6287	197.01
STM_{20}	STM_{20}	4568	36.58	4578	24.91
STM_{20}	STM_{30}	7599	216.78	7611	143.72
STM_{20}	STM_{40}	10343	840.34	10354	520.89
STM_{20}	STM_{50}	13010	2234.6	13027	1500.6
STM_{30}	STM_{30}	11171	712.07	11199	491.93
STM_{30}	STM_{40}	15489	2563.8	15504	1731.5
STM_{30}	STM_{50}	19497	6225.7	19521	4999.5
STM_{40}	STM_{40}	20170	6165.1	20207	4156.8
STM_{40}	STM_{50}	25848	15348.00	25873	12075.00
STM_{50}	STM_{50}	31525	30637.00	31573	24410.00

5 Conclusion

We have proposed an approach for computing an approximate solution of matrix equation $AXB = C$, where A and B are Stieltjes matrices. In this approach, by using the BFS algorithm, we presented a new algorithm for

obtaining the inverse of Vaidya's maximum spanning tree preconditioner as an approximate inverse preconditioner. This preconditioner does not require solving any linear systems and is highly parallelizable. We observed that this algorithm furnishes an efficient preconditioner for the matrix equations. The numerical experiments showed that, for large matrices, this preconditioner is better than the other preconditioner in terms of iteration time and total time and the new algorithm is very efficient on the GPU.

Acknowledgements

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

References

1. Bai, Z.Z. *On Hermitian and skew-Hermitian splitting iteration methods for continuous Sylvester equations*, J. Comput. Math. 29(2) (2011), 185–198.

2. Baur, U., Benner, P. *Cross-gramian based model reduction for data-sparse systems*, Electron. Trans. Numer. Anal. 31 (2008), 256–270.

3. Beauwens, R. *Upper eigenvalue bounds for pencils of matrices*, Linear Algebra Appl. 62 (1984), 87–104.

4. Beauwens, R. *Approximate factorizations with modified S/P consistently ordered M-factors*, Numer. Linear Algebra Appl. 1(1) (1994), 3–17.

5. Benzi, M., Meyer, C.D. and Tuma, M. *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput.17(5) (1996), 1135–1149.

6. Benzi, M. and Tuma, M. *A robust incomplete factorization preconditioner for positive definite matrices*, Preconditioning, 2001 (Tahoe City, CA). Numer. Linear Algebra Appl.10(5-6) (2003), 385–400.

7. Bern, M., Gilbert, J.R., Hendrickson, B., Nguyen, N. and Toledo, S. *Support-graph preconditioners*, SIAM J. Matrix Anal. Appl. 27(4) (2006), 930–951.

8. Bouhamidi, A., Jbilou, K., Reichel, L. and Sadok, H. *An extrapolated TSVD method for linear discrete ill-posed problems with kronecker structure*, Linear Algebra Appl. 434(7) (2011), 1677–1688.

9. Calvetti, D. and Reichel, L. *Application of ADI iterative methods to the restoration of noisy images*, SIAM J. Matrix Anal. Appl. 17(1) (1996), 165–186.

10. Datta, B.N. *Numerical methods for linear control systems: design and analysis*, Elsevier, Academic Press, 2004.

11. Deng, Y.B., Bai, Z.Z. and Gao, Y.H. *Iterative orthogonal direction methods for hermitian minimum norm solutions of two consistent matrix equations*, Numer. Linear Algebra Appl. 13(10) (2006), 801–823.

12. George, J.A. *Computer implementation of the finite element method*, Technical report, Department of Computer Science, Stanford University, 1971.

13. George, J.A. and Liu, J.W.H. *Computer Solution of Large Sparse Positive Definite System*, Prentice-Hall, 1981.

14. Gremban, K. D. *Combinatorial preconditioners for sparse, symmetric, diagonally dominant linear systems*, PhD thesis, Carnegie Mellon University, 1996.

15. Gremban, K.D., Miller, G.L. and Zagha, M. *Performance evaluation of a new parallel preconditioner*, 9th International Parallel Processing Symposium, IEEE, (1995), 65–69.

16. Guennouni, A.E., Jbilou, K. and Riquet, A.J. *Block Krylov subspace methods for solving large Sylvester equations*, Matrix iterative analysis and biorthogonality (Luminy, 2000).Numer. Algorithms, 29(1-3) (2002), 75–96.

17. Horn, R.A., and Johnson, C.R. *Topics in matrix analysis*, Cambridge University Press, 1991.

18. Khojasteh-Salkuyeh, D. *Cg-type algorithms to solve symmetric matrix equations,* Appl. Math. Comput. 172(2) (2006), 985–999.

19. Khojasteh-Salkuyeh, D. and Toutounian, F. *New approaches for solving large Sylvester equations,* Appl. Math. Comput. 173(1) (2006), 9–18.

20. Matrix Market, Available at `http//math.nist.gov/Matrix Market,` May 2007.

21. Notay, Y. *Solving positive (semi) definite linear systems by preconditioned iterative methods*, Preconditioned Conjugate Gradient Methods, Lectures Notes in Mathematics, Springer, 1990.

22. Notay, Y. *Conditioning analysis of modified block incomplete factorizations*, Linear Algebra Appl. 154 (1991), 711–722.

23. Notay, Y. *Conditioning of Stieltjes matrices by S/P consistently ordered approximate factorizations*, Appl. Numer. Math. 10(5) (1992), 381–396.

24. Saad, Y. *Iterative methods for sparse linear systems*, SIAM, 2003.

25. Skiena, S.S. *The algorithm design manual: Text,* Springer Science & Business Media, 1998.

26. Vaidya, P.M. *Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners,* Unpublished manuscript UIUC, A talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation, 1991.

27. Wang, M. and Feng, Y. *An iterative algorithm for solving a class of matrix equations,* J. Control Theory Appl. 7(1) (2009), 68–72.

28. Xie, L., Ding, J. and Ding, F. *Gradient based iterative solutions for general linear matrix equations*, Comput. Math. Appl. 58(7) (2009), 1441–1448.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 17–29
DOI:10.22067/ijnao.v9i2.67976

Research Article

Differential transform method for conformable fractional partial differential equations

M. Eslami* and S.A. Taleghani

Abstract

We expand a new generalization of the two-dimensional differential transform method. The new generalization is based on the two-dimensional differential transform method, fractional power series expansions, and conformable fractional derivative. We use the new method for solving a nonlinear conformable fractional partial differential equation and a system of conformable fractional partial differential equation. Finally, numerical examples are presented to illustrate the preciseness and effectiveness of the new technique.

AMS(2010): 46N40; 35Q99.

Keywords: Conformable fractional derivative; Differential transform method; two-dimensional differential transform method.

1 Introduction

Fractional partial differential equations and systems of partial differential equations arise in many areas of mathematics, engineering, and the physical science, which make it very important to find efficient methods for solving the partial fractional differential equations.

The differential transform method is an analytical method for solving differential equations. The concept of differential transform method was first introduced by Zhou in solving linear and nonlinear initial value problems in

*Corresponding author
Received 15 October 2017; revised 24 September 2018; accepted 13 November 2018
Mostafa Eslami
Department of Mathematics, Faculty of Mathematical Sciences, University of Mazandaran, Babolsar, Iran. e-mail: Mostafa.Eslami@umz.ac.ir

Sanaz Abedin Taleghani
Department of Mathematics, Faculty of Mathematical Sciences, University of Mazandaran, Babolsar, Iran. e-mail: Abedintaleghanisanaz@gmail.com

electrical circuit analysis [16]. The method provides an analytical solution in the form of a polynomial.

Fractional calculus [9, 13] is a field of Mathematics in which derivatives and integrals of arbitrary orders are discussed. Studies show many physical systems compatible with fraction derivatives theory. Therefore, there have been significant interest in this subject trying to introduce a definition for fractional derivative in most of which integral forms have been widely used. The two most commonly used definitions are Riemann–Liouville, and Caputo.

Recently, a new kind of fractional derivative called a conformable fractional derivative was proposed by Khalil et al. [8]. This new definition satisfies formulas of derivative of product and quotient of two functions. In addition to the conformable fractional derivative definition, the conformable fractional integral definition, Rolle theorem, and mean value theorem for the conformable fractional differentiable functions were given. Since then, many researchers have made a huge number of scientific articles on the topic. As an instance, Abdeljawad [1] provided fractional versions of the chain rule, exponential functions, Gronwall's inequality, integration by parts, Taylor power series expansions, and Laplace transforms. Cenesiz and Kurt [4,10] found the solutions of time and space-fractional heat differential equations by the conformable fractional derivative and the approximate analytical solution of the time conformable fractional Burger's equation via homotopy analysis method. Acan et al. [2,3] introduced a new type reduced differential transform method called the conformable fractional reduced differential transform method and conformable variational iteration method based on the new defined fractional derivative. Masalmeh [11] applied the series solution for a case of conformable fractional Riccati differential equation with variable coefficients and Ilie et al. [11] introduced a general solution to conformable fractional differential equations. Therefore, it is clearly deducted that further studies and explanations can be made on this newly introduced fractional derivative.

2 Basic definition

Definition 1. [8] Given a function $f : [a, \infty] \to R$. Then the conformable fractional derivative of f order α is defined by

$$\left({}_aT_t^{\alpha} f\right)(t) = \lim_{\varepsilon \to 0} \frac{f(t + \varepsilon(t-a)^{1-\alpha}) - f(t)}{\varepsilon}, \tag{1}$$

for all $t > 0$ and $\alpha \in (0, 1]$.

Definition 2. [1] Given a function $f : [a, \infty] \to R$. Let $n < \alpha \leq n + 1$ and $\beta = \alpha - n$. Then the conformable (left) fractional of f order α, where exists, is defined by

$$(T_{\alpha}^{a} f)(t) = (T_{\beta}^{a} f^{(n)})(t).$$

Definition 3. [15] Assume that $f(x)$ is an infinite-differentiable function for some $\alpha \in (0, 1]$. The conformable fractional differential transform of $f(x)$ is defined as

$$F_{\alpha}(k) = \frac{1}{\alpha^{k} k!} \left[(T_{\alpha}^{x_0} f)^{(k)}(x) \right]_{x=x_0}. \tag{2}$$

Definition 4. The inverse conformable fractional differential transform of $F_{\alpha}(k)$ is defined as [15]

$$f(x) = \sum_{k=0}^{\infty} F_{\alpha}(k)(x - x_0)^{k\alpha}. \tag{3}$$

3 Two-dimensional conformable fractional differential transform method

Consider a function of two variables $u(x, t)$, and suppose that it can be represented as a product of two single variable functions, that is, $u(x, t) = f(x).g(t)$; see [12]. On the basis of the properties of one-dimensional conformable fractional differential transform [15], the function $u(x, t)$ can be represented as

$$\begin{aligned} u(x,t) &= \sum_{k=0}^{\infty} F_{\alpha}(k)(x - x_0)^{k\alpha} \times \sum_{h=0}^{\infty} G_{\beta}(h)(t - t_0)^{h\beta} \\ &= \sum_{k=0}^{\infty} \sum_{h=0}^{\infty} U_{\alpha,\beta}(k, h)(x - x_0)^{k\alpha}(t - t_0)^{h\beta}, \end{aligned} \tag{4}$$

where $0 < \alpha, \beta \leq 1$, $U_{\alpha,\beta}(k, h) = F_{\alpha}(k)G_{\beta}(h)$ is called the spectrum of $u(x, t)$.

The conformable fractional differential transform of a function $u(x, t)$ is as follows:

$$U_{\alpha,\beta}(k, h) = \frac{1}{\alpha^{k} k! \beta^{h} h!} \left[({}_{x_0}T_{x}^{\alpha})^{(k)} ({}_{t_0}T_{t}^{\beta})^{(h)} u(x, t) \right]_{(x_0,t_0)}. \tag{5}$$

Based on (4) and (5), we have the following results.

Theorem 1. *Suppose that $U_{\alpha,\beta}(k, h)$, $V_{\alpha,\beta}(k, h)$, and $W_{\alpha,\beta}(k, h)$ are the conformable fractional differential transformations of the functions $u(x, t)$, $v(x, t)$, and $w(x, t)$, respectively. Then*

1. *if* $u(x,t) = v(x,t) \pm w(x,t)$, *then* $U_{\alpha,\beta}(k,h) = V_{\alpha,\beta}(k,h) \pm W_{\alpha,\beta}(k,h)$;

2. *if* $u(x,t) = C.v(x,t)$, $C \in R$, *then* $U_{\alpha,\beta}(k,h) = C.V_{\alpha,\beta}(k,h)$;

3. *if* $u(x,t) = v(x,t).w(x,t)$, *then*

$$U_{\alpha,\beta}(k,h) = \sum_{r=0}^{k} \sum_{s=0}^{h} V_{\alpha,\beta}(r,h-s).W_{\alpha,\beta}(k-r,s);$$

4. *if* $u(x,t) = v(x,t).w(x,t).z(x,t)$, *then*

$$U_{\alpha,\beta}(k,h) = \sum_{r=0}^{k} \sum_{t=0}^{k-r} \sum_{s=0}^{h} \sum_{p=0}^{h-s} V_{\alpha,\beta}(r,h-s-p).W_{\alpha,\beta}(t,s)Z_{\alpha,\beta}(k-r-t,p);$$

5. *if* $u(x,t) = (x-x_0)^m(t-t_0)^n$, *then* $U_{\alpha,\beta}(k,h) = \delta(k-\frac{m}{\alpha})\delta(h-\frac{n}{\beta})$;

6. *if* $u(x,t) = e^{\lambda(\frac{(x-x_0)^\alpha}{\alpha}+\frac{(t-t_0)^\beta}{\beta})}$, *where* λ *is constant, then* $U_{\alpha,\beta}(k,h) = \frac{\lambda^{k+h}}{\alpha^k k!\beta^h h!}$;

Proof. These items can be proved by using (4) and (5). □

Theorem 2. *If* $u(x,t) = {}_{x_0}T_x^\alpha v(x,t)$, $0 < \alpha \leq 1$, *then*

$$U_{\alpha,\beta}(k,h) = \alpha(k+1).V_{\alpha,\beta}(k+1,h). \tag{6}$$

Proof. From (5), we have

$$\begin{aligned} U_{\alpha,\beta}(k,h) &= \frac{1}{\alpha^k k!\beta^h h!}\left[({}_{x_0}T_x^\alpha)^k({}_{t_0}T_t^\beta)^h {}_{x_0}T_x^\alpha v(x,t)\right]_{(x_0,t_0)} \\ &= \frac{1}{\alpha^k k!\beta^h h!}\left[({}_{x_0}T_x^\alpha)^{k+1}({}_{t_0}T_t^\beta)^h v(x,t)\right]_{(x_0,t_0)} \\ &= \alpha(h+1)V_{\alpha,\beta}(k+1,h). \end{aligned} \tag{7}$$

□

Theorem 3. *If* $u(x,t) = {}_{t_0}T_t^\gamma v(x,t)$, $m < \gamma \leq m+1$, *then*

$$U_{\alpha,\beta}(k,h) = V_{\alpha,\beta}(k,h+\frac{\gamma}{\beta}).\frac{\Gamma(h\beta+\gamma+1)}{\Gamma(h\beta+\gamma-m)}. \tag{8}$$

Proof. Consider the initial condition of the problem,

$$T_t^\gamma v(x,t) = {}_{t_0}T_t^\gamma \left[v(x,t) - f(x) \times \sum_{h=0}^{m} \frac{g^{(h)}(0)}{h!}(t-t_0)^h \right]$$
$$= {}_{t_0}T_t^\gamma \left[f(x) \times \sum_{h=0}^{\infty} G_\beta(h)(t-t_0)^{h\beta} - f(x) \times \sum_{h=0}^{m} \frac{g^{(h)}(0)}{h!}(t-t_0)^h \right].$$

Substituting $h\beta$ in place of k in the second series and considering the initial conditions, we obtain

$$T_t^\gamma v(x,t) = {}_{t_0}T_t^\gamma \left[f(x) \times \sum_{h=0}^{\infty} G_\beta(h)(t-t_0)^{h\beta} - f(x) \times \sum_{h=0}^{\frac{\gamma}{\beta}-1} \frac{g^{(h\beta)}(0)}{(h\beta)!}(t-t_0)^{h\beta} \right]$$
$$= {}_{t_0}T_t^\gamma \left[f(x) \times \sum_{h=0}^{\infty} G_\beta(h)(t-t_0)^{h\beta} - f(x) \times \sum_{h=0}^{\frac{\gamma}{\beta}-1} G_\beta(h)(t-t_0)^{h\beta} \right]$$
$$= {}_{t_0}T_t^\gamma \left[f(x) \times \sum_{h=\frac{\gamma}{\beta}}^{\infty} G_\beta(h)(t-t_0)^{h\beta} \right]$$
$$= f(x) \times \sum_{h=\frac{\gamma}{\beta}}^{\infty} G_\beta(h) \frac{\Gamma(h\beta+1)}{\Gamma(h\beta-m)}(t-t_0)^{h\beta-\gamma}$$
$$= f(x) \times \sum_{h=0}^{\infty} G_\beta(h+\frac{\gamma}{\beta}) \frac{\Gamma(h\beta+\gamma+1)}{\Gamma(h\beta+\gamma-m)}(t-t_0)^{h\beta}$$
$$= \sum_{k=0}^{\infty} F_\alpha(k)(x-x_0)^{k\alpha} \times \sum_{h=0}^{\infty} G_\beta(h+\frac{\gamma}{\beta}) \frac{\Gamma(h\beta+\gamma+1)}{\Gamma(h\beta+\gamma-m)}(t-t_0)^{h\beta}$$
$$= \sum_{k=0}^{\infty} \frac{\Gamma(h\beta+\gamma+1)}{\Gamma(h\beta+\gamma-m)} U_{\alpha,\beta}(k,h+\frac{\gamma}{\beta})(x-x_0)^{k\alpha}(t-t_0)^{h\beta}.$$

According to this result, we have

$$U_{\alpha,\beta}(k,h) = V_{\alpha,\beta}(k,h+\frac{\gamma}{\beta}).\frac{\Gamma(h\beta+\gamma+1)}{\Gamma(h\beta+\gamma-m)}.$$

□

4 Applications

In this section, we will apply the new method to nonlinear conformable fractional partial differential equations and systems of nonlinear fractional partial differential equations. Also, we compare these solutions with the exact solu-

tions.

Example 1. Consider the fractional model of nonlinear Schrödinger equation [7]

$$i\frac{\partial^\beta u(x,t)}{\partial t^\beta} = -\frac{1}{2}\frac{\partial^2 u(x,t)}{\partial x^2} + u\cos^2(x) + |u|^2 u, \quad 0 < \beta \leq 1, \tag{9}$$

subject to the initial condition $u(x,0) = \sin(x)$. Taking the two-dimensional conformable fractional differential transform of (9), then we have

$$\begin{aligned} &i\beta(h+1)U_{1,\beta}(k,h+1) \\ &\quad = \frac{\Gamma(k+2+1)}{\Gamma(k+2-1)}U_{1,\beta}(k+2,h) - \frac{U_{1,\beta}(k,h)}{2} \\ &\quad - \frac{1}{2}\sum_{r=0}^{k}\sum_{s=0}^{h}\frac{2^r}{r!}\cos(\frac{r\pi}{2})U_{1,\beta}(k-r,s) \\ &\quad - \sum_{r=0}^{k}\sum_{t=0}^{k-r}\sum_{s=0}^{h}\sum_{p=0}^{h-s}U_{1,\beta}(r,h-s-p)\overline{U_{1,\beta}}(t,s)U_{1,\beta}(k-r-t,p) \end{aligned} \tag{10}$$

and from the initial condition, we get

$$U_{1,\beta}(k,0) = \frac{1}{k!}\sin(\frac{k\pi}{2}), \quad k = 0,1,\ldots. \tag{11}$$

Substituting (10) to (11), the following series solution is obtained:

$$\begin{aligned} u(x,t) &= \sum_{k=0}^{\infty}\sum_{h=0}^{\infty}U_{1,\beta}(k,h)x^k t^{h\beta} \\ &= \left(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots\right) \times \left(1 - \frac{3it^\beta}{2\beta} - \frac{9}{4}\frac{t^{2\beta}}{\alpha^2 2!} + \frac{27}{8}\frac{it^{3\beta}}{\beta^3 3!} - \frac{81}{16}\frac{t^{4\beta}}{\beta^4 4!} - \cdots\right) \\ &= \sin(x)e^{-\frac{3it^\beta}{2\beta}}, \end{aligned}$$

which is the exact solution.

Example 2. We consider the Klein–Gordon equation

$$\frac{\partial^\beta u}{\partial t^\beta} - \frac{\partial^2 u}{\partial x^2} + u^2 = -x\sin(\frac{t^\beta}{\beta}) + x^2\cos^2(\frac{t^\beta}{\beta}), \quad 0 < \beta \leq 1, \tag{12}$$

subject to the initial conditions

$$u(x,0) = x. \tag{13}$$

Taking the two-dimensional conformable fractional differential transform of (12), then we have

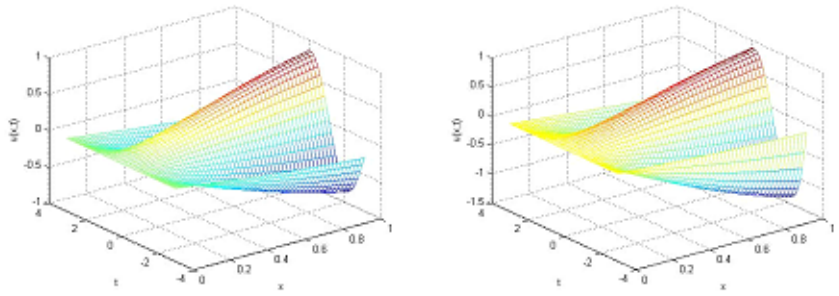


Figure 1: Graphs of the $u(x,t)$ for $\beta = 1$ and $\beta = 0.95$, when $N = 10$ (the number of terms), form left to right

$$\begin{aligned}\beta(h+1)U_{1,\beta}(k,h+1) = {} & \frac{\Gamma(k+2+1)}{\Gamma(k+2-1)}U_{1,\beta}(k+2,h) \\ & - \sum_{r=0}^{k}\sum_{s=0}^{h} U_{1,\beta}(r,h-s)U_{1,\beta}(k-r,s) \\ & - \delta(k-1)\frac{1}{h!}\sin(\frac{h\pi}{2}) + \frac{1}{2}\delta(k-2) \\ & + \frac{1}{2}\delta(k-2).\frac{2^h}{h!}\cos(\frac{h\pi}{2}). \end{aligned} \tag{14}$$

From the initial conditions (13), we can write

$$U_{1,\beta}(0,0) = 0, \qquad U_{1,\beta}(1,0) = 1 \qquad U_{1,\beta}(k,0) = 0, \quad k = 2,3,\ldots. \tag{15}$$

Substituting (15) into (14), we obtain the series solution as

$$\begin{aligned} u(x,t) &= \sum_{k=0}^{\infty}\sum_{h=0}^{\infty} U_{1,\beta}(k,h)x^k t^{h\beta} \\ &= x\left(1 - \frac{t^{2\beta}}{\beta 2!} + \frac{t^{4\beta}}{\beta^4 4!} - \cdots\right) = x.\cos(\frac{t^\beta}{\beta}), \end{aligned}$$

which is the exact solution.

Example 3. Consider the nonlinear time fractional Fokker–Planck equation [14]

$$\frac{\partial^\beta u}{\partial t^\beta} = [-\frac{\partial}{\partial x}(3u - \frac{x}{2}) + \frac{\partial^2}{\partial x^2}(xu)]u, \tag{16}$$

where $x > 0$, $t > 0$, $0 < \beta \leq 1$, and the initial condition is

$$u(x,0) = x. \tag{17}$$

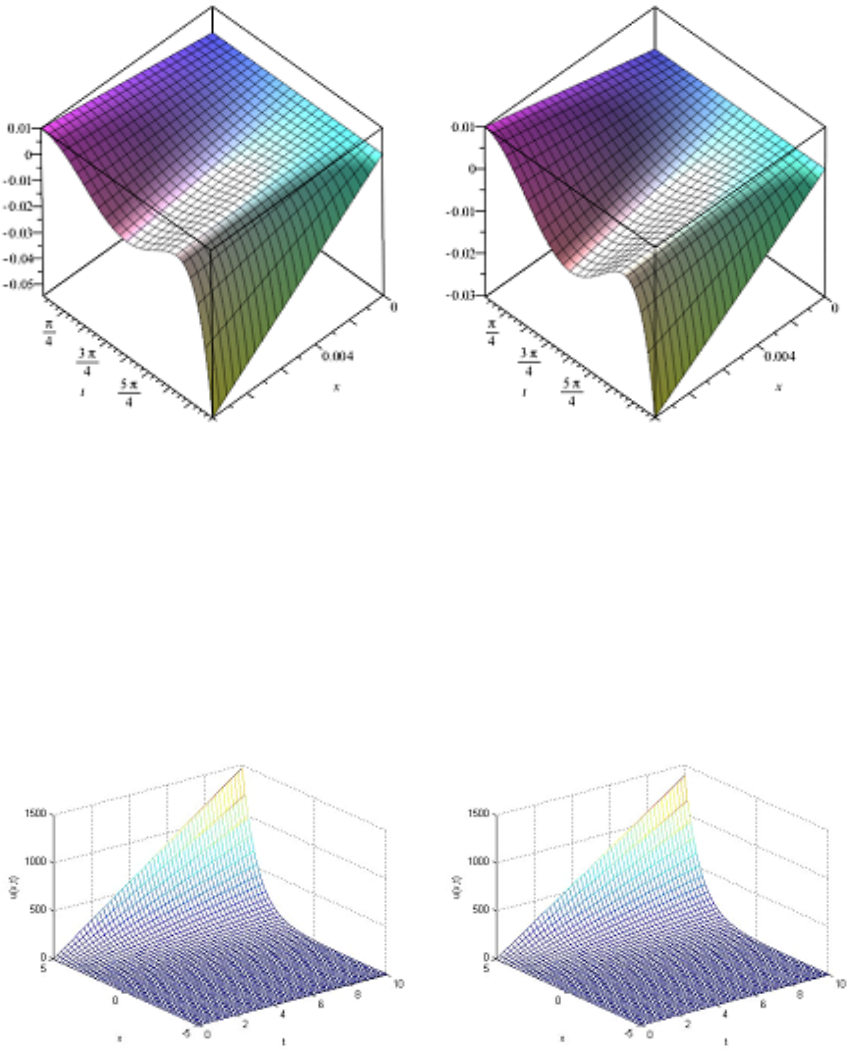


Figure 2: Graphs of the $u(x,t)$ for $\beta = 1$ and $\beta = 0.95$, when $N = 10$ (the number of terms), from left to right

Selecting $\alpha = 1$ and taking the two-dimensional conformable fractional differential transform of (16), we have

$$\begin{aligned}\beta(h+1)U_{1,\beta}(k,h+1) &= -(k+1)\left\{3\sum_{r=0}^{k+1}\sum_{s=0}^{h}U_{1,\beta}(r,h-s)U_{1,\beta}(k+1-r,s)-\frac{1}{2}U_{1,\beta}(k,h)\right\}\\ &\quad+(k+2)(k+1)\sum_{r=0}^{k+1}\sum_{s=0}^{h}U_{1,\beta}(r,h-s)U_{1,\beta}(k+1-r,s). \qquad (18)\end{aligned}$$

By using the initial condition (17), we write

$$U_{1,\beta}(k,0) = \delta(k-1). \qquad (19)$$

Substituting (19) in (18), we obtain the closed form series solution as

$$\begin{aligned}u(x,t) &= \sum_{k=0}^{\infty}\sum_{h=0}^{\infty}U_{1,\beta}(k,h)x^k t^{h\beta}\\ &= x(1+\frac{t^\beta}{\beta}+\frac{t^{2\beta}}{2\beta^2}+\frac{t^{3\beta}}{6\beta^3}+\cdots) = x\sum_{h=0}^{\infty}\frac{t^{h\beta}}{\beta^h h!} = xe^{\frac{t^\beta}{\beta}},\end{aligned}$$

which is the exact solution.

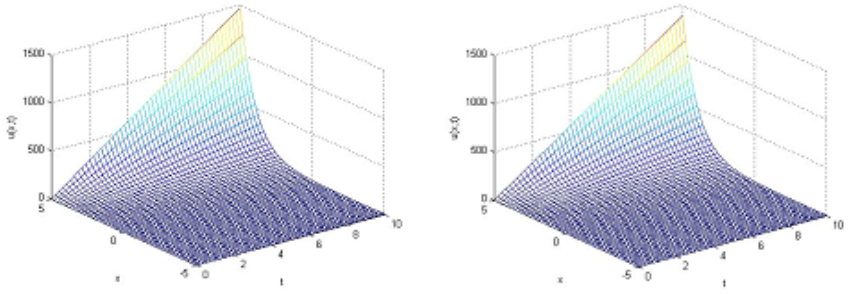


Figure 3: Graphs of the $u(x,t)$ for $\beta = 1$ and $\beta = 0.98$, when $N = 10$ (the number of terms), from left to right.

Example 4. Consider the time fractional inhomogeneous nonlinear system of PDEs [5]

$$\begin{cases}\frac{\partial^\beta u}{\partial t^\beta}+v\frac{\partial u}{\partial x}+u=1, & 0<\beta\le 1,\\ \frac{\partial^\beta v}{\partial t^\beta}-u\frac{\partial v}{\partial x}-v=1,\end{cases} \qquad (20)$$

subject to the initial conditions

$$u(x,0) = e^{x}\,,\; v(x,0) = e^{-x}\,. \tag{21}$$

Taking the two-dimensional conformable fractional differential transform of (20), we have

$$\begin{cases} \beta(h+1)U(k,h+1) = -\sum\limits_{r=0}^{k}\sum\limits_{s=0}^{h}(k-r+1)V(r,h-s)U(k-r+1,s) \\ \qquad\qquad - U(k,h) + \delta(k)\delta(h) \\ \beta(h+1)V(k,h+1) = \sum\limits_{r=0}^{k}\sum\limits_{s=0}^{h}(k-r+1)U(r,h-s)V(k-r+1,s) \\ \qquad\qquad + V(k,h) + \delta(k)\delta(h) \end{cases} \tag{22}$$

By using the initial conditions (21), we write

$$\begin{cases} U(k,0) = \frac{1}{k!}, \\ V(k,0) = \frac{(-1)^k}{k!}. \end{cases} \tag{23}$$

Substituting (23) in (22), we obtain the closed form series solution as

$$\begin{aligned} u(x,t) &= \sum_{k=0}^{\infty}\sum_{h=0}^{\infty}U(k,h)x^k t^{h\beta} \\ &= (1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots)(1 - \frac{t^\alpha}{\alpha} + \frac{t^{2\alpha}}{\alpha^2 2!} - \frac{t^{3\alpha}}{\alpha^3 3!} + \cdots) = e^{x - \frac{t^\beta}{\beta}} \\ v(x,t) &= \sum_{k=0}^{\infty}\sum_{h=0}^{\infty}V(k,h)x^k t^{h\beta} \\ &= (1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots)(1 + \frac{t^\alpha}{\alpha} + \frac{t^{2\alpha}}{\alpha^2 2!} + \frac{t^{3\alpha}}{\alpha^3 3!} + \cdots) = e^{-x + \frac{t^\beta}{\beta}} \end{aligned}$$

which are the exact solutions.

5 Conclusion

In this paper, we presented a two-dimensional conformable fractional differential transform method. Then we apply the new method to some conformable fractional partial differential equations and system of conformable fractional partial differential equations. Comparison of the results obtained by using the new method with exact solution reveals that the present method is very effective and convenient for solving nonlinear partial differential equations and system of partial differential equations of fractional order.

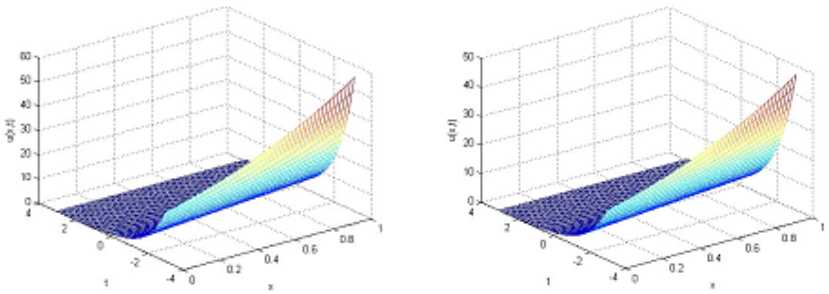


Figure 4: Graphs of the $u(x,t)$ for $\beta = 1$ and $\beta = 0.95$, when $N = 10$ (the number of terms), from left to right.

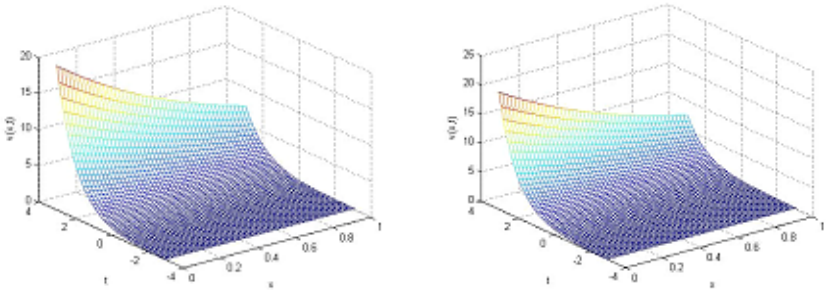


Figure 5: Graphs of the $v(x,t)$for $\beta = 1$,$\beta = 0.95$ when N=10 (the number of terms), from left to right.

References

1. Abdeljawad, T. *On fractional calculus*, J. Comput. Appl. Math. 279 (2015) 57–66.
2. Acan, O.,Firat, O., Keskin, Y. and Oturanc, G.*Solution of conformable fractional partial differential equations by reduced differential transform method*, Selcuk J. Appl. Math. (2016) (In press).
3. Acan, O., Firat, O., Keskin, Y. and Oturanc, G. *Conformable variational iteration method*, New Trends in Mathematical Sciences, 5(1) (2017) 172–178.
4. Cenesiz, Y. and Kurt, A.*The solutions of time and space conformable fractional heat equations with conformable Fourier transform*, Acta Univ. Sapientiae Math. 7(2) (2015), 130–140.
5. Garg, M. and Manohar, P. *Analytical solution of space-time fractional Fokker Planck equations by generalized differential transform method.* Matematiche (Catania) 66(2) (2011), 91–101.
6. Ilie, M., Biazar, J. and Ayati, Z.*General solution of Bernoulli and Riccati fractional differential equations based on conformable fractional derivative*, Int. J. Appl. Math. Res. 6 (2017) 49–51.
7. Kanth, A.S.V.R. and Aruna, K.*Two-dimensional differential transform method for solving linear and non-linear Schrödinger equations*, Chaos, Solitons. Fract. 41 (2009) 2277–2281.
8. Khalil, R., Al Horani, M., Yousef, A., and Sababheh, M.*A new definition of fractional derivative*, J. Comput. Appl. Math. 264 (2014), 65–70.
9. Kilbas, A.A., Srivastava, H.M. and Trujillo, J.J. *Theory and applications of fractional differential equations*. North-Holland Mathematics Studies, 204. Elsevier Science B.V., Amsterdam, 2006.
10. A. Kurt, Y. Cenesiz, O. Tasbozan, *On the solution of Burgers' equation with the new fractional derivative*, Open Phys. 13 (2015) 355–360.
11. Masalmeh, M. *Series method to solve conformable fractional Riccati differential equations*, Int. J. Appl. Math. Res. 6 (2017) 30–33.
12. Odibat, Z. and Momani, S.*A generalized differential transform method for linear partial differential equations of fractional order*. Appl. Math. Lett. 21(2) (2008), 194–199.
13. Podlubny, I.*Fractional differential equations. An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications.* Mathematics in Science and Engineering, 198. Academic Press, Inc., San Diego, CA, 1999.

14. Sohail, M. and Mohyud-Din, S.T. *Reduced differential transform method for solving a system of fractional PDEs*, International J. Modern Math. Sci. 4 (2012), 21–29.

15. Ünal, E. and Gökdoethan, A. *Solution of conformable fractional ordinary differential equations via differential transform method*, Optik-International Journal for Light and Electron Optics, 128 (2017) 264–273.

16. Zhou, J.K. *Differential Transformation and its Application for Electrical Circuits*, Huarjung University Press Wuuhahn, China, 1986, (in Chinese).

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 31–47
DOI:10.22067/ijnao.v9i2.73724

Research Article

Approximate solution of a system of singular integral equations of the first kind by using Chebyshev polynomials

S. Ahdiaghdam* and S. Shahmorad

Abstract

The aim of the present work is to introduce a method based on the Chebyshev polynomials for numerical solution of a system of Cauchy type singular integral equations of the first kind on a finite segment. Moreover, an estimation error is computed for the approximate solution. Numerical results demonstrate the effectiveness of the proposed method.

AMS(2010): 45F15; 33C45; 42C10; 41A50.

Keywords: System of singular linear integral equations; Orthogonal polynomials; Fourier series; Best approximation.

1 Introduction

Let us consider a system of singular integral equations of the form

$$A(t)\Phi(t) + B(t)\int_{-1}^{1}\frac{\Phi(\tau)}{\tau - t}\,d\tau + \int_{-1}^{1}K(t,\tau)\Phi(\tau)d\tau = F(t), \quad -1 < t < 1, \quad (1)$$

where

*Corresponding author
Received 25 June 2018; revised 7 December 2018; accepted 23 January 2019
Samad Ahdiaghdam
Department of Mathematics, Marand Branch, Islamic Azad University, Marand, Iran. e-mail: ahdi@marandiau.ac.ir

Sedaghat Shahmorad
Department of Applied Mathematics, University of Tabriz, Tabriz, Iran.
e-mail: shahmorad@tabrizu.ac.ir

$$\begin{aligned} K(t,\tau) &= [K_{ij}(t,\tau)]\,, \quad i,j = 1,2,\ldots,N,\\ F(t) &= [f_1(t)\,, f_2(t),\ldots,f_N(t)]^T,\\ \Phi(t) &= [\phi_1(t),\phi_2(t),\ldots,\phi_N(t)]^T,\\ A(t) &= [a_{ij}(t)]\,, \quad i,j = 1,2,\ldots,N,\\ B(t) &= [b_{ij}(t)]\,, \quad i,j = 1,2,\ldots,N. \end{aligned}$$

Here, $\{K_{ij}\}_{i,j=1}^{N}$ and $\{f_i\}_{i=1}^{N}$ are given real-valued Hölder functions and $\{\phi_j\}_{j=1}^{N}$ are the unknown functions. The matrices A and B are known such that $S = A + B$ and $D = A - B$ are nonsingular for all $t \in [-1, 1]$. In some familiar physical problems, the entries of the matrices A and B are constants.

The singular integral equations play important roles in physics and theoretical mechanics, particularly in the areas of elasticity, aerodynamics, and unsteady aerofoil theory. They are highly effective in solving boundary value problems occurring in the theory of functions of a complex variable, potential theory, the theory of elasticity, and the theory of fluid mechanics. A general theory of the system of equations (1) has given in [12].

We study the system (1) in the case that $A(t) = 0$ and $B(t)$ is a constant matrix. Therefore, the ith equation of system (1) takes the form

$$\sum_{j=1}^{N} b_{ij} \int_{-1}^{1} \frac{\phi_j(\tau)}{\tau - t}\, d\tau + \sum_{j=1}^{N} \int_{-1}^{1} K_{ij}(t,\tau)\phi_j(\tau) d\tau = f_i(t)\,, \quad -1 < t < 1. \tag{2}$$

Studies on this singular integral equation can be found in some literatures (see [1,3,6,7]). Chakrabarti and Berghe [3] proposed a method for solving (2), using polynomial approximation and collocation points have chosen to be the zeros of the Chebyshev polynomials of the first kind for all cases. Kashfi and Shahmorad [7] constructed another approximate solution of this equation by using the Chebyshev polynomials of the first and second kinds. Some other methods for solving this equation can be found in [1,6]. A convergence analysis of Galerkin and collocation methods for (2) has been given by Miel [11].

A special type of (2) is the famous Cauchy singular integral equation

$$\int_{-1}^{1} \frac{\phi(\tau)}{\tau - t}\, d\tau = f(t)\,, \quad -1 < t < 1, \tag{3}$$

which has the following analytical solutions in four special cases based on boundedness of the unknown function ϕ at the endpoints of the interval $[-1, 1]$; see [3,9,13].

Case 1. If the function ϕ is unbounded at the endpoints $\tau = \pm 1$, then

$$\phi(\tau) = \frac{a_0}{\sqrt{1-\tau^2}} - \frac{1}{\pi^2\sqrt{1-\tau^2}} \int_{-1}^{1} \frac{\sqrt{1-t^2} f(t)}{t-\tau}\, dt\,, \quad -1 < \tau < 1,$$

where a_0 is an arbitrary constant.

Case 2. If the function ϕ is bounded at the endpoints $\tau = \pm 1$, then

$$\phi(\tau) = -\frac{\sqrt{1-\tau^2}}{\pi^2} \int_{-1}^{1} \frac{f(t)}{\sqrt{1-t^2}\,(t-\tau)}\, dt\,, \quad -1 < \tau < 1,$$

and a necessary and sufficient condition of existing this solution is

$$\int_{-1}^{1} \frac{f(t)}{\sqrt{1-t^2}}\, dt = 0\,.$$

Case 3. If the function ϕ is bounded at the endpoint $\tau = -1$ and unbounded at the endpoint $\tau = 1$, then

$$\phi(\tau) = -\frac{1}{\pi^2} \sqrt{\frac{1+\tau}{1-\tau}} \int_{-1}^{1} \sqrt{\frac{1-t}{1+t}} \frac{f(t)}{t-\tau}\, dt\,, \quad -1 < \tau < 1.$$

Case 4. If the function ϕ is bounded at the endpoint $\tau = 1$ and unbounded at the endpoint $\tau = -1$, then

$$\phi(\tau) = -\frac{1}{\pi^2} \sqrt{\frac{1-\tau}{1+\tau}} \int_{-1}^{1} \sqrt{\frac{1+t}{1-t}} \frac{f(t)}{t-\tau}\, dt\,, \quad -1 < \tau < 1.$$

An application of (3) were given in [2] by reducing a system of dual integral equations to Cauchy type singular integral equations, and more methods for solving this equation have given in [5, 8, 13, 15].

In the next section, we investigate approximate solutions for system (1) in the above four cases.

2 Approximate solution

To find approximate solutions for system (1) in the cases **1,2,3,4**, for $\nu \in \{1, 2, 3, 4\}$, we set

$$\phi_j(\tau) \simeq \varphi_{\nu,j}(\tau) := \frac{\lambda_\nu(\tau)}{\sqrt{1-\tau^2}} \sum_{l=0}^{M} \beta_{jl}\, P_{\nu,l}(\tau)\,, \quad j = 1, 2, \ldots, N, \tag{4}$$

and

$$K_{ij}(t,\tau) := \sum_{k=0}^{M} \gamma_{ijk}(t)\, P_{\nu,k}(\tau)\,, \quad i,j = 1,2,\ldots,N\,, \tag{5}$$

where

$$P_{\nu,j}(x) = \begin{cases} T_j(x) = \cos(j\theta), & \nu = 1, \\ U_j(x) = \frac{\sin((j+1)\theta)}{\sin(\theta)}, & \nu = 2, \\ V_j(x) = \frac{\cos\left((j+\frac{1}{2})\theta\right)}{\cos(\frac{\theta}{2})}, & \nu = 3, \\ W_j(x) = \frac{\sin\left((j+\frac{1}{2})\theta\right)}{\sin(\frac{\theta}{2})}, & \nu = 4, \end{cases}$$

are the Chebyshev polynomials of the first to fourth kinds and

$$\lambda_\nu(t) = \begin{cases} 1, & \nu = 1, \\ 1-t^2, & \nu = 2, \\ 1+t, & \nu = 3, \\ 1-t, & \nu = 4, \end{cases}$$

in which $x = \cos(\theta)$. The roots of Chebyshev polynomials $P_{\nu,M+1}(x)$ are given by

$$x_{\nu,n} = \begin{cases} \cos\left(\frac{(2n-1)\pi}{2(M+1)}\right), & \nu = 1, \\ \cos\left(\frac{n\pi}{M+2}\right), & \nu = 2, \\ \cos\left(\frac{(2n-1)\pi}{2M+3}\right), & \nu = 3, \\ \cos\left(\frac{2n\pi}{2M+3}\right), & \nu = 4, \end{cases} \tag{6}$$

where $n = 1,2,\ldots,M+1$. These roots are used as the nodes of Gauss–Chebyshev quadrature rules.

Lemma 1. *[10] The Chebyshev polynomials satisfy the orthogonality property*

$$\int_{-1}^{1} \frac{\lambda_\nu(t)}{\sqrt{1-t^2}} P_{\nu,i}(t) P_{\nu,j}(t)\, dt = \begin{cases} 0, & i \neq j, \\ \pi, & i = j = 0\,, \quad \nu = 1, \\ \frac{\pi}{2}, & i = j \neq 0, \quad \nu = 1, \\ \frac{\pi}{2}, & i = j, \qquad \nu = 2, \\ \pi, & i = j, \qquad \nu = 3\,,\ 4. \end{cases} \tag{7}$$

Theorem 1. *[10] As the Cauchy principle value of a singular integral, we have*

$$\int_{-1}^{1} \frac{\lambda_\nu(\tau)}{\sqrt{1-\tau^2}} \frac{P_{\nu,j}(\tau)}{\tau - t}\, d\tau = \pi \begin{cases} U_{j-1}(t), & \nu = 1, \\ -T_{j+1}(t), & \nu = 2, \\ W_j(t), & \nu = 3, \\ -V_j(t), & \nu = 4. \end{cases} \tag{8}$$

Now we describe details of finding approximate solution in cases **1-4**.

Case 1. For $\nu = 1$, the relations (4)–(5) take the forms

$$\phi_j(\tau) \simeq \varphi_{1,j}(\tau) := \frac{1}{\sqrt{1-\tau^2}} {\sum_{l=0}^{M}}' \beta_{jl}\, T_l(\tau)\,, \qquad j = 1, 2, \ldots, N, \tag{9}$$

and

$$K_{ij}(t,\tau) := {\sum_{k=0}^{M}}' \gamma_{ijk}(t)\, T_k(\tau)\,, \qquad i, j = 1, 2, \ldots, N, \tag{10}$$

where β_{jl} $(j = 1, 2, \ldots, N, \quad l = 0, 1, \ldots, M)$ are unknown coefficients and the symbol $(\sum')$ denotes that the first term in the summation is halved. The functions

$$\gamma_{ijk}(t) = \frac{2}{\pi} \int_{-1}^{1} \frac{K_{ij}(t,\tau) T_k(\tau)}{\sqrt{1-\tau^2}}\, d\tau\,, \quad i, j = 1, 2, \ldots, N, \quad k = 0, 1, \ldots, M,$$

can be determined exactly or may be approximated by using the Gauss–Chebyshev quadrature rule, that is,

$$\gamma_{ijk}(t) \simeq \frac{2}{M+1} \sum_{s=1}^{M+1} K_{ij}(t, x_{1,s}) T_k(x_{1,s})\,,$$

where $x_{1,s}$ obtain from (6).

Substituting from (9)–(10) into (2) and using (7)–(8) for $\nu = 1$, gives the system

$$\sum_{j=1}^{N}\sum_{l=1}^{M} b_{ij}\beta_{jl}U_{l-1}(t) + \frac{1}{2}\sum_{j=1}^{N}{\sum_{k=0}^{M}}' \gamma_{ijk}(t)\beta_{jk} = \frac{1}{\pi}f_i(t)\,, \quad i = 1, 2, \ldots, N. \tag{11}$$

If the given functions f_i and γ_{ijk} are square integrable on $[-1\,,\ 1]$ with respect to the weight function $\frac{\lambda_1(t)}{\sqrt{1-t^2}}$, then they can be expanded as

$$\begin{cases} \gamma_{ijk}(t) \simeq \sum_{l=0}^{M-1} G_{ijkl}\, U_l(t),\ i, j = 1, 2, \ldots, N,\ k = 0\,, 1\,, \ldots\,, M, \\ \\ \frac{1}{\pi} f_i(t) \simeq \sum_{l=0}^{M-1} c_{il}\, U_l(t), \qquad i = 1, 2, \ldots, N, \end{cases} \tag{12}$$

where the coefficients

$$\begin{cases} G_{ijkl} = \frac{2}{\pi}\int_{-1}^{1}\sqrt{1-t^2}\gamma_{ijk}(t)U_l(t)dt \\ \qquad\quad = \frac{4}{\pi^2}\int_{-1}^{1}\int_{-1}^{1}\sqrt{\frac{1-t^2}{1-\tau^2}}K_{ij}(t,\tau)U_l(t)\,T_k(\tau)d\tau dt \\ \qquad\qquad\quad i, j = 1, 2, \ldots, N, \quad k = 0, 1, \ldots, M,\ l = 0, 1, \ldots, M-1, \\ \\ c_{il} \quad = \frac{1}{\pi^2}\int_{-1}^{1}\sqrt{1-t^2}f_i(t)U_l(t)dt,\ i = 1, 2, \ldots, N,\ l = 0, 1, \ldots, M-1, \end{cases}$$

can be approximately determined from

$$\begin{cases} G_{ijkl} \simeq \frac{4}{(M+1)^2}\sum_{r=1}^{M}\sum_{s=1}^{M+1}(1-x_{2,r}^2)K_{ij}(x_{2,r}, x_{1,s})U_l(x_{2,r})T_k(x_{1,s}), \\ \\ c_{il} \simeq \frac{2}{\pi(M+1)}\sum_{r=1}^{M}(1-x_{2,r}^2)f_i(x_{2,r})U_l(x_{2,r}). \end{cases} \tag{13}$$

Using (12) in (11), we have

$$\sum_{l=0}^{M-1}\sum_{j=1}^{N} b_{ij}\beta_{j\{l+1\}}U_l(t) + \frac{1}{2}\sum_{l=0}^{M-1}\sum_{j=1}^{N}{\sum_{k=0}^{M}}' G_{ijkl}\beta_{jk}U_l(t) = \sum_{l=0}^{M-1} c_{il}\, U_l(t),$$

which leads to the linear system

$$\sum_{j=1}^{N}\left[b_{ij}\beta_{j\{l+1\}} + \frac{1}{2}{\sum_{k=0}^{M}}' G_{ijkl}\beta_{jk}\right] = c_{il},\ i = 1, 2, \ldots, N,\ l = 0, 1, \ldots, M-1, \tag{14}$$

for the unknown values β_{jk} $(j = 1, 2, \ldots, N, \quad k = 0, 1, \ldots, M)$. Taking $\beta_{11}, \ldots, \beta_{N1}$ arbitrary, the remaining coefficients β_{jk} are uniquely found from the linear system (14) which determine the elements of the vector function Φ via (9).

Case 2. We set $\nu = 2$ in (4)–(5) and substitute them in (2) to get

$$-\sum_{j=1}^{N}\sum_{l=0}^{M} b_{ij}\beta_{jl}T_{l+1}(t) + \frac{1}{2}\sum_{j=1}^{N}\sum_{k=0}^{M}\gamma_{ijk}(t)\beta_{jk} = \frac{1}{\pi}f_i(t), \quad i = 1, 2, \ldots, N, \tag{15}$$

where we used the formulas (7)–(8). Then, we expand the functions f_i and γ_{ijk} as

$$\begin{cases} \gamma_{ijk}(t) \simeq \sum_{l=0}^{M}{}' G_{ijkl}\, T_l(t)\,, \quad i, j = 1, 2, \ldots, N, \;\; k = 0\,, 1\,, \ldots\,, M \\ \\ \frac{1}{\pi} f_i(t) \simeq \sum_{l=0}^{M}{}' c_{il}\, T_l(t)\,, \qquad i = 1, 2, \ldots, N, \end{cases}$$

where the coefficients are determined by

$$\begin{cases} G_{ijkl} = \frac{2}{\pi}\int_{-1}^{1} \frac{1}{\sqrt{1-t^2}}\gamma_{ijk}(t)T_l(t)\,dt, \quad i, j = 1, 2, \ldots, N,\, k, l = 0, 1, \ldots, M, \\ \qquad\quad = \frac{4}{\pi^2}\int_{-1}^{1}\int_{-1}^{1}\sqrt{\frac{1-\tau^2}{1-t^2}}K_{ij}(t,\tau)T_l(t)\,U_k(\tau)\,d\tau dt \\ \\ c_{il} = \frac{2}{\pi^2}\int_{-1}^{1}\frac{1}{\sqrt{1-t^2}}f_i(t)T_l(t)\,dt, \qquad i = 1, 2, \ldots, N, \qquad l = 0, 1, \ldots, M, \end{cases}$$

or approximated by

$$\begin{cases} G_{ijkl} \simeq \frac{4}{(M+1)(M+2)}\sum_{r=1}^{M+1}\sum_{s=1}^{M+1}(1 - x_{2,s}^2)K_{ij}(x_{1,r}, x_{2,s})T_l(x_{1,r})U_k(x_{2,s}), \\ \\ c_{il} \simeq \frac{2}{\pi(M+1)}\sum_{r=1}^{M+1} f_i(x_{1,r})T_l(x_{1,r}). \end{cases}$$

Using the last expansions in (15), returns the following linear system of equations

$$\begin{cases} \frac{1}{2}\sum_{j=1}^{N}\sum_{k=0}^{M} G_{ijkl}\beta_{jk} = c_{il}, \qquad\qquad i = 1, \ldots, N,\, l = 0, \\ \\ \sum_{j=1}^{N}\left\{-b_{ij}\beta_{j\{l-1\}} + \frac{1}{2}\sum_{k=0}^{M} G_{ijkl}\beta_{jk}\right\} = c_{il},\, i = 1, \ldots, N,\, l = 1, \ldots, M, \end{cases}$$

for the unknown values β_{jl} $(j = 1, 2, \ldots, N, \quad l = 0, 1, \ldots, M)$. Then elements of the vector function $\Phi(t)$ obtain from (4).

Cases 3,4. Similar to cases 1 and 2, we get the linear systems

$$\sum_{j=1}^{N}\left\{b_{ij}\beta_{jl} + \sum_{k=0}^{M} G_{ijkl}\beta_{jk}\right\} = c_{il}, \quad i = 1, 2, \ldots, N, \quad l = 0, 1, \ldots, M, \tag{16}$$

and

$$\sum_{j=1}^{N}\left\{-b_{ij}\beta_{jl}+\sum_{k=0}^{M}G_{ijkl}\beta_{jk}\right\}=c_{il},\quad i=1,2,\ldots,N,\quad l=0,1,\ldots,M,(17)$$

respectively for $\nu=3$ and $\nu=4$, and then we determine the elements of corresponding vector Φ via (4).

3 An estimation error and numerical results

In this section, we describe an estimation error for the approximate solution. Let

$$\overline{\Phi}(t)=[\varphi_1(t)\,,\,\varphi_2(t)\,,\,\ldots\,,\,\varphi_N(t)]^T$$

be the vector of approximate solution of the system (1) and let $E(t)=\overline{\Phi}(t)-\Phi(t)$ be the associated vector valued error function. Due to the approximation $\overline{\Phi}(t)$, for $A(t)=0$, system (1) may be written as

$$B(t)\int_{-1}^{1}\frac{\overline{\Phi}(\tau)}{\tau-t}\,d\tau+\int_{-1}^{1}K(t,\tau)\overline{\Phi}(\tau)d\tau=F(t)+H(t),\quad -1<t<1,\ (18)$$

where the perturbation term $H(t)$ obtains from

$$H(t)=B(t)\int_{-1}^{1}\frac{\overline{\Phi}(\tau)}{\tau-t}\,d\tau+\int_{-1}^{1}K(t,\tau)\overline{\Phi}(\tau)d\tau-F(t),\quad -1<t<1.$$

Subtracting (18) from (1) yields a system of error equations as

$$B(t)\int_{-1}^{1}\frac{E(\tau)}{\tau-t}\,d\tau+\int_{-1}^{1}K(t,\tau)E(\tau)d\tau=H(t)\,,\quad -1<t<1,$$

which is solvable approximately like system (1).

The following examples illustrate application of the method.

Example 1. Let

$$A(t)=0,\quad B(t)=\begin{bmatrix}1&0\\0&1\end{bmatrix},\quad K(t,\tau)=\begin{bmatrix}\tau-t&t\\\tau&\tau+t\end{bmatrix},\quad F(t)=\begin{bmatrix}\pi\\2\pi t\end{bmatrix},$$

and find the solution of system (1) in case **1**.

By the above information, the system (1) is reduced to

$$\begin{cases}\int_{-1}^{1}\frac{\phi_1(\tau)}{\tau-t}\,d\tau+\int_{-1}^{1}(\tau-t)\phi_1(\tau)d\tau+\int_{-1}^{1}t\phi_2(\tau)d\tau=\pi,\quad -1<t<1,\\ \int_{-1}^{1}\frac{\phi_2(\tau)}{\tau-t}\,d\tau+\int_{-1}^{1}\tau\phi_1(\tau)d\tau+\int_{-1}^{1}(\tau+t)\phi_2(\tau)d\tau=2\pi t,\ -1<t<1,\end{cases}\quad (19)$$

and since the matrices $S = A+B = I_2$ and $D = A-B = -I_2$ are nonsingular, then system (19) has a unique solution. The kernels $K_{1j}(t,\tau)$, $K_{2j}(t,\tau)$ $(j = 1, 2)$, and the functions f_1 and f_2 are polynomials of degree at most 1, so we set

$$\phi_j(\tau) := \frac{1}{\sqrt{1-\tau^2}} \left\{ \beta_{j0}\, T_0(\tau) + \beta_{j1}\, T_1(\tau) + \beta_{j2}\, T_2(\tau) \right\}, \quad j = 1, 2, \tag{20}$$

and

$$\begin{array}{ll} K_{ij}(t,\tau) = \gamma_{ij0}(t)T_0(\tau) + \gamma_{ij1}(t)T_1(\tau), & i,j = 1,2, \\ f_i(t) = c_{i0}U_0(t) + c_{i1}U_1(t), & i = 1,2, \end{array}$$

where

$$\begin{array}{llll} \gamma_{110}(t) = -\frac{1}{2}U_1(t), & \gamma_{111}(t) = U_0(t), & \gamma_{120}(t) = \frac{1}{2}U_1(t), & \gamma_{121}(t) = 0, \\ \gamma_{210}(t) = 0, & \gamma_{211}(t) = U_0(t), & \gamma_{220}(t) = \frac{1}{2}U_1(t), & \gamma_{221}(t) = U_0(t), \\ c_{10}(t) = 1, & c_{11}(t) = 0, & c_{20}(t) = 0, & c_{21}(t) = 1. \end{array}$$

Substituting these expansions into (19) and using (7)–(8), for $\nu = 1$, we obtain

$$\begin{cases} \beta_{11}U_0(t) + \beta_{12}U_1(t) - \frac{1}{2}\beta_{10}U_1(t) + \frac{1}{2}\beta_{11}U_0(t) + \frac{1}{2}\beta_{20}U_1(t) = U_0(t), \\ \beta_{21}U_0(t) + \beta_{22}U_1(t) + \frac{1}{2}\beta_{11}U_0(t) + \frac{1}{2}\beta_{20}U_1(t) + \frac{1}{2}\beta_{21}U_0(t) = U_1(t). \end{cases}$$

Then the linear independency of $\{U_0(t),\ U_1(t)\}$ implies

$$\begin{cases} \frac{3}{2}\beta_{11} = 1, \\ -\frac{1}{2}\beta_{10} + \beta_{12} + \frac{1}{2}\beta_{20} = 0, \\ \frac{1}{2}\beta_{11} + \frac{3}{2}\beta_{21} = 0, \\ \frac{1}{2}\beta_{20} + \beta_{22} = 1. \end{cases}$$

A nonunique solution of this system for the arbitrary values of β_{10} and β_{20} is given by

$$\beta_{10}, \quad \beta_{11} = \frac{2}{3}, \quad \beta_{12} = \frac{1}{2}\left(\beta_{10} - \beta_{20}\right), \quad \beta_{20}, \quad \beta_{21} = -\frac{2}{9}, \quad \beta_{22} = 1 - \frac{1}{2}\beta_{20}.$$

For example, if $\beta_{10} = \beta_{20} = 2$, then $\beta_{12} = \beta_{22} = 0$ and so we find from (20) that

$$\phi_1(\tau) = \frac{\frac{2}{3}\tau + 2}{\sqrt{1-\tau^2}}, \qquad \phi_2(\tau) = \frac{-\frac{2}{9}\tau + 2}{\sqrt{1-\tau^2}},$$

(see Figure 1 for the behavior of these solutions).

Example 2. Solve the problem of Example 1 in the case **3**.

In this case, we set

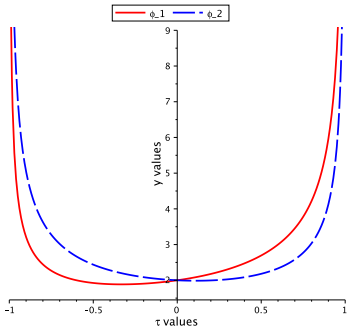


Figure 1: The plots of approximate solutions of Example 1 for M=2.

$$\phi_j(\tau) := \sqrt{\frac{1+\tau}{1-\tau}}\,\{\beta_{j0}\,V_0(\tau) + \beta_{j1}\,V_1(\tau)\}\ , \qquad j = 1, 2, \tag{21}$$

and

$$\begin{aligned} &K_{ij}(t,\tau) = \gamma_{ij0}(t)V_0(\tau) + \gamma_{ij1}(t)V_1(\tau)\,, && i,j = 1,2,\\ &f_i(t) = c_{i0}W_0(t) + c_{i1}W_1(t), && i = 1,2, \end{aligned}$$

where

$$\begin{aligned} &\gamma_{110}(t) = W_0(t) - \tfrac{1}{2}W_1(t), && \gamma_{111}(t) = \gamma_{210}(t) = \gamma_{211}(t) = \gamma_{221}(t) = \tfrac{1}{2}W_0(t),\\ &\gamma_{120}(t) = -\tfrac{1}{2}W_0(t) + \tfrac{1}{2}W_1(t), && \gamma_{121}(t) = 0, \qquad \gamma_{220}(t) = \tfrac{1}{2}W_1(t),\\ &c_{10}(t) = 1, \qquad c_{11}(t) = 0, \qquad c_{20}(t) = -1, \qquad c_{21}(t) = 1. \end{aligned}$$

Substituting these expansions into (19) and using (7)–(8) for $\nu = 3$, result

$$\begin{cases} \beta_{10}W_0(t) + \beta_{11}W_1(t) + \beta_{10}W_0(t) - \frac{1}{2}\beta_{10}W_1(t)\\ \qquad\qquad + \frac{1}{2}\beta_{11}W_0(t) - \frac{1}{2}\beta_{20}W_0(t) + \frac{1}{2}\beta_{20}W_1(t) = W_0(t),\\ \\ \beta_{20}W_0(t) + \beta_{21}W_1(t) + \frac{1}{2}\beta_{10}W_0(t) + \frac{1}{2}\beta_{11}W_0(t)\\ \qquad\qquad + \frac{1}{2}\beta_{20}W_1(t) + \frac{1}{2}\beta_{21}W_0(t) = -W_0(t) + W_1(t), \end{cases}$$

and from the linear independency of $\{W_0(t)$ and $W_1(t)\}$, we get the algebraic system

$$\begin{cases} 2\beta_{10} + \frac{1}{2}\beta_{11} - \frac{1}{2}\beta_{20} = 1,\\ -\frac{1}{2}\beta_{10} + \beta_{11} + \frac{1}{2}\beta_{20} = 0,\\ \frac{1}{2}\beta_{10} + \frac{1}{2}\beta_{11} + \beta_{20} + \frac{1}{2}\beta_{21} = -1,\\ \frac{1}{2}\beta_{20} + \beta_{21} = 1, \end{cases}$$

which has the unique solution

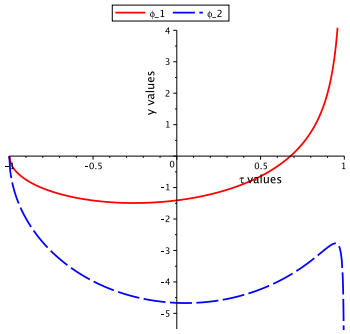


Figure 2: The plots of approximate solutions of Example 2 for M=2.

$$\beta_{10} = -\frac{10}{27}\,,\quad \beta_{11} = \frac{28}{27}\,,\quad \beta_{20} = -\frac{22}{9}\,,\quad \beta_{21} = \frac{20}{9}\,,$$

and the solutions of (19) can be found via (21). The graphs of these solutions plotted in Figure 2.

Example 3. Solve the following system in the case **3** [16]:

$$\begin{cases} 3\int_{-1}^{1} \frac{\phi_1(\tau)}{\tau - t}\, d\tau + \int_{-1}^{1} \frac{\phi_2(\tau)}{\tau - t}\, d\tau = (t^2+1)\sin 2t, \; -1 < t < 1, \\ 2\int_{-1}^{1} \frac{\phi_1(\tau)}{\tau - t}\, d\tau + \int_{-1}^{1} \frac{\phi_2(\tau)}{\tau - t}\, d\tau = t\cos 2t, \qquad -1 < t < 1. \end{cases}$$

In this case, we will approximate the unknown functions as

$$\phi_i(\tau) \simeq \varphi_i(\tau) := \sqrt{1-\tau^2} \sum_{j=0}^{M} \beta_{ij}\, U_j(\tau), \qquad i = 1, 2,$$

where the exact solutions are

$$\begin{cases} \phi_1(\tau) = -\frac{\sqrt{1-\tau^2}}{\pi^2} \int_{-1}^{1} \frac{(t^2+1)\sin 2t - t\cos 2t}{\sqrt{1-t^2}(t-\tau)}\, dt, \\ \phi_2(\tau) = -\frac{\sqrt{1-\tau^2}}{\pi^2} \int_{-1}^{1} \frac{3t\cos 2t - 2(t^2+1)\sin 2t}{\sqrt{1-t^2}(t-\tau)}\, dt. \end{cases}$$

We define the error functions as

$$E_i(\tau) = |\phi_i(\tau) - \varphi_i(\tau)|, \qquad i = 1, 2,$$

where the exact solutions ϕ_1 and ϕ_2 are calculated by using the Maple code `int(expression,x=a..b, CauchyPrincipalValue)`.

Table 1: Comparison of our results (E_i) with the results of [16] (ε_i) for M=8.

τ	$\phi_1(\tau)$	$\phi_2(\tau)$	$E_1(\tau)$	$E_2(\tau)$	$\varepsilon_1(\tau)$	$\varepsilon_2(\tau)$
-0.96	-0.2164694659571311	0.5107921932649533	1e$-$8	2e$-$8	8e$-$5	2e$-$4
-0.70	-0.5394637532569845	1.185306580331174	3e$-$7	6e$-$7	4e$-$5	1e$-$4
-0.25	-0.5831756914063946	1.129312022840245	4e$-$7	9e$-$7	9e$-$6	2e$-$5
0.25	-0.5831756914063946	1.129312022840245	4e$-$7	9e$-$7	9e$-$6	2e$-$5
0.75	-0.5394637532569845	1.185306580331174	3e$-$7	6e$-$7	4e$-$5	1e$-$4
0.96	-0.2164694659571311	0.5107921932649533	1e$-$8	2e$-$8	8e$-$5	2e$-$4

Table 2: The approximated values $\varphi_i(\tau)$ for $M = 15$ by using 16-digits arithmetic.

τ	$\varphi_1(\tau)$	$\varphi_2(\tau)$	$E_1(\tau)$	$E_2(\tau)$	$\varepsilon_1(\tau)$	$\varepsilon_2(\tau)$
-0.96	-0.2164694659571311	0.5107921932649527	0	6e$-$16	-	-
-0.70	-0.5394637532569844	1.185306580331173	1e$-$16	1e$-$15	-	-
-0.25	-0.5831756914063935	1.129312022840244	1e$-$15	1e$-$15	-	-
0.25	-0.5831756914063935	1.129312022840244	1e$-$15	1e$-$15	-	-
0.75	-0.5394637532569844	1.185306580331173	1e$-$16	1e$-$15	-	-
0.96	-0.2164694659571311	0.5107921932649527	0	6e$-$16	-	-

In Table 1, we compared our numerical results (absolute errors E_1 and E_2) with those reported in [16] (absolute errors ε_1 and ε_2). Table 2 shows our results for $M = 15$, which were not reported in [16].

Example 4. Consider the singular integral equation

$$\int_{-1}^{1} \frac{\phi(\tau)}{\tau - t}\, d\tau + \int_{-1}^{1} \mathbf{e}^{\mathbf{i}t}\phi(\tau)\, d\tau = 1 - 2t^2 + \mathbf{i}t + \frac{1}{2}\mathbf{e}^{-\mathbf{i}t}, \quad -1 < t < 1, \quad (22)$$

with the exact solution $\phi(\tau) = \frac{\sqrt{1-\tau^2}}{\pi}(2\tau - \mathbf{i})$ in the complex plane, where $\mathbf{i} = \sqrt{-1}$.

By taking $\phi_1 := Re\{\phi\}$ and $\phi_2 := Im\{\phi\}$, equation (22) is reduced to the system of singular integral equations

$$\begin{cases} \int_{-1}^{1} \frac{\phi_1(\tau)}{\tau - t}\, d\tau + \int_{-1}^{1} \cos(t)\phi_1(\tau)\, d\tau - \int_{-1}^{1} \sin(t)\phi_2(\tau)\, d\tau = 1 - 2t^2 + \frac{1}{2}\sin t, & -1 < t < 1, \\ \int_{-1}^{1} \frac{\phi_2(\tau)}{\tau - t}\, d\tau + \int_{-1}^{1} \sin(t)\phi_1(\tau)\, d\tau + \int_{-1}^{1} \cos(t)\phi_2(\tau)\, d\tau = t - \frac{1}{2}\cos t, & -1 < t < 1. \end{cases}$$

Similar to Examples 1 and 2, by setting

$$\phi_i(\tau) \simeq \varphi_i(\tau) := \sqrt{1-\tau^2} \sum_{j=0}^{M} \beta_{ij}\, U_j(\tau), \qquad i = 1, 2,$$

we get the exact solutions for $M = 1$.

Example 5. Consider the system of singular integral equations [14]

$$\begin{cases} \frac{1000}{\pi}\int_{-1}^{1}\frac{\phi_1(\tau)}{\tau-t}\,d\tau+\frac{10}{\pi}\int_{-1}^{1}\frac{\phi_2(\tau)}{\tau-t}\,d\tau=f_1(t)+\mathbf{i}g_1(t),\ -1<t<1,\\ \frac{500}{\pi}\int_{-1}^{1}\frac{\phi_1(\tau)}{\tau-t}\,d\tau+\frac{200}{\pi}\int_{-1}^{1}\frac{\phi_2(\tau)}{\tau-t}\,d\tau=f_2(t)+\mathbf{i}g_2(t),\ -1<t<1, \end{cases} \tag{23}$$

where

$$\begin{cases} f_1(t)=-990t^8+1089t^7+937t^6-\frac{26704}{25}t^5-\frac{349161}{1000}t^4+\frac{792327}{2000}t^3+\frac{1761}{250}t^2-\frac{53511}{4000}t-\frac{53929}{2000},\\ g_1(t)=990t^8-1189t^7-\frac{8971}{10}t^6+\frac{119047}{100}t^5+\frac{163961}{500}t^4-\frac{279198}{625}t^3-\frac{30873}{10000}t^2+\frac{69533}{4000}t+\frac{1130501}{40000},\\ f_2(t)=-300t^8+330t^7+215t^6-\frac{2447}{10}t^5-\frac{8607}{100}t^4+\frac{735}{8}t^3-\frac{17253}{2000}t^2+\frac{29541}{4000}t-\frac{14701}{1000},\\ g_2(t)=300t^8-380t^7-197t^6+\frac{1462}{5}t^5+\frac{9419}{100}t^4-\frac{27549}{250}t^3-\frac{183}{400}t^2-\frac{7957}{2000}t+\frac{57583}{4000}. \end{cases}$$

It is easy to see that system (23) is equivalent to the following disjointed singular integral equations,

$$\begin{cases} \frac{1}{\pi}\int_{-1}^{1}\frac{Re\{\phi_1(\tau)\}}{\tau-t}\,d\tau=\frac{20f_1(t)-f_2(t)}{19500},\ -1<t<1,\\ \frac{1}{\pi}\int_{-1}^{1}\frac{Im\{\phi_1(\tau)\}}{\tau-t}\,d\tau=\frac{20g_1(t)-g_2(t)}{19500},\ -1<t<1,\\ \frac{1}{\pi}\int_{-1}^{1}\frac{Re\{\phi_2(\tau)\}}{\tau-t}\,d\tau=\frac{2f_2(t)-f_1(t)}{390},\ -1<t<1,\\ \frac{1}{\pi}\int_{-1}^{1}\frac{Im\{\phi_2(\tau)\}}{\tau-t}\,d\tau=\frac{2g_2(t)-g_1(t)}{390},\ -1<t<1. \end{cases} \tag{24}$$

The exact solution of system (23) was reported in [14] for the case $\mathbf{r}=4$. By applying our method for the equation in (24), we get the real parts of unknown functions exactly but, for the imaginary parts, we obtain the error functions

$$E_i(\tau)=|Im\{\phi_i(\tau)\}-Im\{\varphi_i(\tau)\}|=\sqrt{\frac{1-\tau}{1+\tau}}h_i(\tau),\quad i=1,2,$$

where

$$\begin{cases} h_1(\tau)=5.128205128205128\times10^{-8}(1+\tau), & -1<\tau<1,\\ h_2(\tau)=5.128205128205128\times10^{-6}(1+\tau), & -1<\tau<1. \end{cases}$$

The plots of h_1 and h_2 and the regular parts of the error functions, are shown in Figure 3. As shown in these plots, the absolute errors increase when τ approaches to 1, because in case 4, the singularity for the integral part happens at $\tau=1$.

Example 6. Consider the problem of a half plane containing a crack parallel to the boundary, which is illustrated in Figure 4 and formulated as the system [4]

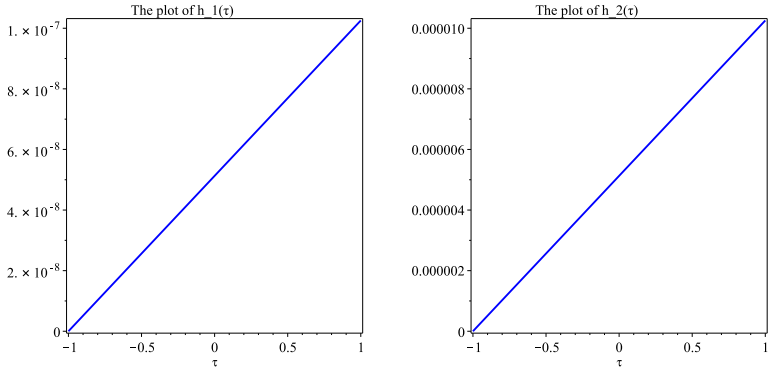


Figure 3: The plots of the regular parts of the error functions in Example 5.

$$\begin{cases} \int_{-1}^{1} \frac{\phi_1(\tau)}{\tau - t}\, d\tau + \int_{-1}^{1} \left[K_{11}(t,\tau)\phi_1(\tau) + K_{12}(t,\tau)\phi_2(\tau) d\tau\right] = 0, \\ \int_{-1}^{1} \frac{\phi_2(\tau)}{\tau - t}\, d\tau + \int_{-1}^{1} \left[K_{21}(t,\tau)\phi_1(\tau) + K_{22}(t,\tau)\phi_2(\tau) d\tau\right] = \pi, \end{cases} \tag{25}$$

with

$$K_{11}(t,\tau) = -\frac{\tau - t}{(\tau-t)^2+4h^2} + \frac{8h^2(\tau-t)}{[(\tau-t)^2+4h^2]^2} - \frac{4h^2(\tau-t)\left[12h^2-(\tau-t)^2\right]}{[(\tau-t)^2+4h^2]^3},$$

$$K_{12}(t,\tau) = K_{21}(t,\tau) = -\frac{8h^3\left[4h^2-3(\tau-t)^2\right]}{[(\tau-t)^2+4h^2]^3},$$

$$K_{22}(t,\tau) = -\frac{\tau - t}{(\tau-t)^2+4h^2} - \frac{8h^2(\tau-t)}{[(\tau-t)^2+4h^2]^2} - \frac{4h^2(\tau-t)\left[12h^2-(\tau-t)^2\right]}{[(\tau-t)^2+4h^2]^3},$$

where h is the distance of crack from the boundary. The physical conditions of the problem impose that the relations

$$\int_{-1}^{1} \phi_1(\tau)\, d\tau = 0\,, \qquad \int_{-1}^{1} \phi_2(\tau)\, d\tau = 0, \tag{26}$$

and

$$\phi_1(t) = \phi_1(-t), \quad \phi_2(t) = -\phi_2(-t)$$

are satisfied. Therefore the unknown functions may be expressed as

$$\phi_1(\tau) \simeq \frac{1}{\sqrt{1-\tau^2}} \sum_{j=0}^{M} \beta_{1j}\, T_{2j}(\tau), \quad \phi_2(\tau) \simeq \frac{1}{\sqrt{1-\tau^2}} \sum_{j=1}^{M} \beta_{2j}\, T_{2j-1}(\tau). \tag{27}$$

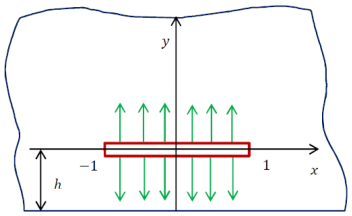


Figure 4: Crack parallel to a free boundary in Example 6.

For $\nu = 1$, it follows from the orthogonality condition (7) that the second condition in (26) satisfies and the first one gives $\beta_{10} = 0$.

Taking $\beta_{11} = \beta_{21} = 0$, the remaining coefficients β_{ij} are uniquely determined from the linear algebraic system (14) for each values of h and M. This leads to find the functions ϕ_1 and ϕ_2 from (27).

The stress intensity factors

$$k_1 = \lim_{\tau \to 1^-} \sqrt{1-\tau^2}\, \phi_2(\tau)$$
$$k_2 = \lim_{\tau \to 1^-} \sqrt{1-\tau^2}\, \phi_1(\tau)$$

and their absolute estimation errors(Est.Err.) are reported in Table 3. For $h = \infty$ and $K_{ij}(t,\tau) = 0$, from (27) and (13)–(14), the exact solutions of (25) are obtained as

$$\phi_1(\tau) = 0, \qquad \phi_2(\tau) = \frac{\tau}{\sqrt{1-\tau^2}},$$

which give $k_1 = 1$ and $k_2 = 0$. This is shown in the last row of Table 3. The table shows the rapid convergence of the results even for relatively small values of M.

4 Conclusions

We described a new idea of using Chebyshev polynomials for the numerical solution of the system of singular integral equations of the first kind. In Section 3, we illustrated this idea by using system of different kind of singular integral equations (Examples 1–5). In Example 6, we studied a crack problem in solid mechanics and reported the numerical results (see Table 3) to show the efficiency and rapid convergence of the proposed method for all these kinds of problems.

Table 3: Stress intensity factors for the crack parallel to the boundary

h	M	k_1	Est.Err. k_1	k_2	Est.Err. k_2
0.2	6	4.878800637605022	6.3e-14	1.750099102171126	6.2e-14
	7	4.788277537335018	1.1e-14	1.727809740547429	4.1e-15
	8	4.760729834685963	4.8e-14	1.719782910590219	1.1e-14
0.4	3	2.607272141646415	4.3e-15	0.7745787927510580	1.6e-16
	4	2.594500911475041	7.3e-15	0.7266641783709941	5.0e-15
	6	2.594423234973139	4.2e-14	0.7376171346942053	3.3e-16
0.6	2	1.834057544899021	1.1e-15	0.5664257041432605	4.1e-16
	5	1.960455689663461	6.1e-15	0.4297949760368867	1.6e-15
0.8	2	1.608371955353828	1.6e-15	0.3323260582700188	2.8e-16
	3	1.660617572058080	8.7e-16	0.2675691556476836	4.6e-16
1.0	2	1.461157081431933	2.0e-15	0.2104682299562445	1.1e-16
	4	1.485914720666516	2.1e-16	0.1796691052492212	1.1e-16
1.2	4	1.372176156193755	5.0e-16	0.1234414146531335	0.
1.5	4	1.262800608570183	1.6e-15	0.07465158121522054	1.7e-16
2.0	3	1.162112249974693	1.1e-15	0.03662808437088003	0.
3.0	2	1.077621553329114	3.3e-16	0.01274529646673066	0.
10	2	1.007451045420713	2.6e-16	0.00037197952964307	0.
∞	1	1	0	0	0

References

1. Abdolkawi, M. *Solution of Cauchy type singular integral equations of the first kind by using differential transform method*, Appl. Math, Model. 39 (2015), 2107–2118.

2. Ahdiaghdam, S., Shahmorad S. and Ivaz, K. *Approximate solution of dual integral equations using Chebyshev polynomials*, Int. J. Comput. Math. 94(3) (2017), 493–502.

3. Chakrabarti, A. and Berghe, G.V. *Approximate solution of singular integral equations*, Appl. Math. Lett., 17 (2004), 533–559.

4. Erdogan, F., Gupta, G.D. and Cook, T.S. *Mechanics of fracture*, Noordhoff International Publishing, Leyden, Netherlands, 1973.

5. Eshkuvatov, Z.K., Nik Long, N.M.A. and Abdulkawi, M. *Approximate solution of singular integral equations of the first kind with Cauchy kernel*, Appl. Math. Lett., 22 (2009), 651–657.

6. Karczmarek, P., Pylak, D. and Sheshko, M.A. *Application of Jacobi polynomials to approximate solution of a singular integral equation with Cauchy kernel*, Appl. Math. Comput., 181 (2006), 694–707.

7. Kashfi, M. and Shahmorad, S. *Approximate solution of a singular integral Cauchy-kernel equation of the first kind*, Comput. Methods Appl. Math. 10(4) (2010), 345–353.

8. Liu, D., Zhang, X. and Wu, J. *A collocation scheme for a certain Cauchy singular integral equation based on the superconvergence analysis*, Appl. Math. Comput. 219 (2013), 5198–5209.

9. Mandal, B.N. and Chakrabarti, A. *Applied singular integral equations*, Taylor and Francis Group, CRC Press, 2011.

10. Mason, J.C. and Handscomb, D.C. *Chebyshev polynomials*, Chapman and Hall, CRC Press, 2003.

11. Miel, G. *On the Galerkin and collocation methods for a Cauchy singular integral equation*, SIAM J. Numer. Anal. 23(1) (1986), 135–143.

12. Muskhelishvili, N.I. and RADOK, J.R.M. *Singular integral equations*, Wolters-Noordhoff Publishing Groningen, Netherlands, 1958.

13. Setia, A. *Numerical solution of various cases of Cauchy type singular integral equation*, Appl. Math. Comput. 230 (2014), 200–207.

14. Sharma, V., Setia, A. and Agarwal, R. *Numerical solution for system of Cauchy type singular integral equations with its error analysis in complex plane*, Appl. Math. Comput., 328 (2018), 338–352.

15. Tsalamengas, J.L. *A direct method to quadrature rules for a certain class of singular integrals with logarithmic, Cauchy, or Hadamard-type singularities*, Int. J. Numer. Model. 25 (2012), 512–524.

16. Turhan, I., Oguz, H. and Yusufoglu, E. *Chebyshev polynomial solution of the system of Cauchy-type singular integral equations of the first kind*, Int. J. Comput. Math. 90(5) (2013), 944–954.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 49–75
DOI:10.22067/ijnao.v9i2.73622

Research Article

A discrete orthogonal polynomials approach for fractional optimal control problems with time delay

F. Mohammadi*

Abstract

An efficient direct and numerical method has been proposed to approximate a solution of time-delay fractional optimal control problems. First, a class of discrete orthogonal polynomials, called Hahn polynomials, has been introduced and their properties are investigated. These properties are employed to derive a general formulation of their operational matrix of fractional integration, in the Riemann–Liouville sense. Then, the fractional derivative of the state function in the dynamic constraint of time-delay fractional optimal control problems is approximated by the Hahn polynomials with unknown coefficients. The operational matrix of fractional integration together with the dynamical constraints is used to approximate the control function directly as a function of the state function. Finally, these approximations were put in the performance index and necessary conditions for optimality transform the under consideration time-delay fractional optimal control problems into an algebraic system. Some illustrative examples are given and the obtained numerical results are compared with those previously published in the literature.

AMS(2010): 34A08; 65M70; 33C45; 49J21.

Keywords: Delay fractional optimal control problems; Riemann–Liouville integration; Hahn polynomials; Operational matrix.

1 Introduction

It has been found that integer-order calculus is not an appropriate tool for modeling complex systems in science and engineering. Fractional calculus, as an extension of derivatives and integrals to noninteger orders, has been recently used to model many fundamental problems; see [32, 38]. Recently,

*Corresponding author
Received 19 June 2018; revised 28 December 2018; accepted 16 February 2019
Fakhrodin Mohammadi Department of Mathematics, Faculty of Sciences, University of Hormozgan, Bandar Abbas, Iran. e-mail: f.mohammadi62@hotmail.com

fractional differential equations have found many applications in various fields of engineering and physics such as colored noise, signal processing, electromagnetism, electrochemistry, dynamic of viscoelastic materials, continuum and statistical mechanics, solid mechanics, and fluid-dynamic traffic model; see [3, 4, 9, 12, 40].

Contrary to ordinary differential equations (ODEs), delay differential equations (DDEs) contain derivatives that depend on the solution at previous times, thus making the mathematical model closer to the real-world phenomenon; see [26,27]. Recently, many researches have been focused on DDEs and their numerical solution. For example, the finite difference method [22], Adams methods [41], Adomian decomposition method [13], variational iteration method [45], hybrid functions [29], spectral methods [19, 39, 44], and wavelet methods [35, 36] have been employed to approximate solution of DDEs.

Over the last decades, numerical methods based on orthogonal functions have been frequently used to approximate solution of differential and integral equations [7, 28, 31, 34, 39]. The main characteristic of these methods is that they reduce under consideration problems to those of solving a system of algebraic equations, thus greatly simplifying the problems. Depending on their structure, the orthogonal functions are classified into three families: the first class consists of sets of piecewise constant orthogonal functions such as the Walsh functions and block pulse functions. The second class consists of sets of sine-cosine functions, and the last class is the most widely used orthogonal polynomials, such as Laguerre, Legendre, and Chebyshev polynomials; see [28,31]. It is well known that orthogonal polynomials that are solutions of singular Sturm–Liouville problems allow approximation of smooth functions, where truncation error approaches zero faster than any negative power of the number of basic functions used in the approximation. This phenomenon is usually referred to as spectral accuracy; see [8, 30]. Moreover, according to the defined inner product in the solution space, orthogonal polynomials are classified into two main classes: continuous and discrete. For continuous orthogonal polynomials such as Legendre, Chebyshev, Hermite, and Laguerre, one has to evaluate an integral in the inner product, whereas discrete orthogonal polynomials come with a discrete scalar product and hence the integral becomes a sum. Although continuous orthogonal polynomials have been more frequently used to approximate solution of functional equations, there are some advantages of using discrete orthogonal polynomials. For example, by using discrete orthogonal polynomials, the Fourier coefficients can be calculated with the aid of a summation and the obtained coefficients are exact. Consequently, comparison to continuous cases, implementation of discrete orthogonal polynomials is more efficient and less complex; see [16, 43]. Moreover, there is a close connection between stochastic processes and discrete orthogonal polynomials that motivated many researchers to consider them to solve stochastic differential equations; see [2, 43].

The optimal control of a system is one of the most practical subjects in science and engineering [5, 6, 17, 21, 25, 33]. The optimal control problems involve minimization of a performance index subject to a dynamical system. As generalizations of the classical optimal control, FOCPs are problems in which fractional derivatives or integrals are used in the performance index or constraints. Similar to the other types of fractional functional equations, most of FOCPs do not have exact and analytic solutions [7, 34, 35, 37]. Furthermore, a time-delay fractional optimal control problem (TDFOCP) is a fractional optimal control problem in which the dynamic of system contains some time-delay equations. The control of time-delay systems frequently in electronic, age structure, biological, chemical, electronic, and transportation systems [14, 20, 23, 41]. Due to their variety of applications in the realistic models of phenomena, the control of time-delay systems has been investigated by many engineers and scientists. Also, considerable attention has been focused on the approximate and numerical solution of them [7,34,35,37]. Generally, there are two main approaches to the approximate solution of TDFOCPs. The first one is based on constructing a Hamiltonian system and then solve the arising two-point boundary value problem, which is the indirect approach, and the other involves facing directly the problems by discretizing or approximating the functions without constructing the Hamiltonian equations. Recently, more attention has been done to direct approaches [7,22,35].

The main purpose of this work is to introduce a new kind of discrete orthonormal polynomials. Some properties of these discrete polynomials are studied and a general formulation of their operational matrix of fractional integration, in the Riemann–Liouville sense, is derived. Make use of these orthogonal polynomials and their operational matrix, an efficient direct numerical method is proposed to approximate the solution of the following TDFOCP:

$$Min\ J = \frac{1}{2}\int_0^1 \left[r(t)x^2(t) + s(t)u^2(t)\right]dt, \tag{1}$$

subjected to

$$D^{\nu}x(t) = a(t)\,x(t) + b(t)\,u(t) + c(t)\,x(t-\tau) + e(t)\,u(t-\eta) + f(t), \tag{2}$$

$$t \in [0,1],\ 0 < \nu \leq 1, \tag{3}$$

$$\begin{aligned} x(t) &= g_1(t), \quad t \in [-\tau, 0], \\ u(t) &= g_2(t), \quad t \in [-\eta, 0], \end{aligned} \tag{4}$$

in which $x(t)$ and $u(t)$ are the state and control functions, respectively. Furthermore, $r(t)$ and $s(t)$ are positive functions and $a(t)$, $b(t)$, $c(t)$, $e(t)$, and $f(t)$ are continuous functions, $g_1(t)$ and $g_2(t)$ are arbitrary known functions defined on the intervals $[-\tau, 0]$ and $[-\eta, 0]$, respectively. Also, τ and $\eta > 0$

are given constant delays and $D^{\nu}x\,(t)$ is the fractional derivative of state function $x(t)$ in the Caputo sense.

This paper is organized as follows: In Section 2, some basic definitions and preliminary remarks on the fractional calculus are presented. The Hahn polynomials and their properties are introduced in Section 3. The operational matrix of fractional integration and product operational matrix for the Hahn polynomials have been derived in Section 4. In Section 5, an efficient direct method is proposed to solve TDFOCPs. To confirm the efficiency and accuracy of the presented method, some illustrative examples are given in Section 6. Finally, a conclusion is given in Section 7.

2 Definitions and preliminaries

Although, the fractional order operators enable to model a wider class of problems, there does not exist a unique definition of fractional derivative. Among the several formulations of the generalized derivative, the Riemann–Liouville and Caputo definitions are the most commonly used, which can be described as follows [32,38].

Definition 1. A real function $f(t), t > 0$, is said to belong the space $C_{\mu}, \mu \in \mathbb{R}$ if there exist a real number $p > \mu$ and a function $f_1(t) \in C[0,\infty)$ such that $f(t) = t^p f_1(t)$. Moreover, if $f^{(n)} \in C_{\mu}$, then $f(t)$ is said to belong the space $C_{\mu}^{n},\ n \in \mathbb{N}$.

Definition 2. The Riemann–Liouville fractional integration of order $\nu \geq 0$ of a function $f \in C_{\mu}, \mu \geq -1$, is defined as

$$\left(I^{\nu} f\right)(t) = \begin{cases} \dfrac{1}{\Gamma(\nu)} \displaystyle\int_0^t (t-\tau)^{\nu-1} f(\tau) d\tau, & \nu > 0, \\ f(t), & \nu = 0. \end{cases}$$

The Riemann–Liouville fractional operator I^{ν} has the following properties:

$$I^{\nu_1}\left(J^{\nu_2} f(t)\right) = I^{\nu_2}\left(I^{\nu_1} f(t)\right), \quad \nu_1, \nu_2 \geq 0,$$

$$I^{\nu_1}\left(I^{\nu_2} f(t)\right) = I^{\nu_1+\nu_2} f(t), \quad \nu_1, \nu_2 \geq 0,$$

$$I^{\nu} t^{\lambda} = \frac{\Gamma(\lambda+1)}{\Gamma(\lambda+\nu+1)} t^{\nu+\lambda}, \quad \nu \geq 0, \lambda > -1.$$

Definition 3. The Caputo fractional derivative of order $\nu > 0$ is defined as

$$\mathcal{D}^{\nu} f(t) = \begin{cases} \dfrac{d^n f(t)}{dt^n}, & \nu = n \in \mathbb{N}, \\ \dfrac{1}{\Gamma(n-\nu)} \displaystyle\int_0^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\nu-n+1}} d\tau, & t > 0,\ 0 \le n-1 < \nu < n, \end{cases}$$

where n is integer, $t > 0$, and $f \in C_1^n$.

For $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$, $f \in C_\mu$, $\mu, \lambda \ge -1$, and $n-1 < \nu \le n$, some useful and practical properties of the Caputo fractional operators $\mathcal{D}^{\nu}$ are given by the following expressions:

$$I^{\nu} D^{\nu} f(t) = f(t) - \sum_{k=0}^{n-1} f^{(k)}(0^+) \frac{t^k}{k!}, \quad t > 0,$$

$$\mathcal{D}^{\nu} I^{\nu} f(t) = f(t),$$

$$\mathcal{D}^{\nu} t^{\lambda} = \begin{cases} 0 & for\ \lambda \in \mathbb{N}_0\ and\ \lambda < \nu, \\ \dfrac{\Gamma(\lambda+1)}{\Gamma(\lambda-\nu+1)} t^{\lambda-\nu} & otherwise. \end{cases}$$

For more details on fractional calculus and their applications we refer the reader to [32, 38].

3 Hahn polynomials and their properties

The Hahn polynomials are a class of orthogonal polynomials, which are introduced and investigated by Wolfgang Hahn [18]. These polynomials can be considered as a discrete analog of the classical Jacobi polynomials [16, 42]. This section is devoted to a brief definition of Hahn polynomials and their properties. For more details and some applications of the Hahn polynomials, one can refer the reader to [16, 18, 24, 42, 43].

Definition 4. For arbitrary complex numbers a_i, $i = 1, 2, 3$ and $b_j \ne 0$, $j = 1, 2$, the generalized hypergeometric series ${}_3F_2\left(a_1, a_2, a_3; b_1, b_2; x\right)$ is defined as

$${}_3F_2\left(a_1, a_2, a_3; b_1, b_2; x\right) = \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k (a_3)_k}{(b_1)_k (b_2)_k} \frac{x^k}{k!}$$

in which a can be derived as follows:

$$(a)_0 = 1,$$
$$(a)_k = a(a+1)\cdots(a+k-1), \quad k \geq 1,$$

and the series terminates if one of the numbers a_i is zero or a negative integer.

Definition 5. Let $\alpha, \beta > -1$ be real numbers and let N be a positive integer number. The Hahn polynomial $H_n(t;\alpha,\beta,N)$ may be defined in terms of a hypergeometric series

$$\begin{aligned} Q_n(x;\alpha,\beta,N) &= {}_3F_2(-n, n+\alpha+\beta+1, -x; \alpha+1, -N; 1) \\ &= \sum_{k=0}^{n} \frac{(-n)_k (n+\alpha+\beta+1)_k (-x)_k}{k!(\alpha+1)_k (-N)_k}, \quad n = 0, 1, \ldots, N. \end{aligned}$$

Remark 1. Hereafter, the simple notation $Q_n(x)$ is used to denote the nth Hahn polynomial $Q_n(x;\alpha,\beta,N)$.

Orthogonality:
The set of Hahn polynomials $\{Q_n(x) : \ n = 0, 1, \ldots, N\}$ are orthogonal on the interval $[0, N]$ with respect to the following discrete norm:

$$\langle f, g \rangle_w = \sum_{r=0}^{N} f(r) g(r) w(r),$$

where $w(t)$ is the weight function and is defined by

$$\begin{aligned} w(x) &= \binom{\alpha+x}{x} \binom{\beta+N-x}{N-x} \\ &= \frac{\Gamma(\alpha+x+1)\,\Gamma(\beta+N-x+1)}{\Gamma(x+1)\,\Gamma(N-x+1)\,\Gamma(\alpha+1)\,\Gamma(\beta+1)}. \end{aligned}$$

Moreover, the orthogonality condition for these polynomials reads as follows:

$$\langle Q_n(x), Q_m(x) \rangle_w = \sum_{r=0}^{N} Q_n(r) Q_m(r) w(r) = \pi(n)\, \delta_{mn},$$

where δ_{mn} is the Kronecker delta and $\pi(n)$ can be defined as

$$\pi(n) = \frac{(-1)^n (n+\alpha+\beta+1)_{N+1} (\beta+1)_n\, n!}{(2n+\alpha+\beta+1)(\alpha+1)_n (-N)_n\, N!}.$$

Recurrence relation:
The set of Hahn polynomials $\{Q_n(x) : n = 0, 1, \ldots, N\}$ can be determined with the aid of the following recurrence formulas:

$$\mu_n Q_{n+1}(x) = (\mu_n + \sigma_n - x)\, Q_n(x) - \sigma_n Q_{n-1}(x), \quad n = 0, 1, \ldots, N-1,$$

where

$$\mu_n = \frac{(n+\alpha+\beta+1)(n+\alpha+1)(N-n)}{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)},$$

and

$$\sigma_n = \frac{n(n+\beta)(n+\alpha+\beta+N)}{(2n+\alpha+\beta)(2n+\alpha+\beta+1)}.$$

Explicit formula:
By using the generalized hypergeometric series ${}_3F_2$, the explicit analytical form of the Hahn polynomials can be derived as follows:

$$Q_n(x) = \sum_{k=0}^{n} h_{k,n} x^k, \quad n = 0, 1, \ldots, N,$$

in which

$$h_{k,n} = \sum_{i=k}^{n} \frac{(-1)^i (-n)_i (n+\alpha+\beta+1)_i S(k,i)}{i!(\alpha+1)_i (-N)_i}, \tag{5}$$

and

$$S(k,i) = \frac{1}{i!} \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} j^k.$$

Shifted Hahn polynomial:
In order to use the Hahn polynomials on the interval $[0,1]$, we define the shifted Hahn polynomials (SHPs) by introducing the change of variable $x = Nt$. Let the nth shifted Hahn polynomial, $Q_n(Nt)$, be denoted by $H_n(t)$. Then, the set of shifted Hahn polynomials $\{H_n(t) : n = 0, 1, \ldots, N\}$ are orthogonal on the interval $[0,1]$ with respect to the shifted weight function $\bar{w}(t) = w(Nt)$ and the following discrete norm:

$$\langle f, g \rangle_{\bar{w}} = \sum_{r=0}^{N} f\left(\frac{r}{N}\right) g\left(\frac{r}{N}\right) \bar{w}\left(\frac{r}{N}\right). \tag{6}$$

Also, the orthogonality condition for SHPs can be defined as follows:

$$\langle H_n(t), H_m(t) \rangle_{\bar{w}} = \sum_{r=0}^{N} H_n\left(\frac{r}{N}\right) H_m\left(\frac{r}{N}\right) \bar{w}\left(\frac{r}{N}\right) = \pi(n)\, \delta_{mn}, \tag{7}$$

where $\pi(n)$ is defined by (10).

Function approximation:
A function $f(t)$, defined over the interval $[0,1]$, may be expanded by the SHPs

as

$$f(t) \simeq \sum_{i=0}^{N} c_i H_i(t) = C^T \Phi(t), \tag{8}$$

where C and $\Phi(x)$ are $(N+1)$ vectors given by

$$C^T = [c_0, c_1, \ldots, c_N], \quad \Phi(t) = [H_0(t), H_1(t), \ldots, H_N(t)]^T, \tag{9}$$

and the coefficients c_i can be derived as follows:

$$c_i = \frac{\langle f(t), H_i(t) \rangle_{\bar{w}}}{\pi(i)}, \quad i = 0, 1, \ldots, N. \tag{10}$$

Convergence analysis:

Convergence analysis and spectral accuracy of the Hahn polynomials expansion are investigated thoroughly in [15,16]. The following Theorem provides conditions that ensure that the series expansion of a function by the discrete Hahn polynomials converges.

Theorem 1. *The series expansion* $\sum_{k=0}^{n} \frac{\langle f, Q_k \rangle}{\langle Q_k, Q_k \rangle} Q_k(x)$ *of a function* f *by the discrete Hahn polynomials converges pointwise, if the series expansion of the function* f *by the Jacobi polynomials converges pointwise and if* $\frac{n^{3+\alpha+max\{1,\alpha\}}}{N} \to 0$ *as* $n, N \to \infty$.

Proof. The proof follows directly from [15, Theorems 1.1, 2.1 and 2.2]. □

4 Operational matrices

In this section, the operational matrix of fractional integration in the Riemann–Liouville sense and product operational matrix for the SHPs vector $\Phi(t)$ will be derived. To this end, the inner product of the SHP $H_n(t)$ and t^s is derived explicitly. Then, by using this explicit formulation, the operational matrix of fractional integration and product operational matrix can be obtained easily.

Lemma 1. *Suppose that, for a positive real number* s*, the inner product of the* n*th Hahn polynomial* $H_n(t)$ *and* t^s *is denoted by* $\lambda(n, s)$*. It can be derived as*

$$\begin{aligned} \lambda(n,s) &= \langle H_n(t), t^s \rangle_{\bar{w}} \\ &= \frac{1}{N^s} \sum_{r=0}^{N} \sum_{k=0}^{n} \frac{h_{k,n} r^{s+k} \Gamma(\alpha + r + 1) \Gamma(\beta + N - r + 1)}{\Gamma(r+1) \Gamma(N - r + 1) \Gamma(\alpha + 1) \Gamma(\beta + 1)}, \end{aligned}$$

in which $h_{k,n}$ is defined in relation (5).

Proof. From the definition of the discrete inner product $\langle .,. \rangle_{\bar{w}}$ in (7), we have

$$\begin{aligned}\lambda(n,s) &= \sum_{r=0}^{N} H_n\left(\frac{r}{N}\right)\frac{r^s}{N^s}\bar{w}\left(\frac{r}{N}\right) = \frac{1}{N^s}\sum_{r=0}^{N} Q_n(r)\, r^s w(r) \\ &= \frac{1}{N^s}\sum_{r=0}^{N}\left(\sum_{k=0}^{n} h_{k,n} r^k\right) r^s \binom{r+\alpha}{r}\binom{\beta+N-r}{N-r} \\ &= \frac{1}{N^s}\sum_{r=0}^{N}\sum_{k=0}^{n} \frac{h_{k,n} r^{s+k}\Gamma(\alpha+r+1)\,\Gamma(\beta+N-r+1)}{\Gamma(r+1)\,\Gamma(N-r+1)\,\Gamma(\alpha+1)\,\Gamma(\beta+1)}. \end{aligned} \tag{11}$$

□

Theorem 2. *Let $\Phi(t)$ be the $(N+1)$ SHPs vector defined in (9). The fractional integration of order ν for this vector can be expressed as*

$$I^{\nu}\Phi(t) \simeq P^{(\nu)}\Phi(t), \tag{12}$$

where $P^{(\nu)}$ is an $(N+1)\times(N+1)$ matrix and its (i,j)th element can be defined as

$$P^{(\nu)}_{i,j} = \sum_{k=0}^{i-1} \frac{h_{k,i-1} N^k \Gamma(k+1)\,\lambda(j,k+\nu)}{\Gamma(k+\nu+1)\,\pi(j)}, \quad i,j = 0,1,\ldots,N.$$

Proof. The ith element of the vector $\Phi(t)$ is $H_{i-1}(t)$ and its fractional integral of order ν can be derived as

$$\begin{aligned} I^{\nu}H_{i-1}(t) &= I^{\nu}Q_{i-1}(Nt) \\ &= I^{\nu}\left(\sum_{k=0}^{i-1} h_{k,i-1} N^k t^k\right) = \sum_{k=0}^{i-1} \frac{h_{k,i-1} N^k \Gamma(k+1)}{\Gamma(k+\nu+1)} t^{\nu+k}. \end{aligned} \tag{13}$$

By expanding $t^{k+\nu}$ in terms of SHPs, we get

$$t^{k+\nu} \simeq \sum_{j=0}^{N} \beta_{j,k} H_j(t), \tag{14}$$

in which $\beta_{r,j}$ can be derived by using Lemma 1 as follows:

$$\beta_{j,k} = \frac{1}{\pi(j)}\left\langle H_j(t), t^{k+\nu}\right\rangle_{\bar{w}} = \frac{\lambda(j,k+\nu)}{\pi(j)}. \tag{15}$$

Now, by inserting (14) and (15) in (13), we get

$$I^{\nu}\Phi_i(t) \simeq \sum_{j=0}^{N}\left(\sum_{k=0}^{i-1}\frac{h_{k,i-1}N^k\Gamma\left(k+1\right)\lambda\left(j,k+\nu\right)}{\Gamma\left(k+\nu+1\right)\pi\left(j\right)}\right)H_j(t),$$

and this leads to the desired results. □

Theorem 3. *Let $\Phi(t)$ be the $(N+1)$ Hahn polynomials vector defined in (9) and let V be an arbitrary $(N+1)$ vector. Then, it is convenient to write*

$$\Phi(t)\Phi^T(t)V = \tilde{V}\Phi(t), \tag{16}$$

where $\tilde{V}$ is the $(N+1)\times(N+1)$ product operational matrix and its (i,j)th element can be defined as

$$\tilde{V}_{i+1,j+1} = \frac{1}{\pi\left(j\right)}\sum_{k=0}^{N}V_k\left\langle H_k(t)H_i(t),H_j(t)\right\rangle_{\bar{w}},\quad i,j=0,1,\ldots,N.$$

Proof. The product of two Hahn polynomial vectors $\Phi(t)$ and $\Phi^T(t)$ is an $(N+1)\times(N+1)$ matrix as follows

$$\Phi(t)\Phi^T(t) = \begin{bmatrix} H_0(t)H_0(t) & H_0(t)H_1(t) & \ldots & H_0(t)H_N(t)\\ H_1(t)H_0(t) & H_1(t)H_1(t) & \ldots & H_1(t)H_N(t)\\ \vdots & \vdots & \ddots & \vdots\\ H_N(t)H_0(t) & H_N(t)H_1(t) & \ldots & H_N(t)H_N(t)\end{bmatrix}.$$

As a result, relation (16) can be rewritten as:

$$\sum_{k=0}^{N}H_k(t)H_i(t)V_{k+1} = \sum_{k=0}^{N}H_k(t)\tilde{V}_{i+1,k+1},\quad i=0,1,\ldots,N.$$

Now, by multiplying both sides of the above equation by $H_j(t)$ and using the defined inner product in (6), we get

$$\sum_{k=0}^{N}\left\langle H_k(t)H_i(t),H_j(t)\right\rangle_{\bar{w}}V_{k+1} = \left\langle H_j(t),H_j(t)\right\rangle_{\bar{w}}\tilde{V}_{i+1,j+1},\quad i,j=0,1,\ldots,N.$$

Therefore, the matrix $\tilde{V}$ may be given by

$$\tilde{V}_{i+1,j+1} = \frac{\sum\limits_{k=0}^{N}V_k\left\langle H_k(t)H_i(t),H_j(t)\right\rangle_{\bar{w}}}{\left\langle H_j(t),H_j(t)\right\rangle_{\bar{w}}} = \frac{1}{\pi\left(j\right)}\sum_{k=0}^{N}V_k\left\langle H_k(t)H_i(t),H_j(t)\right\rangle_{\bar{w}}.$$

□

5 Description of the proposed method

In this section, SHPs along with their operational matrix of fractional integration will be applied to solve TDFOCPs. Consider the optimal control of time-varying linear system (1)–(4) with delays in state and control and with quadratic performance. First, we approximate all functions involved in the problem as follows:

$$\mathcal{D}^{\nu}x(t) \simeq X^{T}\Phi(t), \tag{17}$$

$$u(t) \simeq U^{T}\Phi(t), \tag{18}$$

$$r(t) \simeq R^{T}\Phi(t),\ s(t) \simeq S^{T}\Phi(t), \tag{19}$$

$$a(t) \simeq A^{T}\Phi(t),\ c(t) \simeq C^{T}\Phi(t), \tag{20}$$

$$b(t) \simeq B^{T}\Phi(t),\ e(t) \simeq E^{T}\Phi(t), \tag{21}$$

$$f(t) \simeq F^{T}\Phi(t),\ g_2(t) \simeq G_1^{T}\Phi(t),\ g_1(t) \simeq G_2^{T}\Phi(t), \tag{22}$$

where X and U are unknown vectors to be determined, while R, S, A, C, B, E, G_1, G_2, and F are known SHPs coefficient vectors that can be computed as explained in (8)–(10). By applying the fractional Riemann–Liouville operator I^{ν} on both sides of the relation (17), we have

$$x(t) \simeq I^{\nu}X^{T}\Phi(t) + g_1(0) = X^{T}P^{(\nu)}\Phi(t) + G_1^{T}\Phi(0), \tag{23}$$

in which $P^{(\nu)}$ is the operational matrix of fractional integration of the SHPs vector $\Phi(t)$ derived in Theorem 2. Hence, by using (23) and (18), the delay state function $x(t-\tau)$ and control function $u(t-\eta)$ can be derived as

$$x\left(t-\tau\right) = \begin{cases} G_1^{T}\Phi\left(t-\tau\right), & 0 \le t \le \tau, \\ \left(X^{T}P^{(\alpha)} + d^{T}\right)\Phi(t-\tau),\ \tau \le t \le 1, \end{cases}$$

and

$$u\left(t-\eta\right) = \begin{cases} G_2^{T}\Phi\left(t-\eta\right), & 0 \le t \le \eta, \\ U^{T}\Phi(t-\eta),\ \eta \le t \le 1. \end{cases}$$

These delay functions can be expanded by the Hahn polynomials as

$$x\left(t-\tau\right)=X^{T}P^{(\nu)}\Omega_{\tau}\Phi\left(t\right)+V_{\tau}^{T}\Phi\left(t\right), \tag{24}$$

and

$$u\left(t-\eta\right)=U^{T}\Omega_{\eta}\Phi\left(t\right)+V_{\eta}^{T}\Phi\left(t\right), \tag{25}$$

where V_{τ} and V_{η} are known $(N+1)$ vectors, whereas Ω_{η} and Ω_{η} are known $(N+1)\times(N+1)$ matrices, which are depend on the delay parameters τ and η, respectively (please, see details in Appendix A). Now, by substituting (17)–(25) in dynamical system (4), we get

$$\begin{aligned}X^{T}\Phi(t)=&X^{T}P^{(\nu)}\Phi(t)\Phi^{T}(t)A+G_{1}^{T}\Phi(0)A^{T}\Phi(t)+U^{T}\Phi(t)\Phi^{T}(t)B\\&+X^{T}P^{(\nu)}\Omega_{\tau}\Phi(t)\Phi^{T}(t)C+V_{\tau}^{T}\Phi(t)\Phi^{T}(t)C+U^{T}\Omega_{\eta}\Phi(t)\Phi^{T}(t)E\\&+V_{\eta}^{T}\Phi(t)\Phi^{T}(t)E+F^{T}\Phi(t).\end{aligned}$$

By using the product operational matrix as described in Theorem 3, it is convenient to rewrite this relation as follows:

$$\begin{aligned}X^{T}\Phi(t)=&\Big(X^{T}P^{(\nu)}\tilde{A}+G_{1}^{T}\Phi(0)A^{T}+U^{T}\tilde{B}+X^{T}P^{(\nu)}\Omega_{\tau}\tilde{C}\\&+V_{\tau}^{T}\tilde{C}+U^{T}\Omega_{\eta}\tilde{E}+V_{\eta}^{T}\tilde{E}+F^{T}\Big)\Phi(t).\end{aligned}$$

The above expression is satisfied for all t in the interval $[0,1]$; therefore

$$\begin{aligned}X^{T}=&X^{T}P^{(\nu)}\tilde{A}+G_{1}^{T}\Phi(0)A^{T}+U^{T}\tilde{B}+X^{T}P^{(\nu)}\Omega_{\tau}\tilde{C}\\&+V_{\tau}^{T}\tilde{C}+U^{T}\Omega_{\eta}\tilde{E}+V_{\eta}^{T}\tilde{E}+F^{T}.\end{aligned}$$

Now, from this relation, the unknown vector U can be derived as a function of unknown vector X, by solving the following linear system:

$$U^{T}\Lambda=Y, \tag{26}$$

where

$$\Lambda=-\left(\tilde{B}+\Omega_{\eta}\tilde{E}\right), \tag{27}$$

and

$$\begin{aligned}Y=&X^{T}P^{(\nu)}\tilde{A}+G_{1}^{T}\Phi(0)A^{T}+X^{T}P^{(\nu)}\Omega_{\tau}\tilde{C}\\&+S_{\tau}^{T}\tilde{C}+S_{\eta}^{T}\tilde{E}+F^{T}-X^{T}.\end{aligned} \tag{28}$$

By inserting the obtained vector U in (18), the unknown control function $u(t)$ can be derived as a function of the unknown vector X. Therefore, the performance index can be derived as:

$$J\left[x_{0},x_{1},\ldots,x_{N}\right]\simeq\int_{0}^{1}\Theta\left(t,X\right)dt, \tag{29}$$

where

$$\Theta(t,X)=\left[\left(X^{T}P^{(\nu)}\Phi(t)+d^{T}\Phi(t)\right)^{2}\Phi^{T}(t)R+\left(U^{T}\Phi(t)\right)^{2}\Phi^{T}(t)S\right].$$

Moreover, by applying the Gauss–Legendre quadrature formula in the interval $[0,1]$, the performance index $J[x_0,x_1,\ldots,x_N]$ can be approximated as

$$J[x_0,x_1,\ldots,x_N]\simeq\sum_{k=1}^{r}w_k\Theta(t_k,X), \tag{30}$$

where w_k and t_k are Gauss–Legendre quadrature nodes and weights, respectively. Finally, the necessary conditions for the optimality of the performance index are

$$\frac{\partial J}{\partial x_i}=0,\quad i=0,1,\ldots,N, \tag{31}$$

which constitute a system of algebraic equations for unknown vector X. By solving this system and determining the vector X, the optimal state function $x(t)$ and control function $u(t)$ can be approximated by inserting vector X in (18) and (23), respectively.

Algorithm 1 Algorithm in pseudo-code format

Inputs: N, α, β, ν, r, τ, η, the functions $r(t), s(t), f(t), g_1(t), g_2(t), a(t), c(t)$, and $b(t), e(t)$.

Step 1: Construct the Hahn polynomials vector $\Phi(t)$ and weight function $w(t)$ from (5) and (9).

Step 2: Define an unknown $(N+1)$ vector X and approximate $D^{\nu}x(t)$ in (17).

Step 3: Compute the fractional operational matrix $P^{(\nu)}$ form Theorem 2.

Step 4: Compute the coefficient vectors R, S, A, C, B, E, G_1, G_2, and F from (19)–(22).

Step 5: Compute the product operational matrices $\tilde{A}, \tilde{C}$ and $\tilde{B}, \tilde{E}$ from Theorem 3.

Step 6: Compute the matrices Ω_τ and Ω_η and vectors V_τ and V_η from Lemma 2 in Appendix A.

Step 7: Compute the matrix Λ and vector Y from (27) and (28).

Step 7: Compute the vectors U from (26).

Step 8: Insert the obtained vectors U in (18) to approximate the control function $u(t)$.

Step 9: Compute the performance index J in (29).

Step 10: Apply the Gauss–Legendre quadrature formula to approximate J in (30).

Step 11: Constitute the system of algebraic equations in (31).

Step 12: Solve the obtained system of algebraic equations in Step 11 to get the unknown vector X.

Outputs: The approximate state and control functions $x(t)\simeq X^{T}\Phi(t)$ and $u(t)\simeq U^{T}\Phi(t)$.

5.1 Main features of the proposed method

A direct numerical method based on discrete orthogonal polynomials is proposed to approximate solution of time-delay fractional optimal control problems. As these polynomials are orthogonal with respect to a discrete norm, implementation of the proposed numerical method is more efficient and less complex in comparison to similar methods that in which continuous polynomials are used. In the proposed method, the fractional derivative of an unknown state function is approximated by using the discrete polynomials. Then, the operational matrix of fractional integration together with the dynamical system is used to approximate the control function as a function of state function. Consequently, the need for the direct approximation of control function and Lagrange multiplier method are eliminated. The numerical algorithm of the proposed method in pseudo-code format is designed in the following section.

6 Illustrative examples

In this section, some illustrative examples are considered to investigate the efficiency and accuracy of the presented SHPs method. In all numerical examples, the orthogonal SHPs over the interval $[0,1]$ with $\alpha=\beta=1$ are used to approximate the solution of (1)–(4). For the problems defined in different intervals, first, the interval has been transformed into $[0,1]$. All the numerical simulation have been done by using Maple 17 with 20 digits precision.

Example 1. In the first example, we consider the following TDFOCP [20]:

$$J=\frac{1}{2}\int_0^2 \left[x^2(t)+u^2(t)\right]dt$$

subjected to the dynamical system

$$\begin{aligned} &D^{\nu}x(t)=a_1(t)x\left(t-1\right)+u\left(t\right), \quad 0\le t\le 2,\ 0<\nu\le 1,\\ &x(t)=1, \quad -1\le t\le 0, \end{aligned}$$

where $a_1(t)$ is a known continuous function. For $a_1(t)=1$, $\nu=1$, and $N=12$, Table 1 provides a comparison between the obtained optimal value J and those derived by Walsh series [33], average approximation [5], Bernstein polynomials [37], Boubaker functions [34], shifted Legendre polynomials [7], Bernoulli wavelet method [35], and linear programming [20]. From the presented results of Table 1, it is clear that the numerical results obtained by the presented SHPs method have a good agreement with the true solution given by the average approximation [5], Walsh series [33], and linear pro-

gramming [20]. Furthermore, it is easy to conclude that the reported values in [7, 34, 35, 37] are not in good agreement with the other numerical results. In order to investigate the convergence issue of the proposed algorithm, the obtained optimal values of J for different values of N and ν are presented in Table 2 and compared with the reported values in [7]. For $a_1(t) = t$, the proposed method is employed to solve these TDFOCPs for various values of ν. In Table 3, a comparison is made between the obtained optimal values of J and the derived values by linear programming method [20]. From the presented results in Table 3, we can conclude that the derived optimal values J are in a good agreement with the numerical solutions reported in [20]. Finally, the graph of approximate state function $x(t)$ and control function $u(t)$ with $N = 12$ are plotted in Figures 1 and 2, for $a_1(t) = 1$ and $a_1(t) = t$, respectively. From the obtained results, it is clear that the proposed SHPs method is efficient and accurate in solving TDFOCP and that, as the fractional order ν approaches 1, the approximate solution converges to that of integer-order problem. Moreover, the presented results in Tables 2 and 3 indicate that numerical results converge well as the number of basis function N increases.

Table 1: Comparison of the performance index values obtained by different methods for $a_1(t) = 1$ and $\nu = 1$ (Example 1).

Method	Performance index value J
Presented method	1.64731019
Linear programming [20]	1.64886527
Average approximation method [5]	1.6419
Walsh series [33]	1.6497
Legendre polynomials [7]	0.4727464
Bernstein polynomials [37]	0.6381
Boubaker functions [34]	0.00002674
Bernoulli wavelet [35]	0.3048

Table 2: The performance index J for $a_1(t) = 1$ and different values of ν (Example 1).

	$\nu = 0.7$	$\nu = 0.8$	$\nu = 0.9$	$\nu = 0.95$	$\nu = 0.99$
$N = 6$	1.5581796	1.5964603	1.6266313	1.6384862	1.6464170
$N = 8$	1.5487500	1.5907820	1.6239023	1.6368166	1.6454020
$N = 12$	1.5410926	1.5862331	1.6239023	1.6357471	1.6448406
[7]	–	0.4985242	0.5021900	–	0.4778890

Table 3: The performance index J for $a_1(t) = t$ and different values of ν (Example 1).

	$\nu = 0.7$	$\nu = 0.8$	$\nu = 0.9$	$\nu = 0.95$	$\nu = 1$
$N = 6$	0.92467418	0.97005408	1.01321336	1.03283445	1.05078685
$N = 8$	0.90796836	0.95993974	1.00820712	1.02968701	1.04905258
$N = 12$	0.90014337	0.95537457	1.00625088	1.02863607	1.04863866
[20]	–	1.08069403	1.06577389	–	1.05137240

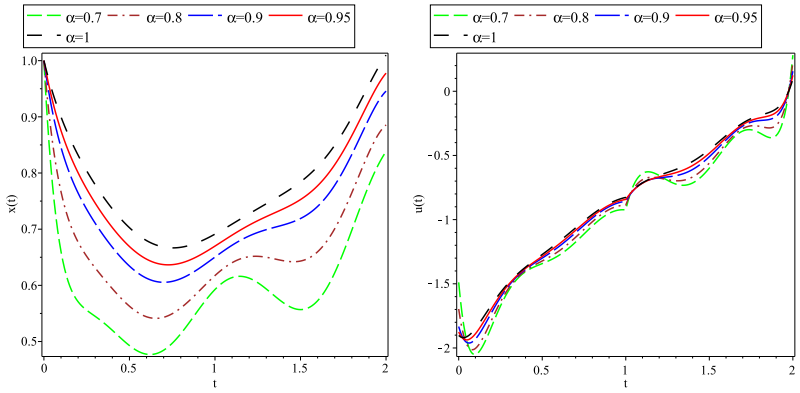


Figure 1: The approximate solutions $x(t)$ and $u(t)$ for $a_1(t) = 1$ and $N = 12$ (Example 1).

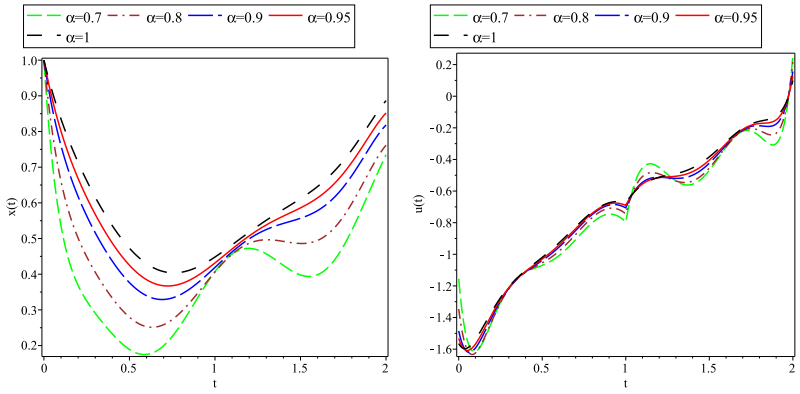


Figure 2: The approximate solutions $x(t)$ and $u(t)$ for $a_1(t) = t$ and $N = 12$ (Example 1).

Example 2. In this example, we consider the following TDFOCP [34, 35]

$$J = \frac{1}{2}\int_0^2 \left[x^2(t) + u^2(t)\right] dt$$

subjected to the dynamical system

$$\begin{aligned} &D^{\nu}x(t) = tx\,(t) + x\,(t-1) + u\,(t)\,, \quad 0 \le t \le 2,\ 0 < \nu \le 1,\\ &x(t) = 0, \quad -1 \le t \le 0. \end{aligned}$$

The SHPs and presented technique in section 5 are used to solve this problem for various values of ν. For $\nu = 1$, Table 4 provides a comparison between the results of our proposed method and those derived by FD method [22], recursive shooting method [21], line-up competition algorithm [10], Legendre multiwavelets [25], Walsh series [33] and Bernoulli wavelet [35]. From Table 4 we can see that there is good agreement between the obtained results and the those reported in Refs. [10, 21, 22]. To investigate the convergence issue of the proposed algorithm, the obtained optimal values of J for different values of N and ν are presented in Table 5. Moreover, Figure 3 shows the approximate state and control functions $x(t)$ and $u(t)$ for various values of ν and $N = 12$. Based on the obtained results, it is clear that proposed SHPs method is efficient and accurate for solving such problems and the approximate solutions converge to the exact solution as fractional order ν approaches integer order 1. Moreover, it is easy to conclude that numerical results converge well as the number of basis function N increases.

Table 4: Comparison of the performance index values obtained by different methods for $\nu = 1$ (Example 2).

Method	Performance index value J
Presented method	4.792110320
FD method [22]	4.79678
Recursive shooting method [21]	4.79682
Line-up competition algorithm [10]	4.7976
Legendre multiwavelets [25]	5.1713
Iterative dynamic programming [11]	5.0674
Walsh series [33]	6.0079
Bernoulli wavelet [35]	2.0481

Table 5: The performance index J for different values of ν (Example 2).

	$\nu = 0.7$	$\nu = 0.8$	$\nu = 0.9$	$\nu = 0.95$	$\nu = 0.99$
$N = 6$	4.04799977	4.32580075	4.58104543	4.69543943	4.77884109
$N = 8$	4.01176960	4.30195764	4.56856743	4.68746981	4.77382051
$N = 12$	3.98516242	4.28472972	4.56062884	4.68299652	4.77143293

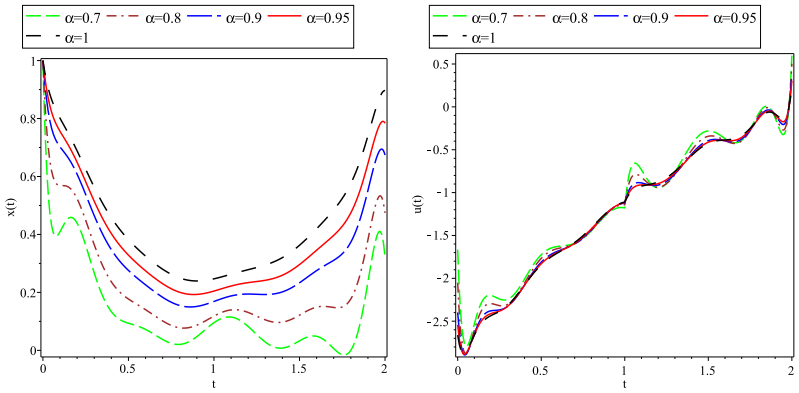


Figure 3: The approximate solutions $x(t)$ and $u(t)$ for $N = 12$ (Example 2).

Example 3. In this example, we consider the following TDFOCP with delays in state and control [7, 34, 35]

$$J = \frac{1}{2}\int_0^1 \left[x^2(t) + \frac{1}{2}u^2(t)\right] dt,$$

subjected to the dynamical system

$$\begin{aligned} &D^\nu x(t) = -x\left(t\right) + x\left(t - \tfrac{1}{3}\right) + u\left(t\right) - \tfrac{1}{2}u\left(t - \tfrac{2}{3}\right), \ 0 \le t \le 1, \quad 0 < \nu \le 1, \\ &x(t) = 1, \ -\tfrac{1}{3} \le t \le 0, \\ &u(t) = 0, \ -\tfrac{2}{3} \le t \le 0, \end{aligned}$$

The proposed SHPs method has been employed to approximate the solution of this TDFOCP for different values of ν. For $\nu = 1$, this problem has been solved by using the Hybrid functions [17, 28] and Bezier curve method [14]. Table 6 provides a comparison between the results of our proposed method and those reported in [7, 14, 17, 28, 34, 35]. From Table 6, we can see that there is a good agreement between our obtained results and the previously reported numerical results in [17, 28]. Also, this Table indicates that the reported values by using Boubaker polynomials [34], Shifted Legendre polynomials [7], Bernoulli wavelet [35], and Bezier curve method [14] are not in a good agreement with our proposed method and the valid values given in [17, 28]. To investigate the convergence issue of the proposed algorithm, the obtained optimal values of J for different values of N and ν are presented in Table 7 and compared with the presented results in [7]. Moreover, Figure 4 shows the approximate state and control functions $x(t)$ and $u(t)$ for various values of and $N = 12$. Based on the obtained results, it is clear that the proposed SHPs method is efficient and accurate for solving such problems and that the approximate solutions converge to the exact solution as fractional order

ν approaches integer order 1. Moreover, it is easy to conclude that numerical results converge well as the number of basis function N increases.

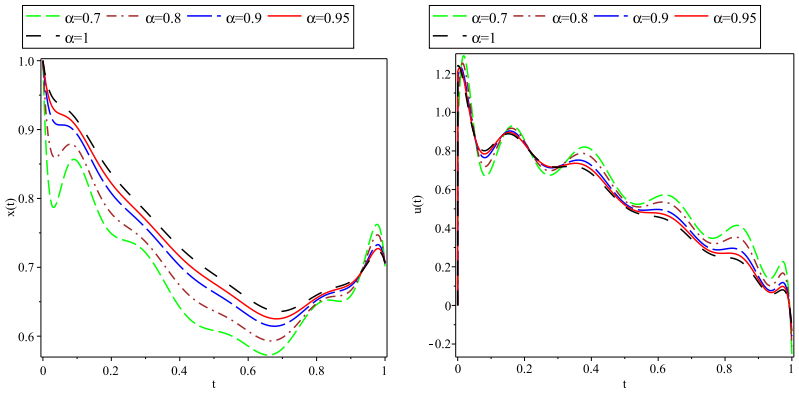


Figure 4: The approximate solutions $x(t)$ and $u(t)$ for $N = 12$ (Example 3).

Table 6: Comparison of the performance index values obtained by different methods for $\nu = 1$ (Example 3).

Method	Performance index value J
Presented method method	0.373230152
Hybrid Legendre functions [28]	0.37311241
Hybrid Bernoulli functions [17]	0.37310517
Bernoulli wavelet [35]	0.1027
Bezier curve method [14]	0.4220497643
Boubaker polynomials [34]	0.04553
Legendre polynomials [7]	0.01451

Table 7: The performance index J for different values of ν (Example 3).

	$\nu = 0.7$	$\nu = 0.8$	$\nu = 0.9$	$\nu = 0.95$	$\nu = 0.99$
$N = 8$	0.3393198	0.3496035	0.3600181	0.3652627	0.3694686
$N = 12$	0.3415913	0.3515520	0.3613160	0.3661582	0.3700268
$N = 16$	0.3549220	0.3549220	0.3637609	0.3683158	0.3720233
Ref. [7]	–	0.0126951	0.0116573	–	0.0139046

Example 4. Finally, consider the following TDFOCP [7, 14]

$$J = \frac{1}{2}\int_0^{\frac{1}{4}} \left[x^2(t) + u^2(t)\right] dt$$

subjected to the dynamical system

$$\begin{aligned} &D^{\nu}x(t) = x\left(t\right) + u\left(t - \tfrac{1}{10}\right) + u\left(t\right),\ 0 \le t \le \tfrac{1}{4},\quad 0 < \nu \le 1,\\ &u(t) = 0,\quad -\tfrac{1}{10} \le t \le 0,\\ &x(0) = 1. \end{aligned}$$

Basin and Gonzalez [6] and Ghomanjani et al. [14] solved this problem for $\nu = 1$ by using the Bezier curve and Linear-quadratic method, respectively. Here, this TDFOCP has been solved by using the proposed Hahn polynomial method for variose values of ν. In Table 8, a comparison is made between the optimal values of J obtained by the SHPs method and those derived by the Bezier curve method [14], shifted Legendre polynomials [7], and Linear-quadratic method [6]. From this Table, we can realize that the SHPs method is efficient and that the obtained results are in good agreement with the presented results in [6, 14]. However, the reported results in [7] are not in good agreement with other methods. For noninteger values of ν, the obtained optimal values of J are presented in Table 9 and compared with the value of J derived by shifted Legendre polynomials [7]. Finally, Figure 5 shows the approximate state function $x(t)$ and the control function $u(t)$ for various values of ν and $N = 12$. Based on the obtained results, it is clear that the proposed SHPs method is efficient and accurate for solving such problems and that the approximate solutions converge to the exact solution as fractional order α approaches integer order 1. Moreover, it is easy to conclude that numerical results converge well as the number of basis function N increases.

Table 8: Comparison of the performance index values obtained by different methods for $\nu = 1$ (Example 4).

Method	Performance index value J
Presented method method	0.1554640279
Bezier curve method [14]	0.1565866913
Linear-quadratic [6]	0.1563
Legendre polynomials [7]	0.0143671

Table 9: The performance index J for different values of N and ν (Example 4).

	$\nu = 0.7$	$\nu = 0.8$	$\nu = 0.9$	$\nu = 0.95$	$\nu = 0.99$
$N = 6$	0.1594198	0.1574747	0.1553910	0.1543186	0.1534526
$N = 8$	0.1607079	0.1584826	0.1561770	0.1550124	0.1540808
$N = 12$	0.1616864	0.1592375	0.1567619	0.1555286	0.1545484
Ref. [7]	–	0.0126951	0.0116573	–	0.0139046

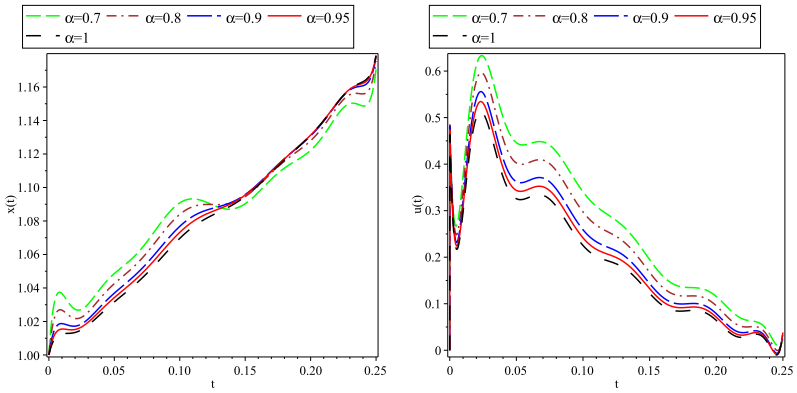


Figure 5: The approximate solutions $x(t)$ and $u(t)$ for $N = 12$ (Example 4).

7 Conclusion

A direct numerical method based on the discrete SHPs was proposed to approximate solution of TDFOCPs with quadratic performance index. First, the operational matrix of fractional integration in the Riemann–Liouville sense and product operational matrix for the shifted Hahn polynomials were derived. The fractional derivative of unknown state function was approximated by using the SHPs. Then, the operational matrix of fractional integration together with the dynamical system were used to approximate the control function as a function of state function. Finally, these approximations were put in the performance index and necessary conditions for optimality transform the under consideration TDFOCPs into an algebraic system. Numerical results confirm that the presented SHPs method is accurate and effective.

References

1. Aiello, W.G., Freedman, H.I. and Wu, J. *Analysis of a model representing stage-structured population growth with state-dependent time delay*, SIAM J. Appl. Math., 52(3), 855–869.

2. Asli, B.H.S. and Flusser, J. *New discrete orthogonal moments for signal analysis*, Signal Proc., 141 (2017), 57–73.

3. Baleanu, D., Diethelm, K., Scalas, E. and Trujillo, J.J. *Fractional calculus: Models and numerical methods*, World Scientific, 2016.

4. Baleanu, D., Maaraba, T. and Jarad, F. *Fractional variational principles with delay*, J. Phys. A 41 (31) (2008), 315403, 8 pp.

5. Banks, H.T.J. and Burns, A. *Hereditary control problems: Numerical methods based on averaging approximations*, SIAM J. Control Optimization, 16(2) (1978), 169–208.

6. Basin, M. and Rodriguez-Gonzalez, J. *Optimal control for linear systems with multiple time delays in control input*, IEEE Trans. Autom. Control., 51(1)(2006), 91–97.

7. Bhrawy, A.H., and Ezz-Eldien, S.S. *A new Legendre operational technique for delay fractional optimal control problems*, Calcolo. 53(4) (2016), 521–543.

8. Canuto, C., Hussaini, M., Quarteroni, A. and Zang, T. *Spectral methods in fluid dynamics*, Springer, 1988.

9. Cattani, C., Guariglia, E., and Wang, S. *On the Critical Strip of the Riemann zeta Fractional Derivative*, Fund. Inform. 151, (1-4) (2017), 459–472.

10. Chen, C.L., Sun, D.Y. and Chang, C.Y. *Numerical solution of time-delayed optimal control problems by iterative dynamic programming*, Optimal Control Appl. Methods 21(3) (2000), 91–105.

11. Dadebo, S. and Luus, R. *Optimal control of time-delay systems by dynamic programming*, Optimal Control Appl. Methods 13(1) (1992), 29–41.

12. Dehghan, M., Abbaszadeh, M. and Deng, W. *Fourth-order numerical method for the space-time tempered fractional diffusion-wave equation*, Appl. Math. Lett. 73 (2017), 120–127.

13. Evans, D.J. and Raslan, K.R. *The Adomian decomposition method for solving delay differential equation*, Int. J. comput. Math. 82(2005), 49–54.

14. Ghomanjani, F., Farahi, M.H. and Gachpazan, M. *Optimal control of time-varying linear delay systems based on the Bezier curves*, Comput. Appl. Math. 33(3) (2014), 687–715.

15. Goertz, R. and Öffner, P. *On Hahn polynomial expansion of a continuous function of bounded variation*, arXiv:1610.06748 (2016).

16. Goertz, R. and Öffner, P. *Spectral accuracy for the Hahn polynomials*, arXiv:1609.07291 [math.NA].

17. Haddadi, N., Ordokhani, Y. and Razzaghi, M. *Optimal control of delay systems by using a hybrid functions approximation*, J. Optim. Theory Appl. 153(2) (2012), 338–356.

18. Hahn, W. *Uber Orthogonalpolynome, die q-Differenzengleichungen genugen*, (German) Math. Nachr. 2, (1949). 4–34.

19. Hwang, C. and Chen, M.Y. *Analysis of time-delay systems using the Galerkin method*, Int. J. Control 44 (1986) 847–866.

20. Jajarmi, A. and Baleanu, D. *Suboptimal control of fractional-order dynamic systems with delay argument*, J. Vib. Control 24(12) (2018), 2430–2446.

21. Jajarmi, A. and Hajipour, M. *An efficient recursive shooting method for the optimal control of time-varying systems with state time-delay*, Appl. Math. Model., 40(4) (2016), 2756–2769.

22. Jajarmi, A. and Hajipour, M. *An efficient finite difference method for the time-delay optimal control problems with time-varying delay*, Asian J. Control, 19(2) (2017), 1–10.

23. Jamshidi, M. and Wang, C.M. *A computational algorithm for large-scale nonlinear time-delay systems*, IEEE Trans. Systems Man Cybernet, (1984) 14, 2–9.

24. Karlin, S. and McGregor, J.L. *The Hahn polynomials, formulas, and an application*, Scripta Math., 26 (1961), pp. 33–46.

25. Khellat, F. *Optimal control of linear time-delayed systems by linear Legendre multiwavelets*, J. Optim. Theory Appl., 143(1)(2009), 107–121.

26. Kuang, Y. *Delay differential equations: with applications in population dynamics*, Academic Press; 1993.

27. Malek-Zavarei, M. and Jamshidi, M. *Time-delay systems: Analysis, optimization and applications*, Elsevier Science Ltd, New York, 1987.

28. Marzban, H.R. and Razzaghi, M. *Optimal control of linear delay systems via hybrid of block-pulse and Legendre polynomials*, J. Franklin Inst. 341(3) (2004), 279–293.

29. Marzban, H.R. and Razzaghi, M. *Solution of multi-delay systems using hybrid of block-pulse functions and Taylor series*, J Sound. Vibration, 292 (2006), 954–963.

30. Mohammadi, F., Hosseini, M.M., and Mohyud-Din, S.T. *Legendre wavelet galerkin method for solving ordinary differential equations with non-analytic solution*, Internat. J. Systems Sci. 42(4) (2011), 579–585.

31. Mohammadi, F. and Mohyud-Din, S.T. *A fractional-order Legendre collocation method for solving the Bagley-Torvik equations*, Adv. Difference Equ. 2016(269) (2016), 14 pp.

32. Oldham, K.B. and Spanier, J. *The fractional calculus*, Academic Press, New York, 1974.

33. Palanisamy, K.R. and Rao, G.P. *Optimal control of linear systems with delays in state and control via Wash functions*, IEE Proc. 130(6) (1983), 300–312.

34. Rabiei, K., Ordokhani, Y. and Babolian, E. *Fractional-order Boubaker functions and their applications in solving delay fractional optimal control problems*, J. Vib. Control, 24(15) (2018), 3370–3383.

35. Rahimkhani, P., Ordokhani, Y. and Babolian, E. *An efficient approximate method for solving delay fractional optimal control problems*, Nonlinear Dynam. 86(3) (2016), 1649–1661.

36. Saeed, U. and Rehman, M.U. *Hermite wavelet method for fractional delay differential equations*. J. Differ. Equ. (2014), 1–8.

37. Safaie, E., Farahi, M.H. and Farmani-Ardehaie, M. *An approximate method for numerically solving multi-dimensional delay fractional optimal control problems by Bernstein polynomials*, Comput. Appl. Math. 34(3) (2015), 831–846.

38. Samko, S.G., Kilbas, A.A. and Marichev, O.I. *Fractional Integrals and Derivatives: Theory and Applications*, Gordon and Breach, Langhorne, 1993.

39. Sedaghat, S., Ordokhani, Y. and Dehghan, M. *Numerical solution of the delay differential equations of pantograph type via Chebyshev polynomials*, Commun. Nonl. Sci. Numer. Simul. 17 (2012), 4815–4830

40. Srivastava, H.M., Kumar, D. and Singh, J. *An efficient analytical technique for fractional model of vibration equation*, Appl. Math. Mod. 45 (2017), 192–204.

41. Wang, X.T. *Numerical solutions of optimal control for time delay systems by hybrid of block-pulse functions and Legendre polynomials*, Appl. Math. Comput. 184(2)(2007), 849–856.

42. Wilson, M.W. *On the Hahn Polynomials*, SIAM J. Math. Anal., 1(1) (1970), 131–139.

43. Xiu, D. and Karniadakis, G.E. *The Wiener-Askey polynomial chaos for stochastic differential equations*, SIAM J Sci. comput. 24(2) (2002), 619–644.

44. Yang, Y. and Huang, Y. *Spectral-collocation methods for fractional pantograph delay-integrodifferential equations*, Adv. Math. Phys. (2013), 1–14.

45. Yu, Z.H. *Variational iteration method for solving the multi-pantograph delay equation*, Phys. Lett. A 372 (2008), 6475–6479.

Appendix A

Lemma 2. *Consider the function*

$$x(t-\tau)=\begin{cases} G_1^T\Phi(t-\tau),\ 0\le t\le\tau,\\ \\ X^TP^{(\nu)}\Phi(t-\tau)+d^T\Phi(t-\tau),\ \tau\le t\le 1,\end{cases}$$

it can be expanded into SHPs as follows:

$$x(t-\tau)=X^TP^{(\nu)}\Omega_\tau\Phi(t)+V_\tau\Phi(t), \tag{32}$$

in which Ω_τ and S_τ are, respectively, $(N+1)\times(N+1)$ and $(N+1)\times 1$ matrices defined as follows:

Proof. By expanding the function $x(t-\tau)$ expanded by the SHPs, we get

$$x(t-\tau)=\sum_{i=0}^{N}z_iH_i(t)=Z^T\Phi(t),$$

where ith element of the coefficient vector Z can be derived as

$$z_i=\frac{\langle x(t-\tau),H_i(t)\rangle_{\bar w}}{\langle H_i(t),H_i(t)\rangle_{\bar w}}=\frac{1}{\pi(i)}\sum_{k=0}^{N}x\left(\frac{k}{N}-\tau\right)H_i\left(\frac{k}{N}\right)\bar w\left(\frac{k}{N}\right),\quad i=0,1,\dots,N.$$

Let l be the greatest integer such that $l\le\tau N$. Then

$$z_i=\frac{G_1^T}{\pi(i)}\sum_{k=0}^{l}\Phi\left(\frac{k}{N}-\tau\right)H_i\left(\frac{k}{N}\right)\bar w\left(\frac{k}{N}\right)+\frac{X^TP^{(\nu)}}{\pi(i)}\sum_{k=l+1}^{N}\Phi\left(\frac{k}{N}-\tau\right)H_i\left(\frac{k}{N}\right)\bar w\left(\frac{k}{N}\right)$$

$$+\frac{d^T}{\pi(i)}\sum_{k=l+1}^{N}\Phi\left(\frac{k}{N}-\tau\right)H_i\left(\frac{k}{N}\right)\bar w\left(\frac{k}{N}\right)$$

Now, let us define

$$\chi_i = \frac{1}{\pi(i)} \begin{bmatrix} \sum_{k=0}^{l} H_0\left(\frac{k}{N}-\tau\right) H_i\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \\ \sum_{k=0}^{l} H_1\left(\frac{k}{N}-\tau\right) H_i\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \\ \vdots \\ \sum_{k=0}^{l} H_N\left(\frac{k}{N}-\tau\right) H_i\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \end{bmatrix},$$

and

$$\Pi_i = \frac{1}{\pi(i)} \begin{bmatrix} \sum_{k=l+1}^{N} H_0\left(\frac{k}{N}-\tau\right) H_i\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \\ \sum_{k=l+1}^{N} H_1\left(\frac{k}{N}-\tau\right) H_i\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \\ \vdots \\ \sum_{k=l+1}^{N} H_N\left(\frac{k}{N}-\tau\right) H_i\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \end{bmatrix},$$

then, the element z_i can be obtained as

$$z_i = G_1^T \chi_i + d^T \Pi_i + X^T P^{(\nu)} \Pi_i.$$

Consequently, we get

$$x(t-\tau) = X^T P^{(\nu)} \Omega_\tau \Phi(t) + V_\tau^T \Phi(t),$$

where V_τ is an $(N+1)\times 1$ vector as

$$V_\tau^T = \left[G_1^T\chi_0 + d^T Y_0, G_1^T\chi_1 + d^T Y_1, \ldots, G_1^T\chi_N + d^T Y_N\right]$$

and Ω_τ is an $(N+1)\times(N+1)$ matrix which its ith column is Π_i, that is,

$$\Omega_\tau = \frac{1}{\pi(i)} \begin{bmatrix} \sum_{k=l+1}^{N} H_0\left(\frac{k}{N}-\tau\right) H_0\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) & \ldots & \sum_{k=l+1}^{N} H_0\left(\frac{k}{N}-\tau\right) H_N\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \\ \sum_{k=l+1}^{N} H_1\left(\frac{k}{N}-\tau\right) H_0\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) & \ldots & \sum_{k=l+1}^{N} H_1\left(\frac{k}{N}-\tau\right) H_N\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \\ \vdots & \ddots & \\ \sum_{k=l+1}^{N} H_N\left(\frac{k}{N}-\tau\right) H_0\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) & \ldots & \sum_{k=l+1}^{N} H_N\left(\frac{k}{N}-\tau\right) H_N\left(\frac{k}{N}\right) \bar{w}\left(\frac{k}{N}\right) \end{bmatrix}.$$

Remark 2. In the same way, it can be proved that the

$$u\left(t-\eta\right) = U^T \Omega_\eta \Phi\left(t\right) + V_\eta^T \Phi\left(t\right),$$

where Ω_η and V_η are $(N+1)\times(N+1)$ and $(N+1)\times 1$ matrices, respectively. □

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 77–102
DOI:10.22067/ijnao.v9i2.72109

Research Article

Chebyshev pseudo-spectral method for optimal control problem of Burgers' equation

F. Mohammadizadeh*, H. A. Tehrani and M. H. Noori Skandari

Abstract

In this study, an indirect method is proposed based on the Chebyshev pseudo-spectral method for solving optimal control problems governed by Burgers' equation. Pseudo-spectral methods are one of the most accurate methods for solving nonlinear continuous-time problems, specially optimal control problems. By using optimality conditions, the original optimal control problem is first reduced to a system of partial differential equations with boundary conditions. Control and state functions are then approximated by interpolating polynomials. The convergence is analyzed, and some numerical examples are solved to show the efficiency and capability of the method.

AMS(2010): 49J20; 35S05; 93C20.

Keywords: Burgers' equation; Optimal control; Chebyshev-Gauss-Lobatto nodes.

1 Introduction

Time-dependent partial differential equations (PDEs) play an important role in various fields of application, such as fluid dynamics [4,28], electro magnetics [17,24], or heat transfers [10]. Often, these differential equations consist of

*Corresponding author
Received 17 April 2018; revised 10 February 2019; accepted 1 March 2019
F. Mohammadizadeh
Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran.
e-mail: f.mohamadi093@gmail.com

H. A. Tehrani
Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran.
e-mail: hahsani@shahroodut.ac.ir

M. H. Noori Skandari
Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran.
e-mail: math.noori@yahoo.com

nonlinear terms, which are challenging to solve numerically. Furthermore, in engineering applications, we often interest in an optimal solution of the considered PDE with a certain objective function. This leads to the mathematical field of PDE constrained optimization where a cost function is minimized and the PDE is considered as a constraint.

Optimal control of viscous Burgers' equation is one of the most important PDEs constraint optimization, which is taken into consideration and several papers have recently been presented in its numerical solution. Kobayashi [12] has concerned with adaptive stabilization and regulator design for a viscous Burgers' equation by nonlinear boundary control. Yilmaz and Karasozen [29] by using the high level modeling and simulation package COMSOL Multiphysics solved the optimal control of unsteady Burgers' equation without constraints and with control constraints. Also, they applied an all-at-once method for the optimal control of the unsteady Burgers' equation [30]. Moreover, in other work, they [11] transformed the optimality system for boundary controlled unsteady Burgers' equation after linearization into a biharmonic equation in the space-time domain. Smaoui et al. [23] have dealt with the sliding mode control (SMC) of the forced generalized Burgers' equation via the Karhunen–Loeve (K-L) Galerkin method. Hashemi and Werner [8] presented a finite difference scheme for the one-dimensional viscous Burgers' equation, which boundary conditions are taken as control inputs. Zeng and Zhang [31] designed a new preconditioning technique along with MINRES (minimal residual) in nonstandard inner product for the linear system of equations arising for optimal control of the unsteady Burgers' equation. Kucuk and Sadek [13] analyzed the dynamics of the forced Burgers' equation subject to Dirichlet boundary conditions by using the boundary control with the objective of minimizing the distance between the final state function and target profile along with the energy of the control. Also, they applied a robust technique for solving optimal control of coupled Burgers' equation; see [21]. Noack and Walther [16] used adjoint techniques for efficient evaluations of the gradient of the objective in gradient based on optimization algorithms. Marburger and Pinnau [15] showed the convergence of method to solve optimal control problems, where the constraints have been discretized by a particle method. Allahverdi et al. [2] discussed the efficiency of various numerical methods for the inverse design of the Burgers' equation. An augmented Lagrangian-SQP technique depending upon second-order sufficient optimality condition is analyzed in [27]. In [5], a comparison of three different numerical methods for optimal control of Burgers' equation is carried out. In [25], a Lagrangian-Newton-SQP method is presented for the solution of optimal control of Burgers' equation, where the control is restricted by pointwise lower and upper bounds. Proper orthogonal decomposition method is utilized in [14] to solve optimal control problems of the Burgers' equation. Furthermore, some controllability results for viscous Burgers' equation with distributed controls are presented in [6].

In some papers, at first the problem is discretized on one of the variables and then the Runge–Kutta method is applied to discrete the optimality conditions. To achieve good results, we have to take the number of points of discretization big enough that it can be too time consuming and has computational complexity.

The pseudo-spectral (PS) methods are one of the most accurate methods to solve continuous-time problems including ordinary differential equations (ODEs) and PDEs. Using the direct Chebyshev PS(CPS) method for optimal control problems governed by PDEs does not usually give an explicit solution. Hence, we use an indirect CPS method for solving necessary optimal conditions in an optimal control problem of Burgers' equation and we obtain an approximate optimal solution. It is the first time that the CPS method is applied by discretization of variables synchronic. By numerical examples, it can be seen that the CPS method is more effective than other methods and we can achieve the better results for the solution of optimal control problem of Burgers' equation. In fact, the error of CPS method is less than that of other methods.

The paper is organized as follows: In Section 2, we introduce the optimal control problem of Burgers' equation. In Section 3, we give the optimality conditions for optimal control problem of Burgers' equation. In Section 4, we utilize the CPS method to discretize optimality conditions. In Section 5, the convergence of the method is analyzed. Finally, the numerical examples and conclusion are given in Sections 6 and 7, respectively.

2 Optimal control problem of Burgers' equation

The distributed optimal control problem for the Burgers' equation can be stated as follows:

$$\text{Minimize} \quad J[y,u] = \frac{1}{2}\int_0^T \int_a^b (y(t,x) - z(t,x))^2 dxdt + \frac{\alpha}{2}\int_0^T \int_a^b u^2(t,x) dxdt \tag{1}$$

subject to

$$y_t(t,x) + y(t,x)y_x(t,x) - \nu y_{xx}(t,x) = \Phi(u), \ (t,x) \in Q = [0,T] \times [a,b], \tag{2}$$

$$y(t,a) = y(t,b) = 0, \quad t \in \Sigma = [0,T], \tag{3}$$

$$y(0,x) = y_0(x), \quad x \in \Omega = [a,b], \tag{4}$$

where $y(.,.)$ is the state variable, $u(.,.)$ is the control variable, $\alpha > 0$ is the regularization parameter, $\nu > 0$ denotes the viscosity parameter, and Φ is a given function. An usual selection of function Φ is as follows:

$$\Phi(u) = \begin{cases} u & u \ in \ \overline{\Omega}, \\ 0 & u \ in \ \Omega - \overline{\Omega}, \end{cases}$$

where $\overline{\Omega}$ is the set of active controls; see [5, 25, 27].

At first, optimality conditions for problem (1)–(4) are given, then we indirectly develop the CPS method to achieve an approximate optimal solution.

3 Optimality conditions

Here, we summarize the process of achieving the optimality conditions for the optimal control problem (1)–(4).
Let $L^1_{loc}(\Omega)$ be the set of all functions that are Lebesgue integrable on every compact subset of Ω and let $W^{k,2}(\Omega)$ be the linear space of all functions $w \in L^2(\Omega)$ having weak derivatives $D^\alpha w$ in $L^2(\Omega)$ for all multi-indices α of length $\mid \alpha \mid \leq k$. Assume that $C_0^\infty(\Omega)$ is the space of differentiable functions of every arbitrary order on Ω. The closure of $C_0^\infty(\Omega)$ in $W^{k,p}(\Omega)$ is denoted by $W_0^{k,p}(\Omega)$. Moreover, we define $H_0^k(\Omega) = W_0^{k,2}(\Omega)$.

Definition 1. Let $y \in L^1_{loc}(\Omega)$ and let some multi-index α be given. If a function $\omega \in L^1_{loc}(\Omega)$ satisfies

$$\int_\Omega y(x) D^\alpha v(x) dx = (-1)^{|\alpha|} \int_\Omega \omega(x) v(x) dx, \qquad \text{for all } v \in C_0^\infty(\Omega),$$

then ω is called the weak derivative of y (associated with α).

Let $H = L^2(\Omega)$ and $V = H_0^1(\Omega)$ be Hilbert spaces. We make use of the following Hilbert space

$$W(0,T) = \{\phi \in L^2([0,T]; V) : \phi_t \in L^2([0,T]; V^\star)\},$$

where $V^\star$ denotes the dual space of V. The inner product in the Hilbert space V is given with the natural inner product in H as

$$(\phi, \psi)_V = (\phi', \psi')_H, \quad for \ \phi, \psi \in V,$$

where ϕ' and ψ' are the derivative of ϕ and ψ, respectively.

Definition 2. Every function $y \in W(0,T)$ that satisfies

$$\begin{cases} < y_t(t), \phi >_{V^\star, V} + \nu(y_t(t), \phi)_V + (y(t) y_x(t), \phi)_H = ((f + \Phi(u))(t), \phi)_H \\ \phi \in V, \ t \in [0,T], \ (y(0), \chi)_H = (y_0, \chi), \quad \chi \in H, \end{cases}$$

where

$$< y_t(t), \phi >_{V^\star, V} = \int_\Omega \eta.\phi dx + \int_\Omega \eta'.\phi' dx, \quad \eta \in V,$$

is called a weak solution for Burgers' equation (2) and conditions (3) and (4).

In order to achieve the optimality conditions, the operator $e : X \longrightarrow Y$ is introduced by

$$e(y,u) = (e_1(y,u), e_2(y,u)) = (y_t - \nu y_{xx} + y y_x - f - \Phi(u), y(0) - y_0),$$

where $y(0) = y(0,x)$, $X = W(V) \times L^2(\bar{\Omega})$, and $Y = L^2(V) \times H$ is identified with $Y^\star = L^2(V^\star) \times H$ the dual of Y. We use $L^2(V)$ for $L^2(\Sigma; V)$. Now, the optimal control problem (1)–(4) can be transformed as the following minimization problem with equality constraints

$$\begin{aligned} &\text{minimize} \quad J(y,u) \\ &\textit{subject to} \quad e(y,u) = 0. \end{aligned}$$

Theorem 1. *Let $(y^\star, u^\star)$ be an optimal solution of (1)–(4). Then there exist Lagrange multipliers $p^\star : [0,T] \times \Omega \longrightarrow \Re$ and $\lambda^\star$ satisfying the first-order necessary optimality conditions*

$$L'(y^\star, u^\star, p^\star, \lambda^\star) = 0, \ e(y^\star, u^\star) = 0$$

with the Lagrangian

$$L(y,u,p,\lambda) = J(y,u) - (e_1(y,u), p)_{L^2(V^\star), L^2(V)} - (e_2(y,u), \lambda)_H.$$

Hence, first-order optimality conditions lead to the following optimality system (see [9, 26, 27])

$$\begin{cases} y_t - \nu y_{xx} + y y_x = \Phi(u), & (t,x) \in Q, \\ p_t + \nu p_{xx} + y p_x = y_d - y, & (t,x) \in Q, \\ y(t,a) = y(t,b) = 0, & t \in \Sigma, \\ y(0,x) = y_0, & x \in \Omega, \\ p(t,a) = p(t,b) = 0, & t \in \Sigma, \\ p(T,x) = 0, & x \in \Omega, \\ \alpha u + p = 0, & (t,x) \in Q. \end{cases} \tag{5}$$

From the last equation of system (5), we have

$$u = -\frac{1}{\alpha} p. \tag{6}$$

By using (6), we can express (5) as follows:

$$\begin{cases} y_t - \nu y_{xx} + y y_x = \Phi(-\frac{1}{\alpha}p), & (t,x) \in Q, \\ p_t + \nu p_{xx} + y p_x = y_d - y, & (t,x) \in Q, \\ y(t,a) = y(t,b) = 0, & t \in \Sigma, \\ p(t,a) = p(t,b) = 0, & t \in \Sigma, \\ y(0,x) = y_0(x), & x \in \Omega, \\ p(T,x) = 0, & x \in \Omega. \end{cases} \tag{7}$$

Now, we utilize the CPS method to solve optimality conditions (7) and obtain an approximate optimal solution for the optimal control problem (1)–(4).

4 CPS method to approximate the optimal solution

Chebyshev polynomials are in center of the CPS method, and hence we first introduce these polynomials and then demonstrate the CPS method to solve optimality conditions (7).

Chebyshev polynomials [1,18] are orthogonal polynomials, which play an important role in the theory of approximation. These polynomials are defined on $[-1,1]$ as follows:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x, \quad \ldots. \tag{8}$$

These polynomials can be generated from the following recurrence relationship:

$$T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x), \quad j \geq 1, \ x \in [-1,1]. \tag{9}$$

It is possible to give an explicit expression of Chebyshev polynomials as

$$T_j(x) = \cos(j \cos^{-1} x). \tag{10}$$

It is now easy to verify that $T_N(.)$ has N zeros within the interval $(-1,1)$ as $x_k = \cos\frac{(2k-1)\pi}{2N}$, $k = 1,2,\ldots,N$. However, in this paper, we use the CGL points on $[-1,1]$ which are as follows:

$$x_j = -\cos\frac{\pi j}{N}, \quad 0 \leq j \leq N, \tag{11}$$

where they are the roots of $(1-x^2)T'_N(x)$. For interpolating in the CPS method, the following Lagrange polynomials are utilized:

$$L_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{N} \frac{x - x_j}{x_k - x_j} = \frac{2}{N\mu_k} \sum_{j=0}^{N} \frac{1}{\mu_j} T_j(x_k) T_j(x), \quad k = 0,1,\ldots,N, \ x \in [-1,1] \ ,$$

where

$$\mu_j = \begin{cases} 2, & j = 0, N, \\ 1, & 1 \le j \le N-1, \end{cases}$$

and we have

$$L_k(t_j) = \delta_{kj} = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases} \tag{12}$$

To use the CPS method, the variables of system (7) are transformed to interval $[-1, 1]$ by the following linear transformations:

$$\begin{cases} t = \dfrac{T}{2}\bar{t} + \dfrac{T}{2}, & t \in [0, T], \quad \bar{t} \in [-1, 1], \\ x = \dfrac{b-a}{2}\bar{x} + \dfrac{b+a}{2}, & x \in [a, b], \quad \bar{x} \in [-1, 1]. \end{cases} \tag{13}$$

By (13), the optimality conditions (7) can be written as follows:

$$\begin{cases} Y_{\bar{t}} = \psi_1\left(Y(\bar{t},\bar{x}), P(\bar{t},\bar{x}), Y_{\bar{x}}(\bar{t},\bar{x}), Y_{\bar{x}\bar{x}}(\bar{t},\bar{x})\right), & (\bar{t},\bar{x}) \in [-1,1]\times[-1,1], \\ P_{\bar{t}} = \psi_2\left(Y(\bar{t},\bar{x}), P(\bar{t},\bar{x}), P_{\bar{x}}(\bar{t},\bar{x}), P_{\bar{x}\bar{x}}(\bar{t},\bar{x})\right), & (\bar{t},\bar{x}) \in [-1,1]\times[-1,1], \\ Y(\bar{t},-1) = Y(\bar{t},1) = 0, & \bar{t} \in [-1,1], \\ P(\bar{t},-1) = P(\bar{t},1) = 0, & \bar{t} \in [-1,1], \\ Y(-1,\bar{x}) = Y_0(\bar{x}), & \bar{x} \in [-1,1], \\ P(1,\bar{x}) = 0, & \bar{t} \in [-1,1], \end{cases} \tag{14}$$

where

$$Y(\bar{t},\bar{x}) = y(\frac{T}{2}\bar{t} + \frac{T}{2}, \frac{b-a}{2}\bar{x} + \frac{b+a}{2}),$$

$$P(\bar{t},\bar{x}) = p(\frac{T}{2}\bar{t} + \frac{T}{2}, \frac{b-a}{2}\bar{x} + \frac{b+a}{2}),$$

$$Y_d(\bar{t},\bar{x}) = y_d(\frac{T}{2}\bar{t} + \frac{T}{2}, \frac{b-a}{2}\bar{x} + \frac{b+a}{2}),$$

$$Y_0(\bar{x}) = y_0(\frac{b-a}{2}\bar{x} + \frac{b+a}{2}),$$

$$\psi_1\left(Y, P, Y_{\bar{x}}, Y_{\bar{x}\bar{x}}\right) = \frac{T}{2}\left((\frac{2}{b-a})^2\nu Y_{\bar{x}\bar{x}} - \frac{2}{b-a}YY_{\bar{x}} - \Phi(-\frac{1}{\alpha}P)\right),$$

$$\psi_2\left(Y, P, P_{\bar{x}}, P_{\bar{x}\bar{x}}\right) = \frac{T}{2}\left(-(\frac{2}{b-a})^2\nu P_{\bar{x}\bar{x}} - \frac{2}{b-a}YP_{\bar{x}} + Y_d - Y\right).$$

We assume that ψ_1 and ψ_2 have bounded and continuous derivatives with respect to their arguments. Hence, there exist constants M_1 and M_2 such

that

$$|\psi_1\left(Y,P,Y_{\bar{x}},Y_{\bar{x}\bar{x}}\right)-\psi_1\left(\tilde{Y},\tilde{P},\tilde{Y}_{\bar{x}},\tilde{Y}_{\bar{x}\bar{x}}\right)|\leq M_1\left(|Y-\tilde{Y}|+|P-\tilde{P}|\right),$$
$$|\psi_2\left(Y,P,P_{\bar{x}},P_{\bar{x}\bar{x}}\right)-\psi_2\left(\tilde{Y},\tilde{P},\tilde{P}_{\bar{x}},\tilde{P}_{\bar{x}\bar{x}}\right)|\leq M_2\left(|Y-\tilde{Y}|+|P-\tilde{P}|\right). \tag{15}$$

Now, to approximate the optimal solution, we utilize the following polynomials interpolating:

$$\begin{cases} Y^N(\bar{t},\bar{x})=\sum_{i=0}^{N}\sum_{j=0}^{N}\bar{a}_{ij}^N L_i(\bar{t})L_j(\bar{x}),\\ P^N(\bar{t},\bar{x})=\sum_{i=0}^{N}\sum_{j=0}^{N}\bar{b}_{ij}^N L_i(\bar{t})L_j(\bar{x}),\end{cases} \tag{16}$$

where $\bar{a}_{ij}^N$ and $\bar{b}_{ij}^N$, for $i,j=0,1,\ldots,N$, are the unknown coefficients. By (12), we have

$$\begin{cases} Y^N(\bar{t}_i,\bar{x}_j)=\bar{a}_{ij}^N,\\ P^N(\bar{t}_i,\bar{x}_j)=\bar{b}_{ij}^N,\end{cases} \tag{17}$$

where

$$\bar{t}_i=-\cos(\frac{\pi i}{N}),\quad \bar{x}_j=-\cos(\frac{\pi j}{N}).$$

To express the derivatives $Y_{\bar{t}}^N(.,.)$, $Y_{\bar{x}}^N(.,.)$, $Y_{\bar{x}\bar{x}}^N(.,.)$, $P_{\bar{t}}^N(.,.)\,P_{\bar{x}}^N(.,.)$, and $P_{\bar{x}\bar{x}}^N(.,.)$, we can use the matrix multiplication $D=(D_{kj})$ and get

$$\begin{cases} Y_{\bar{t}}^N(\bar{t}_p,\bar{x}_k)=\sum_{i=0}^{N}\bar{a}_{ik}^N D_{pi},\\ Y_{\bar{x}}^N(\bar{t}_p,\bar{x}_k)=\sum_{j=0}^{N}\bar{a}_{pj}^N D_{kj},\\ Y_{\bar{x}\bar{x}}^N(\bar{t}_p,\bar{x}_k)=\sum_{j=0}^{N}\bar{a}_{pj}^N \hat{D}_{kj},\\ P_{\bar{t}}^N(\bar{t}_p,\bar{x}_k)=\sum_{i=0}^{N}\bar{b}_{ik}^N D_{pi},\\ P_{\bar{x}}^N(\bar{t}_p,\bar{x}_k)=\sum_{j=0}^{N}\bar{b}_{pj}^N D_{kj},\\ P_{\bar{x}\bar{x}}^N(\bar{t}_p,\bar{x}_k)=\sum_{j=0}^{N}\bar{b}_{pj}^N \hat{D}_{kj},\end{cases} \tag{18}$$

where

$$D_{kj}=L_j'(\bar{t}_k)=\begin{cases}\dfrac{\mu_k}{\mu_j}(-1)^{k+j}\dfrac{1}{\bar{t}_k-\bar{t}_j}, & j\neq k,\\ -\dfrac{\bar{t}_k}{2-2\bar{t}_k^2}, & 0\leq j=k\leq N-1,\\ -\dfrac{2N^2+1}{6}, & j=k=0,\\ \dfrac{2N^2+1}{6}, & j=k=N,\end{cases} \tag{19}$$

and $\hat{D}=D\cdot D=(\hat{D}_{kj})$ where $\hat{D}_{kj}=\sum_{l=0}^{N}D_{kl}D_{lj},\ k,j=0,1,\ldots,N$. In fact, multiplying by the matrix D transforms a vector of the state variables at the CGL points to the vector of approximate derivatives at these points.

Now, by relations (16), (17), and (18), conditions (7) can be written as the following discrete form:

$$\begin{cases} \sum_{i=0}^{N} \bar{a}_{ik}^{N} D_{pi} - \psi_1 \left(\bar{a}_{pk}^{N}, \bar{b}_{pk}^{N}, \sum_{j=0}^{N} \bar{a}_{pj}^{N} D_{kj}, \sum_{j=0}^{N} \bar{a}_{pj}^{N} \hat{D}_{kj} \right) = 0, \\ \sum_{i=0}^{N} \bar{b}_{ik}^{N} D_{pi} - \psi_2 \left(\bar{a}_{pk}^{N}, \bar{b}_{pk}^{N}, \sum_{j=0}^{N} \bar{b}_{pj}^{N} D_{kj}, \sum_{j=0}^{N} \bar{b}_{pj}^{N} \hat{D}_{kj} \right) = 0, \\ \bar{a}_{p0}^{N} = \bar{a}_{pN}^{N} = 0, \ \bar{b}_{p0}^{N} = \bar{b}_{pN}^{N} = 0, \ \bar{a}_{0k}^{N} = Y_0(\bar{x}_k), \ \bar{b}_{Nk}^{N} = 0, k, p = 0, 1, \ldots, N. \end{cases} \tag{20}$$

By solving the system of algebraic equations (20), we can obtain pointwise and continuous approximate optimal solutions as (16) and (17), respectively. Also, by (6), the approximate optimal control can be given as

$$U^N(\bar{t}, \bar{x}) = \frac{-1}{\alpha} \sum_{i=0}^{N} \sum_{j=0}^{N} \bar{b}_{ij}^{N} L_i(\bar{t}) L_j(\bar{x}), \quad (\bar{t}, \bar{x}) \in [-1, 1] \times [-1, 1].$$

Moreover, the optimal value of functional (1) can be approximated as follows:

$$J_N = \frac{T}{8} \sum_{k=0}^{N} \sum_{p=0}^{N} w_k w_p [\bar{a}_{pk}^{N} - z \left(\frac{T}{2} (\bar{t}_p + 1), \frac{b-a}{2} \bar{x}_k + \frac{b+a}{2} \right) + \alpha \bar{c}_{pk}^{N2}] \tag{21}$$

where $\bar{c}_{pk}^{N} = \frac{1}{\alpha} \bar{b}_{pk}^{N}$, and $w_s, s = 0, 1, \ldots, N$, are the quadrature weights of the numerical approximation (21). For even N, the weights are

$$\begin{cases} w_0 = w_N = \frac{1}{N^2 - 1} \\ w_s = w_{N-s} = \frac{4}{N} \sum_{j=0}^{\frac{N}{2}''} \frac{1}{1 - 4j^2} \cos(\frac{2\pi j s}{N}), \quad s = 1, 2, \ldots, \frac{N}{2}, \end{cases} \tag{22}$$

and for odd N,

$$\begin{cases} w_0 = w_N = \frac{1}{N^2} \\ w_s = w_{N-s} = \frac{4}{N} \sum_{j=0}^{\frac{N-1}{2}''} \frac{1}{1 - 4j^2} \cos(\frac{2\pi j s}{N}), \quad s = 1, 2, \ldots, \frac{N-1}{2}. \end{cases} \tag{23}$$

The double prime in the weights formula denotes the first and the last elements have to be halved.

Remark 1. Notice that the approximation solutions (16) can be written as the following Kronecker (tensor) product form:

$$Y(\bar{t}, \bar{x}) \approx Y^N(\bar{t}, \bar{x}) = \sum_{i=0}^{N} \sum_{j=0}^{N} \bar{a}_{ij}^{N} L_i(\bar{t}) L_j(\bar{x}) = L(\bar{t}, \bar{x}) T_Y$$

$$P(\bar{t}, \bar{x}) \approx P^N(\bar{t}, \bar{x}) = \sum_{i=0}^{N} \sum_{j=0}^{N} \bar{b}_{ij}^{N} L_i(\bar{t}) L_j(\bar{x}) = L(\bar{t}, \bar{x}) T_P,$$

where

$$\begin{aligned} L(\bar{t},\bar{x}) &= [L_0(\bar{t})L_0(\bar{x})\cdots L_0(\bar{t})L_N(\bar{x})L_1(\bar{t})L_0(\bar{x})\cdots L_1(\bar{t})L_N(\bar{x})\cdots \\ &\qquad L_N(\bar{t})L_0(\bar{x})\cdots L_N(\bar{t})L_N(\bar{x})] \\ &= [L_0(\bar{t})\ L_1(\bar{t})\cdots L_N(\bar{t})]\otimes[L_0(\bar{x})\ L_1(\bar{x})\cdots L_N(\bar{x})] \\ &= L(\bar{t})\otimes L(\bar{x}), \end{aligned}$$

and the symbol " $\otimes$ " denotes the Kronecker product. Also

$$\begin{aligned} T_Y &= [\bar{a}_{00}^N\cdots\bar{a}_{0N}^N\bar{a}_{10}^N\cdots\bar{a}_{1N}^N\cdots\bar{a}_{N0}^N\cdots\bar{a}_{NN}^N]', \\ T_P &= [\bar{b}_{00}^N\cdots\bar{b}_{0N}^N\bar{b}_{10}^N\cdots\bar{b}_{1N}^N\cdots\bar{b}_{N0}^N\cdots\bar{b}_{NN}^N]', \end{aligned}$$

where the symbol " $'$ " denotes the transpose. All the components of T_Y and T_P are unknown and our aim is to obtain them (A similar representation can be seen in [22, 32, 33]).

5 The convergence and error analysis

In this section, first we give the following definition and then analyze the convergence of the presented method.

Assume that $\bar{\Omega} = [-1,1]\times[-1,1]$. We apply $C^r(\bar{\Omega})$ for the space of functions with continuous derivatives of rth order.

Definition 3. [19] The function $W : \mathbb{R}^+ \to \mathbb{R}^+$ with the following properties is called a modulus of continuity if

i) W is increasing,

ii) $\lim_{z\to 0} W(z) = 0$,

iii) $W(z_1+z_2) \le W(z_1)+W(z_2)$, for any z_1 and $z_2 \in \mathbb{R}^+$,

iv) there exists a constant c such that $cW(z) \ge z$ for all $0 < z \le 2$.

Some important modulus of continuity can be defined as

$$W(z) = z^{\alpha}, \quad 0 < \alpha \le 1. \tag{24}$$

Now, assume that B^2 is the unit circle in $\mathbb{R}^2$. We say that a continuous function $f(\cdot,\cdot)$ on $\bar{\Omega}$ admits $W(.)$ as a modulus of continuity, if the following value is finite

$$|f(\cdot,\cdot)|_W = \sup\{\frac{|f(\bar{t},\bar{x})-f(\tilde{t},\tilde{x})|}{W\left(\|(\bar{t},\bar{x})-(\tilde{t},\tilde{x})\|_\infty\right)} : (\bar{t},\bar{x}),\ (\tilde{t},\tilde{x}) \in \bar{\Omega}, (\bar{t},\bar{x}) \ne (\tilde{t},\tilde{x})\}. \tag{25}$$

Suppose that space $C^1_W(B^2)$ includes all functions $f(\cdot,\cdot)$ on B^2 with continuous first order partial derivatives, and it equips with the following norm:

$$\|f(\cdot,\cdot)\|_{1,W} = \|f(\cdot,\cdot)\|_\infty + \|f_{\bar{t}}(\cdot,\cdot)\|_\infty + \|f_{\bar{x}}(\cdot,\cdot)\|_\infty + |f_{\bar{t}}(\cdot,\cdot)|_W + |f_{\bar{x}}(\cdot,\cdot)|_W. \tag{26}$$

Also, we define the space $C^1_W(\bar{\Omega})$ as follows:

$$\begin{aligned} C^1_W(\bar{\Omega}) = \ & \{f(\cdot,\cdot) \in C^1(\bar{\Omega}) : \text{ for all } (\tilde{t},\tilde{x}) \in \bar{\Omega}, \text{ there exists a map} \phi : B^2 \to \bar{\Omega}, \\ & s.t. \ (\tilde{t},\tilde{x}) \in \ Int(\phi(B^2)) \text{ and } f \circ \phi(\cdot,\cdot) \in C^1_W(B^2)\}. \end{aligned} \tag{27}$$

It can be proved that if $\bar{\Omega} = \bigcup_{i=1}^{l} Int(\phi_i(B^2))$ for some $\phi_1, \ldots, \phi_l$, then $f(\cdot,\cdot) \in C^1_W(\bar{\Omega})$ if and only if $f \circ \phi_i(\cdot,\cdot) \in C_W(B^2)$ for each $i = 1, \ldots, l$. Moreover, $C^1_W(\bar{\Omega})$ with norm

$$\|f(\cdot,\cdot)\|_{1,W} = \sum_{i=1}^{l} \|f \circ \phi_i(\cdot,\cdot)\|_{1,W} \tag{28}$$

is a Banach space (for more details, see [19]). Now, we show the space of all polynomials of total degree at most $2N$ on $\bar{\Omega}$ by $Pol(N,N,\bar{\Omega})$, that is,

$$Pol(N,N,\bar{\Omega}) = \{\eta(\tilde{t},\tilde{x}) = \sum_{i=0}^{N}\sum_{j=0}^{N} \gamma_{ij}\tilde{t}^i\tilde{x}^j : (\tilde{t},\tilde{x}) \in \bar{\Omega}, \gamma_{ij} \in \mathbb{R}\}.$$

Theorem 2. *For any $f(\cdot,\cdot) \in C^1_W(\bar{\Omega})$, there is a polynomial $\eta(\cdot,\cdot) \in Pol(N,N,\bar{\Omega})$ such that*

$$\|f(\cdot,\cdot) - \eta(\cdot,\cdot)\|_\infty \le \frac{c_0 c_1}{2N} W(\frac{1}{2N}), \tag{29}$$

where $c_1 = \|f(\cdot,\cdot)\|_{1,W}$ and c_0 is a constant that depends on $W(\cdot)$, but independent of N.

Proof. The proof is a result of Theorem 2.1 in [19]. □

Now, to guarantee the existence of solution for the system of algebraic equations (20), we relax it as follows:

$$\begin{cases} |\sum_{i=0}^{N} \bar{a}^N_{ik} D_{pi} - \psi_1\left(\bar{a}^N_{pk}, \bar{b}^N_{pk}, \sum_{i=0}^{N} \bar{a}^N_{ik} D_{pi}, \sum_{j=0}^{N} \bar{a}^N_{pj} D_{kj}, \sum_{j=0}^{N} \bar{a}^N_{pj} \hat{D}_{kj}\right)| \\ \qquad \le \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}), \quad p,k = 1,2,\ldots,N-1 \\ |\sum_{i=0}^{N} \bar{b}^N_{ik} D_{pi} - \psi_2\left(\bar{a}^N_{pk}, \bar{b}^N_{pk}, \sum_{i=0}^{N} \bar{b}^N_{ik} D_{pi}, \sum_{j=0}^{N} \bar{b}^N_{pj} D_{kj}, \sum_{j=0}^{N} \bar{b}^N_{pj} \hat{D}_{kj}\right)| \\ \qquad \le \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}), \quad p,k = 1,2,\ldots,N-1 \\ |\bar{a}^N_{p0}| \le \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}), |\bar{a}^N_{pN}| \le \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}), \quad p = 0,1,\ldots,N, \\ |\bar{b}^N_{p0}| \le \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}), |\bar{b}^N_{pN}| \le \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}), \quad p = 0,1,\ldots,N, \\ |\bar{a}^N_{0k} - Y_0(\bar{x}_k)| \le \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}), |\bar{b}^N_{Nk}| \le \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}), \quad k = 0,1,\ldots,N, \end{cases} \tag{30}$$

where N is sufficiently big and $W(\cdot)$ is a given modulus of continuity. Since $\lim_{N\to\infty} \frac{\sqrt{N}}{2N-1} W(\frac{1}{2N-1}) = 0$, any solution $(\bar{a},\bar{b})_N = ((\bar{a}^N_{pk},\bar{b}^N_{pk}); p,k = 0,1,\ldots,N)$ for system of algebraic inequalities (30) is a solution for system of algebraic equations (20) when N tends to infinity.

Now, we show that system (30) is feasible, that is, it has at least one solution $(\bar{a},\bar{b})_N$.

Theorem 3. *Let $(Y(\cdot,\cdot),P(\cdot,\cdot))$ be a solution for system (14), where $Y(\cdot,\cdot)$ and $P(\cdot,\cdot)$ are in $C^1_W(\bar{\Omega})$. Then there is a positive integer K such that, for any $N \geq K$, the system (30) has a solution as*

$$(\bar{a},\bar{b})_N = ((\bar{a}_{pk},\bar{b}_{pk}); p,k = 0,1,\ldots,N). \tag{31}$$

Moreover, $(\bar{a},\bar{b})_N$ satisfies

$$\begin{cases} |Y(\bar{t}_p,\bar{x}_k) - \bar{a}^N_{pk}| \leq \frac{L}{2N-1} W(\frac{1}{2N-1}), \ p,k = 0,1,\ldots,N, \\ |P(\bar{t}_p,\bar{x}_k) - \bar{b}^N_{pk}| \leq \frac{L}{2N-1} W(\frac{1}{2N-1}), \ p,k = 0,1,\ldots,N, \end{cases} \tag{32}$$

where L is a positive constant independent of N.

Proof. Assume that $\eta_1(\cdot,\cdot)$ and $\eta_2(\cdot,\cdot)$ in $Pol(N-1,N,\bar{\Omega})$ are the best polynomial approximations of $Y_{\bar{t}}(\cdot,\cdot)$ and $P_{\bar{t}}(\cdot,\cdot)$, respectively. By Theorem 2, we get

$$\begin{cases} \|Y_{\bar{t}}(\bar{t},\bar{x}) - \eta_1(\bar{t},\bar{x})\|_\infty \leq \frac{\gamma_1}{2N-1} W(\frac{1}{2N-1}), \quad (\bar{t},\bar{x}) \in \bar{\Omega}, \\ \|P_{\bar{t}}(\bar{t},\bar{x}) - \eta_2(\bar{t},\bar{x})\|_\infty \leq \frac{\gamma_2}{2N-1} W(\frac{1}{2N-1}), \quad (\bar{t},\bar{x}) \in \bar{\Omega}, \end{cases} \tag{33}$$

where γ_1 and γ_2 are two constants independent of N. We define

$$\begin{cases} \tilde{Y}(\bar{t},\bar{x}) = Y(-1,\bar{x}) + \int_{-1}^{\bar{t}} \eta_1(\tau,\bar{x}) d\tau, \quad (\bar{t},\bar{x}) \in \bar{\Omega} \\ \tilde{P}(\bar{t},\bar{x}) = P(-1,\bar{x}) + \int_{-1}^{\bar{t}} \eta_2(\tau,\bar{x}) d\tau, \quad (\bar{t},\bar{x}) \in \bar{\Omega}, \end{cases} \tag{34}$$

and

$$\bar{a}^N_{pk} = \tilde{Y}(\bar{t}_p,\bar{x}_k), \quad \bar{b}^N_{pk} = \tilde{P}(\bar{t}_p,\bar{x}_k), \qquad p,k = 0,1,\ldots,N. \tag{35}$$

We show that $(\bar{a},\bar{b})_N = ((\bar{a}^N_{pk},\bar{b}^N_{pk});\ p,k = 0,1,\ldots,N)$ satisfies system (30). By (33), (34), and (35), for $(\bar{t},\bar{x}) \in \bar{\Omega}$, we get

$$\begin{aligned} |Y(\bar{t},\bar{x}) - \tilde{Y}(\bar{t},\bar{x})| &= \left| \int_{-1}^{\tau} (Y_{\bar{t}}(\tau,\bar{x}) - \eta_1(\tau,\bar{x})) d\tau \right| \leq \int_{-1}^{\tau} |Y_{\bar{t}}(\tau,\bar{x}) - \eta_1(\tau,\bar{x})| \, d\tau \\ &\leq \frac{\gamma_1}{2N-1} W(\frac{1}{2N-1}) \int_{-1}^{\tau} d\tau \leq \frac{2\gamma_1}{2N-1} W(\frac{1}{2N-1}). \end{aligned} \tag{36}$$

Also, by a similar procedure, for $(\bar{t},\bar{x}) \in \bar{\Omega}$, we gain

$$|P(\bar{t},\bar{x})-\tilde{P}(\bar{t},\bar{x})| \le \frac{2\gamma_2}{2N-1}W(\frac{1}{2N-1}). \tag{37}$$

Now, by relation (34), the functions $\tilde{Y}(\cdot,\bar{x})$ and $\tilde{P}(\cdot,\bar{x})$, for any $\bar{x}\in[-1,1]$, are polynomials of total degree at most $2N$. Hence, their derivatives at CGL nodes $\bar{t}_0,\bar{t}_1,\ldots,\bar{t}_N$ are exactly equal to the value of polynomial at the nodes multiplied by the differential matrix D, defined by (19). Thus, we have

$$\sum_{i=0}^{N}\bar{a}_{ik}^{N}D_{pi}=\tilde{Y}_{\bar{t}}(\bar{t}_p,\bar{x}_k),\quad \sum_{i=0}^{N}\bar{b}_{ik}^{N}D_{pi}=\tilde{P}_{\bar{t}}(\bar{t}_p,\bar{x}_k);\quad p,k=0,1,\ldots,N. \tag{38}$$

Therefore, by relations (15) and (36), we get

$$\begin{aligned}
&\left|\sum_{i=0}^{N}a_{ik}^{N}D_{pi}-\psi_1\left(\bar{a}_{pk}^{N},\bar{b}_{pk}^{N},\sum_{j=0}^{N}\bar{a}_{pj}^{N}D_{kj},\sum_{j=0}^{N}\bar{a}_{pj}^{N}\hat{D}_{kj}\right)\right|\\
&\le\left|\tilde{Y}_{\bar{t}}(\bar{t}_p,\bar{x}_k)-Y_{\bar{t}}(\bar{t}_p,\bar{x}_k)\right|\\
&\quad+\left|Y_{\bar{t}}(\bar{t}_p,\bar{x}_k)-\psi_1\left(\bar{a}_{pk}^{N},\bar{b}_{pk}^{N},\sum_{j=0}^{N}\bar{a}_{pj}^{N}D_{kj},\sum_{j=0}^{N}\bar{a}_{pj}^{N}\hat{D}_{kj}\right)\right|\\
&=|\eta_1(\bar{t}_p,\bar{x}_k)-Y_{\bar{t}}(\bar{t}_p,\bar{x}_k)|\\
&\quad+\Bigg|\psi_1\left(Y(\bar{t}_p,\bar{x}_k),P(\bar{t}_p,\bar{x}_k),Y_{\bar{x}}(\bar{t}_p,\bar{x}_k),Y_{\bar{x}\bar{x}}(\bar{t}_p,\bar{x}_k)\right)\\
&\qquad-\psi_1\left(\bar{a}_{pk}^{N},\bar{b}_{pk}^{N},\sum_{j=0}^{N}\bar{a}_{pj}^{N}D_{kj},\sum_{j=0}^{N}\bar{a}_{pj}^{N}\hat{D}_{kj}\right)\Bigg|\\
&\le|\eta_1(\bar{t}_p,\bar{x}_k)-Y_{\bar{t}}(\bar{t}_p,\bar{x}_k)|+M_1\left(\left|Y(\bar{t}_p,\bar{x}_k)-\bar{a}_{pk}^{N}\right|+\left|P(\bar{t}_p,\bar{x}_k)-\bar{b}_{pk}^{N}\right|\right)\\
&\le\frac{\gamma_1}{2N-1}W(\frac{1}{2N-1})+M_1\left(\frac{2\gamma_1}{2N-1}W(\frac{1}{2N-1})+\frac{2\gamma_2}{2N-1}W(\frac{1}{2N-1})\right)\\
&=\frac{\gamma_1(2M_1+1)+2M_1\gamma_2}{2N-1}W(\frac{1}{2N-1}),\quad p,k=1,\ldots,N-1,
\end{aligned}\tag{39}$$

where M_1 and M_2 are Lipschitz constants of ψ_1 and ψ_2, respectively. Moreover, by a similar process, we obtain

$$\begin{aligned}
&\left|\sum_{i=0}^{N}b_{ik}^{N}D_{pi}-\psi_2\left(\bar{a}_{pk}^{N},\bar{b}_{pk}^{N},\sum_{j=0}^{N}\bar{b}_{pj}^{N}D_{kj},\sum_{j=0}^{N}\bar{b}_{pj}^{N}\hat{D}_{kj}\right)\right| \qquad (40)\\
&\le\frac{\gamma_2(2M_2+1)+2M_2\gamma_1}{2N-1}W(\frac{1}{2N-1}),\quad p,k=1,\ldots,N-1.
\end{aligned}$$

Further, for boundary conditions, we get

$$\left|\tilde{Y}(-1,\bar{x}_k)-Y_0(\bar{x}_k)\right| \le \left|\tilde{Y}(-1,\bar{x}_k)-Y(-1,\bar{x}_k)\right| + \left|Y(-1,\bar{x}_k)-Y_0(\bar{x}_k)\right|$$

$$\le \frac{2\gamma_1}{2N-1}W\left(\frac{1}{2N-1}\right), \quad k=0,1,\ldots,N. \tag{41}$$

Also, for all $p,k=0,1,\ldots,N$, we have

$$\left|\tilde{P}(1,\bar{x}_k)\right| = \left|\tilde{P}(1,\bar{x}_k)-P(1,\bar{x}_k)\right| \le \frac{2\gamma_2}{2N-1}W\left(\frac{1}{2N-1}\right), \tag{42}$$

$$\left|\tilde{Y}(\bar{t}_p,-1)\right| = \left|\tilde{Y}(\bar{t}_p,-1)-Y(\bar{t}_p,-1)\right| \le \frac{2\gamma_1}{2N-1}W\left(\frac{1}{2N-1}\right), \tag{43}$$

$$\left|\tilde{Y}(\bar{t}_p,1)\right| = \left|\tilde{Y}(\bar{t}_p,1)-Y(\bar{t}_p,1)\right| \le \frac{2\gamma_1}{2N-1}W\left(\frac{1}{2N-1}\right), \tag{44}$$

$$\left|\tilde{P}(\bar{t}_p,-1)\right| = \left|\tilde{P}(\bar{t}_p,-1)-P(\bar{t}_p,-1)\right| \le \frac{2\gamma_2}{2N-1}W\left(\frac{1}{2N-1}\right), \tag{45}$$

$$\left|\tilde{P}(\bar{t}_p,1)\right| = \left|\tilde{P}(\bar{t}_p,1)-P(\bar{t}_p,1)\right| \le \frac{2\gamma_2}{2N-1}W\left(\frac{1}{2N-1}\right). \tag{46}$$

Hence, if we select K such that

$$\max\{\gamma_1(2M_1+1)+2M_1\gamma_2, \gamma_2(2M_2+1)+2M_2\gamma_1, 2\gamma_1, 2\gamma_2\} \le \sqrt{N},$$

for $N \ge K$, then by (39)–(46), pair $(\bar{a},\bar{b})_N$ satisfies system (30). □

Now, we show that the sequence of solutions of problem (30) and sequence of their interpolating polynomials converge to the solution of the problem (14).

Theorem 4. *Let* $\{(\bar{a},\bar{b})_N = ((\bar{a}^N_{pk},\bar{b}^N_{pk});\ p,k=0,1,\ldots,N\}_{N=K}^{\infty}$ *be a sequence of solution of system* (30) *and let* $\{(Y^N(\cdot,\cdot),P^N(\cdot,\cdot))\}_{N=K}^{\infty}$ *be their interpolating polynomials sequence defined by* (16). *Also, we assume that, for any* $\bar{x}$ *in* $[-1,-1]$, *the sequence* $\{(Y^N(-1,\bar{x}),P^N(-1,\bar{x}),Y^N_{\bar{t}}(\cdot,\cdot),P^N_{\bar{t}}(\cdot,\cdot))\}_{N=K}^{\infty}$ *has a subsequence* $\{(Y^{N_i}(-1,\bar{x}),P^{N_i}(-1,\bar{x}),Y^{N_i}_{\bar{t}}(\cdot,\cdot),P^{N_i}_{\bar{t}}(\cdot,\cdot))\}_{i=0}^{\infty}$ *that uniformly converges to* $(\phi_1^{\infty}(\bar{x}),\phi_2^{\infty}(\bar{x}),\lambda_1(\cdot,\cdot),\lambda_2(\cdot,\cdot))$ *where* $\lambda_1(\cdot,\cdot),\lambda_2(\cdot,\cdot) \in C^2(\bar{\Omega})$, $\phi_1^{\infty}(\cdot)$, $\phi_2^{\infty}(\cdot) \in C^2([-1,1])$ *and* $\lim_{i\to\infty} N_i = \infty$. *Then the pair*

$$(\tilde{Y}(\bar{t},\bar{x}),\tilde{P}(\bar{t},\bar{x})) = \left(\lim_{i\to\infty} Y^{N_i}(\bar{t},\bar{x}), \lim_{i\to\infty} P^{N_i}(\bar{t},\bar{x})\right), \tag{47}$$

for $(\bar{t},\bar{x}) \in \bar{\Omega}$, *is a solution of system* (14).

Proof. By attention to the assumptions, we have

$$\begin{cases} \tilde{Y}(\bar{t},\bar{x}) = \phi_1^{\infty}(\bar{x}) + \int_{-1}^{\bar{t}} \lambda_1(\tau,\bar{x})d\tau, \\ \tilde{P}(\bar{t},\bar{x}) = \phi_2^{\infty}(\bar{x}) + \int_{-1}^{\bar{t}} \lambda_2(\tau,\bar{x})d\tau. \end{cases} \tag{48}$$

We first show that $(\tilde{Y}(\bar{t},\bar{x}),\tilde{P}(\bar{t},\bar{x}))$ for $\bar{t}\in[-1,1]$ and $\bar{x}=x_k$, $k=0,1,\ldots,N$ satisfy system (14). Assume that $(\tilde{Y}(\cdot,\bar{x}_k),\tilde{P}(\cdot,\bar{x}_k))$ for some $k=1,\ldots,N-1$ does not satisfy the first equation of (14). Then, there is a τ in $(-1,1)$ such that

$$\tilde{Y}_{\bar{t}}(\tau,\bar{x}_k)-\psi_1\left(\tilde{Y}(\tau,\bar{x}_k),\tilde{P}(\tau,\bar{x}_k),\tilde{Y}_{\bar{x}}(\tau,\bar{x}_k),\tilde{Y}_{\bar{x}\bar{x}}(\tau,\bar{x}_k)\right)\neq 0.$$

Since CGL nodes $\{\bar{t}_p\}_{p=0}^{N}$ when $N\to\infty$ are dense in $[-1,1]$, there is a sequence $\{\bar{t}_{l_{N_i}}\}_{i=1}^{\infty}$ such that $0<l_{N_i}<N_i$ and $\lim_{i\to\infty}\bar{t}_{l_{N_i}}=\tau$. Thus

$$\begin{aligned}\lim_{i\to\infty}\left(\tilde{Y}_{\bar{t}}(\bar{t}_{l_{N_i}},\bar{x}_k)-\psi_1\left(\tilde{Y}(\bar{t}_{l_{N_i}},\bar{x}_k),\tilde{P}(\bar{t}_{l_{N_i}},\bar{x}_k),\tilde{Y}_{\bar{x}}(\bar{t}_{l_{N_i}},\bar{x}_k),\tilde{Y}_{\bar{x}\bar{x}}(\bar{t}_{l_{N_i}},\bar{x}_k)\right)\right)\\ =\tilde{Y}_{\bar{t}}(\tau,\bar{x}_k)-\psi_1\left(\tilde{Y}(\tau,\bar{x}_k),\tilde{P}(\tau,\bar{x}_k),\tilde{Y}_{\bar{x}}(\tau,\bar{x}_k),\tilde{Y}_{\bar{x}\bar{x}}(\tau,\bar{x}_k)\right)\neq 0. \quad (49)\end{aligned}$$

On the other hand, since $\lim_{i\to\infty}\frac{\sqrt{N_i}}{2N_i-1}W(\frac{1}{2N_i-1})=0$, by (30), we obtain

$$\lim_{i\to\infty}\left(\tilde{Y}_{\bar{t}}(\bar{t}_{l_{N_i}},\bar{x}_k)-\psi_1\left(\tilde{Y}(\bar{t}_{l_{N_i}},\bar{x}_k),\tilde{P}(\bar{t}_{l_{N_i}},\bar{x}_k),\tilde{Y}_{\bar{x}}(\bar{t}_{l_{N_i}},\bar{x}_k),\tilde{Y}_{\bar{x}\bar{x}}(\bar{t}_{l_{N_i}},\bar{x}_k)\right)\right)=0,$$

which is a contradiction to (49). Thus $(\tilde{Y}(\bar{t},\bar{x}),\tilde{P}(\bar{t},\bar{x}))$ (for all $\bar{t}\in[-1,1]$ and $\bar{x}=\bar{x}_k$, $k=1,\ldots,N-1$) satisfies the first equation of (14). By a similar procedure, we can show that it satisfies the second equation. Also, it can be easily proved that $(\tilde{Y}(\cdot,\bar{x}_k),\tilde{P}(\cdot,\bar{x}_k))$, for $k=0,1,\ldots,N$, satisfies the boundary conditions. For example we show that $\tilde{Y}(-1,\bar{x}_k)=Y_0(\bar{x}_k)$ for $k=0,1,\ldots,N$. We have

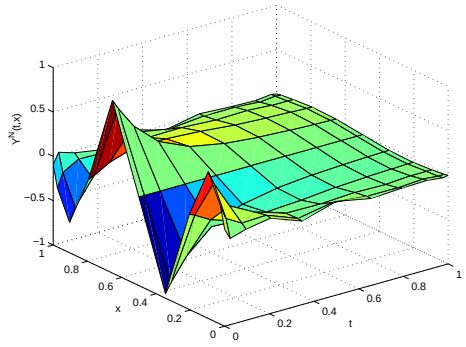
$$\begin{aligned}0\le|\tilde{Y}(-1,\bar{x}_k)-Y_0(\bar{x}_k)|&=|\lim_{i\to\infty}Y^{N_i}(-1,\bar{x}_k)-Y_0(\bar{x}_k)|\\ &=\lim_{i\to\infty}|Y^{N_i}(-1,\bar{x}_k)-Y_0(\bar{x}_k)|=\lim_{i\to\infty}|\bar{a}_{0k}^{N_i}-Y_0(\bar{x}_k)|\\ &\le\lim_{i\to\infty}\frac{\sqrt{N_i}}{2N_i-1}W(\frac{1}{2N_i-1})=0.\end{aligned}$$

Hence $\tilde{Y}(-1,\bar{x}_k)=Y_0(\bar{x}_k)$ for all $k=0,1,\ldots,N$. Now, we know that nodes $\{\bar{x}_k\}_{k=0}^{N}$ when $N\to\infty$, are dense in $[-1,1]$. Therefore the pair $(\tilde{Y}(\cdot,\cdot),\tilde{P}(\cdot,\cdot))$ defined by (47) is a solution for (14) on $\bar{\Omega}=[-1,1]\times[-1,1]$. □

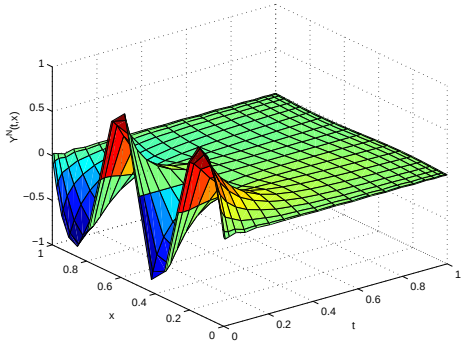
6 Numerical examples

In the following examples, we use the Levenberg–Marquardt method (a quasi-Newton method) for FSOLVE command in MATLAB software to solve algebraic system (20).

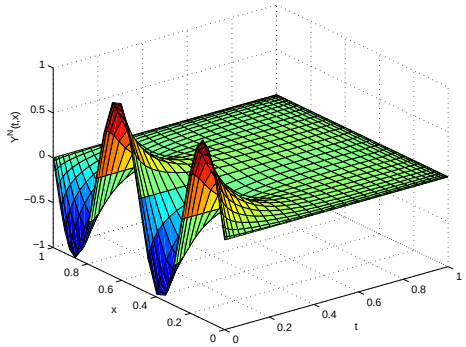
Example 1. Consider problems (1)–(4), where $T = 1$, $\alpha = 1$, $\nu = 0.01$ and 0.05, $y_0 = \sin(4\pi x)$, $\Phi(u) = u$, and $z(t, x) = 0$. The approximate optimal value of objective function computed by the CPS and LPS [20] methods for $\nu = 0.01$ and 0.05 and $N = 10,\ 20,\ 30$ and 40 are shown in Table 1. We observe that our numerical results are better than the results of the LPS method. In Figures 1 and 2, we show the obtained approximate optimal state and control for $\nu = 0.05$ and different values of N.



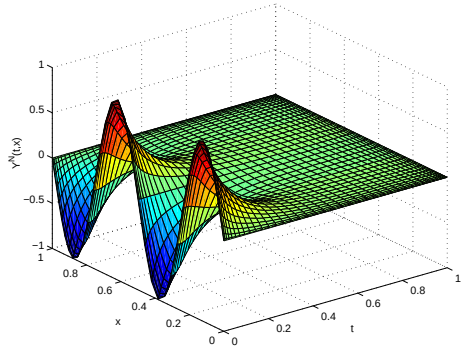
(a) The approximate optimal state for $N = 10$ and $\nu = 0.05$



(b) The approximate optimal state for $N = 20$ and $\nu = 0.05$

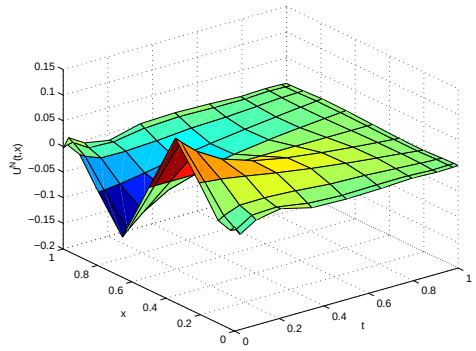


(c) The approximate optimal state for $N = 30$ and $\nu = 0.05$

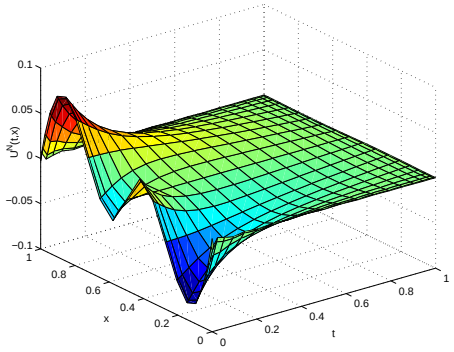


(d) The approximate optimal state for $N = 40$ and $\nu = 0.05$

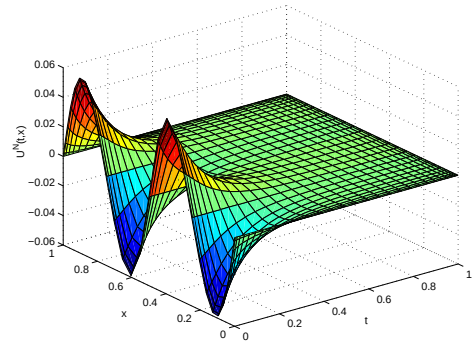
Figure 1: The approximate optimal state for Example 1



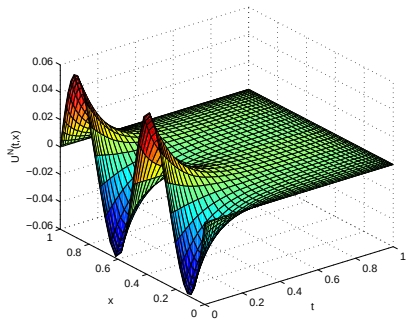
(a) The approximate optimal control for $N = 10$ and $\nu = 0.05$



(b) The approximate optimal control for $N = 20$ and $\nu = 0.05$



(c) The approximate optimal control for $N = 30$ and $\nu = 0.05$



(d) The approximate optimal control for $N = 40$ and $\nu = 0.05$

Figure 2: The approximate optimal control for Example 1

Table 1: Comparison of objective function values for Example 1

	$\nu = 0.01$	$\nu = 0.01$	$\nu = 0.05$	$\nu = 0.05$
N	Presented method	LPS method [20]	Presented method	LPS method [20]
10	0.033014881174862	0.0828638100277	0.016911085761598	0.01590006952876
20	0.040440356851832	0.0620867108909	0.013730446517693	0.01519309308076
30	0.029728232559725	0.0466282421253	0.015082505388501	0.01519227846689
40	0.029005091596013	0.0463124455511	0.015073940981453	0.01519176630695

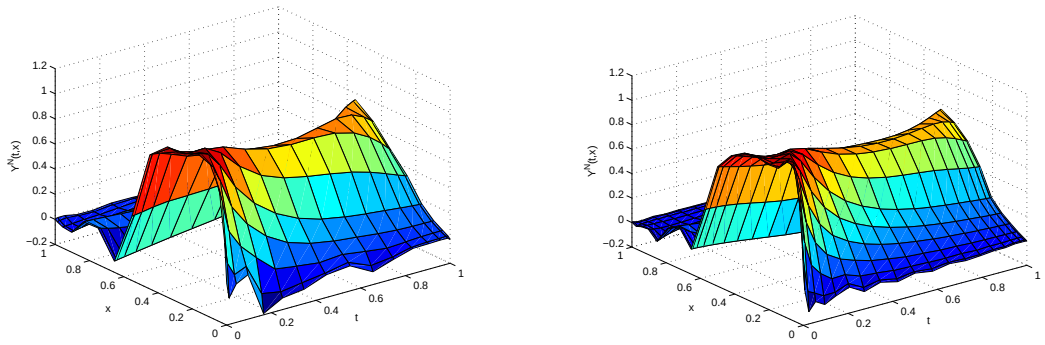
Example 2. Consider optimal control problem (1)–(4). Let $T = 1$, $\alpha = 0.05$, $\nu = 0.01$, and $z(t, x) = y_0(x)$ where

$$y_0(x) = \begin{cases} 1, & x \in (0, \frac{1}{2}], \\ 0 & otherwise. \end{cases}$$

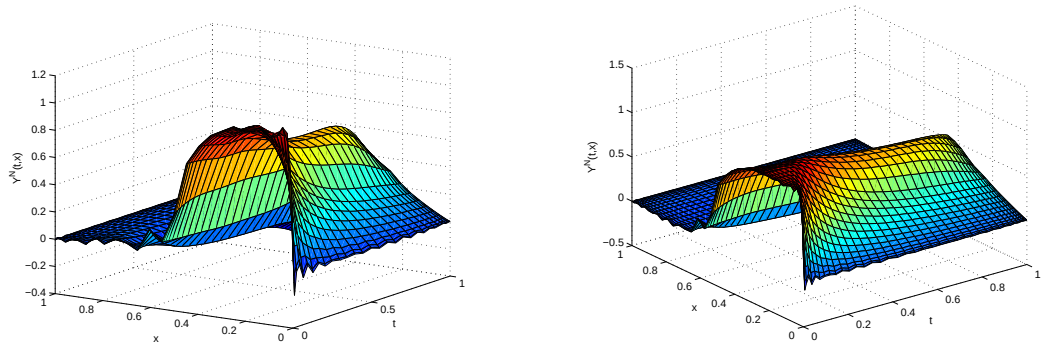
is the initial condition and

$$\Phi(u) = \begin{cases} u, & u \text{ on } (0,T) \times (\frac{1}{4}, \frac{3}{4}), \\ 0, & \text{otherwise.} \end{cases}$$

Table 2 shows the approximate optimal values of objective function for different values of N by the CPS method. In Figures 3 and 4, the optimal state and optimal control are presented, respectively.

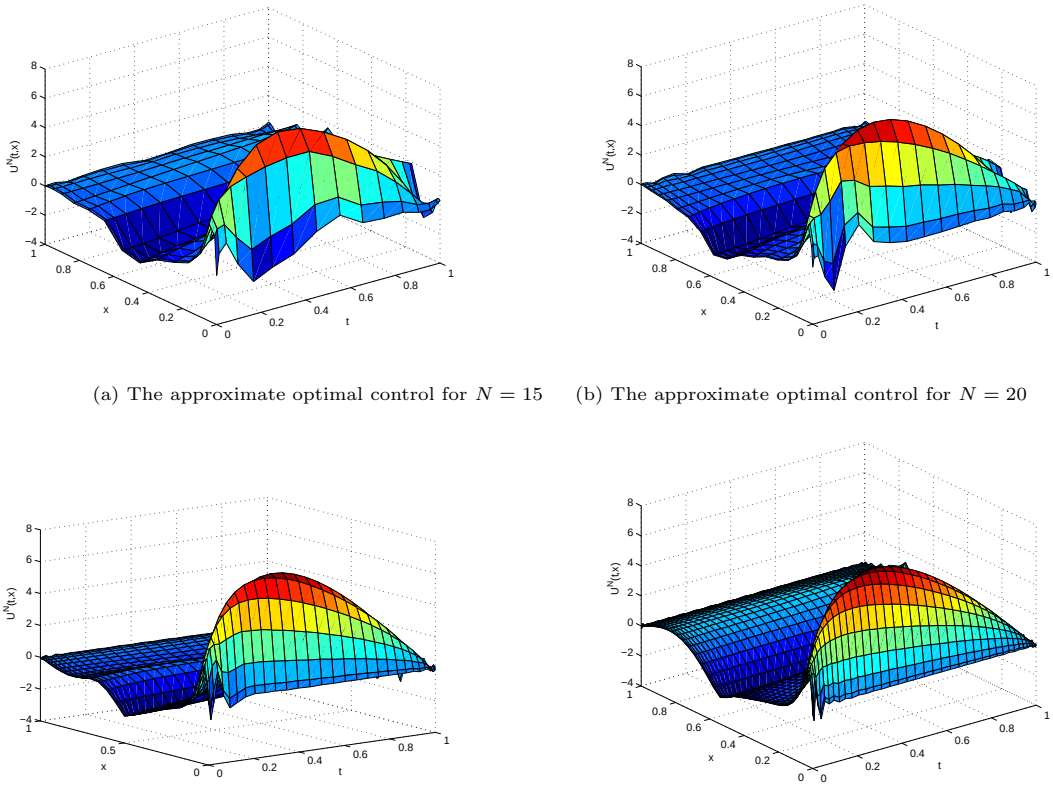


(a) The approximate optimal state for $N = 15$ (b) The approximate optimal state for $N = 20$



(c) The approximate optimal state for $N = 30$ (d) The approximate optimal state for $N = 40$

Figure 3: The approximate optimal state for Example 2



(a) The approximate optimal control for $N = 15$ (b) The approximate optimal control for $N = 20$

(c) The approximate optimal control for $N = 30$ (d) The approximate optimal control for $N = 40$

Figure 4: The approximate optimal control for Example 2

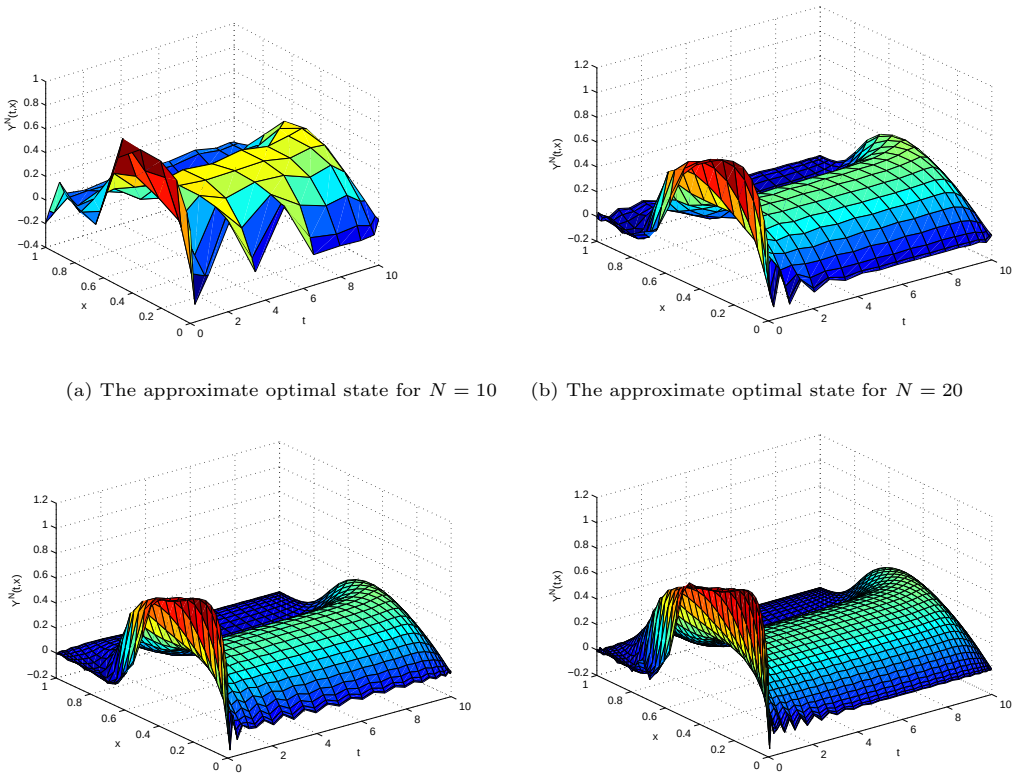
Table 2: Approximate values of objective functional for Example 2

N	J_N
20	0.106416588965259
25	0.085739365971895
30	0.083129631634715

Example 3. Consider optimal control problem (1)–(4). Assume that $T = 10$, $\alpha = 1$, $\Phi(u) = u$, and $z(t, x) = y_0(x)$ where y_0 is defined in Example 2. We apply the CPS method to this problem. In Table 3, the values of cost functional using CPS method for different values of N are listed. In Figures 5 and 6, the approximate optimal state and optimal control are illustrated, respectively.

Table 3: Approximate values of objective functional for Example 3

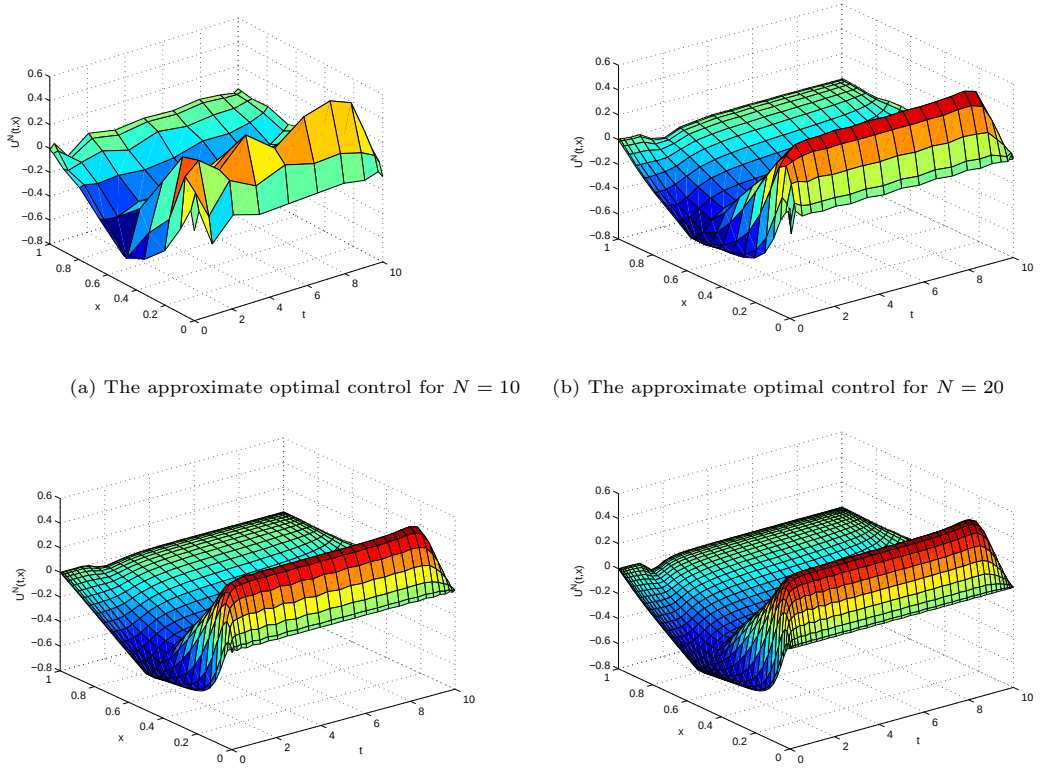
N	J_N
10	0.795502231705757
20	0.732649373291908
30	0.605003194783743



(a) The approximate optimal state for $N = 10$ (b) The approximate optimal state for $N = 20$

(c) The approximate optimal state for $N = 30$ (d) The approximate optimal state for $N = 40$

Figure 5: The approximate optimal state for Example 3



(a) The approximate optimal control for $N = 10$ (b) The approximate optimal control for $N = 20$

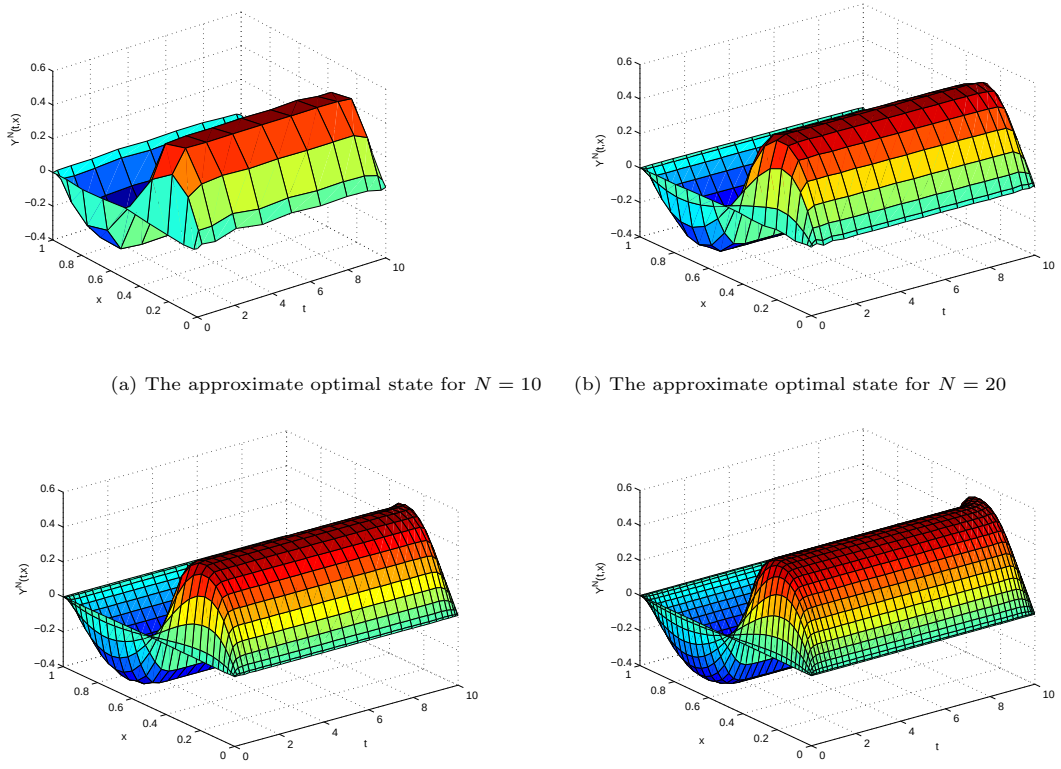
(c) The approximate optimal control for $N = 30$ (d) The approximate optimal control for $N = 40$

Figure 6: The approximate optimal control for Example 3

Example 4. Consider the optimal control of Burgers' equation (1)–(4) with $T = 10$, $\alpha = 0.1$, $\nu = 0.01$, $\Phi(u) = u$, and the desired state $z(t, x) = y_0(x)$ where $y_0 = \exp(-x)\sin(2\pi x)$, $x \in [0, 1]$. The numerical results are displayed in Table 4 for different values of N by using our method. In Figures 7 and 8, we show the approximate optimal state and control, respectively.

Table 4: Approximate values of objective functional for Example 4

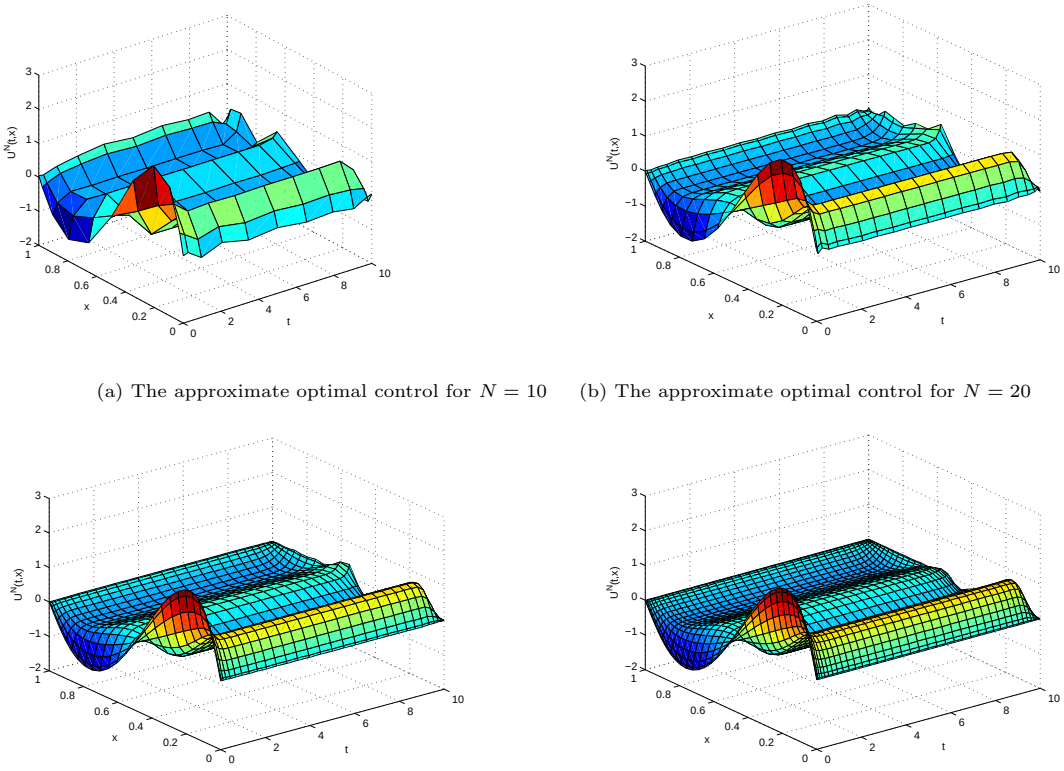
N	J_N
20	0.121081851037199
30	0.116980721067958
40	0.115852005217238



(a) The approximate optimal state for $N = 10$ (b) The approximate optimal state for $N = 20$

(c) The approximate optimal state for $N = 30$ (d) The approximate optimal state for $N = 40$

Figure 7: The approximate optimal state for Example 4



(a) The approximate optimal control for $N = 10$ (b) The approximate optimal control for $N = 20$

(c) The approximate optimal control for $N = 30$ (d) The approximate optimal control for $N = 40$

Figure 8: The approximate optimal control for Example 4

7 Conclusions

In this paper, we proposed an efficient Chebyshev pseudo spectral method to solve the optimal control problem governed by Burgers' equation. By applying this method, we discretized the optimality conditions and obtained a system of algebraic equations. We achieved a good approximate optimal solution with good accuracy. We analyzed the feasibility and convergence of the presented method.

References

1. Abramowitz, M. and Stegun, I. A. *Handbook of Mathematical Functions*, Inc., New York, 1964.

2. Allahverdi, N., Pozo, A. and Zuazua, E. *Numerical aspects of large-time optimal control of Burgers' equation*, M2AN Math. Model. Numer. Anal. 50(5) (2016), 1371–1401.

3. Baumann, M. *Nonlinear Model Order Reduction using POD/DEIM for Optimal Control of Burgers' Equation*, Thesis submitted to the Delft Institute of Applied Mathematics, 2013.

4. Chorin, A.J. and Marsden, J.E. *A Mathematical Introduction to Fluid Mechanics*, Springer-Verlag New York, 1979.

5. De los Reyes, J.C. and Kunisch, K. *A comparison of algorithms for control constrained optimal control of the Burgers' equation*, Calcolo, 41 (2004), 203–225.

6. Fernandez-Cara, E. and Guerrero, S. *Null controllability of the Burgers' system with distributed controls*, Systems Control Lett. 56 (5) (2007), 366–372.

7. Gong, Q., Ross, I.M., Kang, W. and Fahroo, F. *Connections between the covector mapping theorem and convergence of pseudo-spectral methods for optimal control*, Comput Optim Appl. 41 (2008), 307–335.

8. Hashemi, S.M. and Werner, H. *LPV Modelling and Control of Burgers' Equation*, The International Federation of Automatic Control Milano (Italy) August 28–September 2, 2011.

9. Hinze, M. and Volkwein, S. *Analysis of instantaneous control for the Burgers' equation*, Nonlinear Anal. 50 (2002), 1–26.

10. Jaluria, Y. and Torrance, K.K.E. *Computational heat transfer*, Series in Computational and Physical Processes in Mechanics and Thermal Sciences Series. Taylor and Francis Group, 2003.

11. Karasozen, B. and Yilmaz, F. *Optimal boundary control of the unsteady Burgers' equation with simultaneous space-time discretization*, Optimal Control Appl. Methods. 35 (2014), 423–434.

12. Kobayashi, T. *Adaptive regulator design of a viscous Burgers' system by boundary control*, IMA J. Math. Control Inform. 18 (3) (2001), 427–437.

13. Kucuk, I. and Sadek, I. *A numerical approach to an optimal boundary control of the viscous Burgers' equation*, Int. J. Appl. Math. Comput. 210 (2009), 126–135.

14. Kunisch, K. and Volkwein, S. *Control of the Burgers' equation by a reduced-order approach using proper orthogonal decomposition*, J. Optim. Theory Appl. 102 (1999), 345–371.

15. Marburger, J. and Pinnau, R. *Optimal control for Burgers' equation using particle methods*, arxiv:1309.7619, 2013.

16. Noack, A. and Walther, A. *Adjoint concepts for the optimal control of Burgers' equation*, Comput. Optim. Appl. 36 (2007), 109–133.

17. Peterson, A.F., Ray, S.L. and Mittra, R. *Computational Methods for Electromagnetics*, IEEE Press Series on Electromagnetic Wave Theory. Wiley, 1998.

18. Press, W.H., Flannery, B.P., Teutolsky, S.A. and Vetterling, W.T. *Numerical Recipes*, The Art of Scientific Computing. Cambridge University Press, Cambridge, 1990.

19. Ragozin, D.L. *Polynomial approximation on compact manifolds and homogeneous spaces*, Trans. Amer. Math. Soc. 150, 1970.

20. Sabeh, Z., Shamsi, M. and Dehghan, M. *Distributed optimal control of the viscous Burgers' equation via a Legendre pseudo-spectral approach*, Math. Methods Appl. Sci. 39 (12) (2016), 3350–3360.

21. Sadek, I. and Kucuk, I. *A robust technique for solving optimal control of coupled Burgers' equations*, IMA J. Math. Control Inform. 28 (3) (2011), 239–250.

22. Singh, S., Patel, V.K., Singh, V.K. and Tohidi, E. *Application of Bernoulli matrix method for solving two-dimensional hyperbolic telegraph equations with Dirichlet boundary conditions*, Comput. Math. Appl. 75 (7) (2018), 2280–2294.

23. Smaoui, N., Zribi, M. and Almulla, A. *Sliding mode control of the forced generalized Burgers' equation*, IMA J. Math. Control Inform. 23 (3) (2006), 301–323.

24. Taflove, A. and Hagness, S.C. *Computational Electrodynamics: The Finite- Difference Time- Domain Method*, Artech House, third edition, 2005.

25. Troltzsch, F. and Volkwein, S. *The SQP method for control constrained optimal control of the Burgers' equation*, ESAIM Control Optim. Calc. Var. 6 (2001) 649–674.

26. Volkwein, S. *Mesh-independence of an augmented Lagrangian-SQP method in Hilbert spaces and control problems for the Burgers' equation*, Dissertation at the University of Technology Berlin, 1997.

27. Volkwein, S. *Distributed control problems for the Burgers' equation*, Comput. Optim. Appl. 18 (2) (2001), 115–140.

28. Wesseling, P. *Principles of Computational Fluid Dynamics*, Springer-Verlag Berlin Heidelberg, 2001.

29. Yilmaz, F. and Karasozen, B. *Solving optimal control problems for the unsteady Burgers' equation in COMSOL Multiphysics*, J. Comput. Appl. Math. 235 (2011), 4839–4850.

30. Yilmaz, F. and Karasozen, B. *An all-at-once approach for the optimal control of the unsteady Burgers' equation*, J. Comput. Appl. Math. 259 (2014), 771–779.

31. Zeng, M.L. and Zhang, G.F. *Preconditioning optimal control of the unsteady Burgers' equations with H_1 regularized term*, Appl. Math. Comput. 254 (2015), 133–147.

32. Zogheib, B. and Tohidi, E. *An Accurate Space-Time Pseudo-spectral Method for Solving Nonlinear Multi-Dimensional Heat Transfer Problems*, Mediterr. J. Math. 14 (1) (2017).

33. Zogheib, B. and Tohidi, E. *Modal Hermite spectral collocation method for solving multi-dimensional hyperbolic telegraph equations*, Comput. Math. Appl. 75 (2018), 3571–3588.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 103–121
DOI:10.22067/ijnao.v9i2.70045

Research Article

Capturing outlines of generic shapes with cubic Bézier curves using the Nelder–Mead simplex method

A. Ebrahimi, G. B. Loghmani* and M. Sarfraz

Abstract

We design a fast technique for fitting cubic Bézier curves to the boundary of 2D shapes. The technique is implemented by means of the Nelder–Mead simplex procedure to optimize the control points. The natural attributes of the Bézier curve are utilized to discover the initial vertex points of the Nelder–Mead procedure. The proposed technique is faster than traditional methods and helps to obtain a better fit with a desirable precision. The comparative analysis of our results describes that the introduced approach has a high compression ratio and a low fitting error.

AMS(2010): 65D17; 65D10.

Keywords: Interpolation; Splines; Curve fitting; Nelder–Mead simplex method; Computer aided design; Computer graphics.

1 Introduction

Capturing the outlines of 2D objects is a substantial topic in computer aided geometric designs (CAGD), computer graphics, as well as vision and imaging; see [1, 2, 6, 7, 9–12, 15, 20–25, 32, 38, 39]. Curve fitting with Bézier curves is a traditional problem in this process. The goal of a curve fitting method is to

*Corresponding author
Received 8 January 2018; revised 12 January 2019; accepted 8 March 2019
Alireza Ebrahimi
Computer Geometry and Dynamical Systems Laboratory, Faculty of Mathematical Sciences, Yazd University,Yazd, Iran. e-mail: a.ebrahimi@stu.yazd.ac.ir

Ghasem Barid Loghmani
Computer Geometry and Dynamical Systems Laboratory, Faculty of Mathematical Sciences, Yazd University,Yazd, Iran. e-mail: loghmani@yazd.ac.ir

Muhammad Sarfraz
Department of Information Science, College of Computing Sciences & Engineering, Kuwait University, Kuwait. e-mail: prof.m.sarfraz@gmail.com, muhammad.sarfraz@ku.edu.kw

detect a collection of the control points that can precisely indicate the given target shape.

In the literature, many methods have been suggested to represent outline objects [3, 11, 21, 29, 34]. A vast majority of these methods pertain to an optimization problem that finds an appropriate curve for the data generated by the outlines of 2D shapes. Most common algorithms for solving this model are not suitable optimization algorithms. Since they involve complex computations, they cannot be used for real time applications. Itoh and Ohno [11] exerted the least square fitting to approximate cubic Bézier curves without fixing the end points of the curves. Plass and Stone [21] proposed an iterative procedure for fitting a parametric piecewise cubic polynomial curve with an selective endpoint and tangent vector specifications. Srafraz and Khan [28] employed the least square method to certify an appropriate fit. Their method contains several phases, such as extraction of boundaries, exploration corner points and break points, and fitting curves. In another study, the same authors added reparameterization steps to ameliorate the efficiency of the fit [29]. Sarfraz and Razzak [34] employed a generalized Hermite cubic spline to capture the outline of digital character images using characteristic points and the least squares method. The enhanced Bézier curve scheme proposed by Sohel et al. [36] reduces the distance among the Bézier curve and its control polygon without extra computational complexity. An object coding technique using Bézier curves was introduced by Masood and Haq [16]. They determined the control points by searching along the endpoint tangents. Sarfraz and Masood [30] determined the appropriate position of the control points by using the natural attributes of cubic Bézier curves. Masood and Sarfraz [18] presented an outline capturing scheme using Bézier cubic approximation. Their method operates in two phases to find intermediate control points. In Phase 1, the location of the detected control points gets closer to that of the original control points. Phase 2 is exerted to hit the target location very exactly. Masood and Sarfraz [17] used the properties of cubic Bézier curves and analysed the control points spread (CPspread) to subdivide complex segments into two or more segments. Some other techniques based on the search algorithm include evolutionary algorithm [35], simulated annealing approach [26, 27], genetic algorithm [32, 33], and wavelets [37].

In the above mentioned methods, the control points are specified by the least square algorithm or using the properties of cubic Bézier curves. These processes are computationally expensive and not appropriate for practical applications. Our contribution in this study is to introduce a scheme to avoid these time consuming operations and to guarantee a desirable equivalence between compression ratios and fitting errors. We perform the Nelder–Mead algorithm to find the intermediate control points of the cubic Bézier curve. The initial situation of the control points is specified using extracting some effective virtues of the Bézier curve.

The rest of the paper is arranged as follows. Section 2 reviews the segmentation of an outline by corner detection. Section 3 introduces the Nelder–

Mead algorithm. The process of determining the control points for the cubic Bézier curve approximation using the Nelder–Mead simplex method is described in section 4. Section 5 illustrates the experimental observations and compares them with those of other methods. Finally, the paper is closed up with a conclusion in Section 6.

2 Outline segmentation

Outline segmentation is used to divide the shape outline into small part and simplify the curve fitting procedure. We use the corner points to partition the outline into disparate segments from the natural break points. Authors have suggested various corner detection algorithms in the literatures [3,4,31]. These algorithms do curvature analysis with numerical techniques. In this paper, the algorithm designed by Sarfraz et al. [31] is used for the corner detection, because this algorithm is accurate, effective, and robust to noise. The method is concisely explained in this section (readers are referred to [31] for details). The algorithm detects corner points in two steps. Candidate corner points are discovered from the outline data points in the first step. If all the boundary points are $Q_i, 1 \leq i \leq n$, then the boundary point Q_k can be given as

if $(i + L) \leq n$,
then $Q_k = Q_{i+L}$,
else $Q_k = Q_{(i+L)-n}$,

where L represents the length parameter and preserves of the shape scaling and resolution. The default value of L is 14. The perpendicular distance d_j from point $Q_j(x, y)$ to the direct line joining the points $Q_i(x, y)$ and $Q_k(x, y)$, can be given as follows:

if $m_x = 0$,
then $d_j = |Q_{j,x} - Q_{i,x}|$,
else $d_j = \dfrac{|Q_{j,y} - mQ_{j,x} + mQ_{i,x} - Q_{i,y}|}{\sqrt{m^2 + 1}}$,
where $m = \dfrac{m_y}{m_x} = \dfrac{Q_{k,y} - Q_{i,y}}{Q_{k,x} - Q_{i,x}}$.

The point Q_j is specified as a volunteer corner point if its perpendicular distance (d_j) from the direct line Q_iQ_k goes beyond D. The local sharpness and the opening angle of the corners are investigated by the distance parameter D. It also checks the incorrect determination of corner points that may occur due to noise and disarray. The default value of D is set to be 2.6. The new direct line through increasing both i and k detects the next volunteer

corner points. In the second step, the superfluous corner points are determined. A superfluous corner point is one of that any other volunteer with a higher value of d_j is in the domain R. In the other words, a candidate corner point will stay if it has the greatest value of d_j between the R number of points on its both sides. The default value of R is equal to length parameter. In the present study, we utilize this method at its default values for corner detection. All steps of this corner detection method is shown in Algorithm 1.

Algorithm 1 The corner detection algorithm

Set values of L, D, and R;
Detect boundary points $\{Q_i\}_{i=0}^{n}$ of 2D object using the Canny edge detector;
First step: detect candidate corner points;
for $i = 1$ **to** n **do**
  if $(i + L) \leq n$ **then**
    $Q_k = Q_{i+L}$,
  else
    $Q_k = Q_{(i+L)-n}$;
  end if
  for $j = i$ **to** k **do**
    Calculate d_j the perpendicular distance from point Q_j to the straight line joining points Q_i and Q_k;
    if $d_j > D$ **then**
      Add Q_j to candidate corner point;
    end if
  end for
end for
Second step: detect superfluous corner points;
A candidate corner point will stay if it has the greatest value of d_j between the R number of points on its both sides;

3 The Nelder–Mead simplex method (NMSM)

The Nelder–Mead algorithm is a numerical optimization procedure for solving unconstrained optimization problems in a multidimensional space; see [5,14,19]. This method is one of the enormously famous derivative-free techniques. It is a direct search algorithm that only needs a numerical evaluation of the objective function at a finite number of points per iteration. This method is planed for prevalent unconstrained minimization problems, such as nonlinear least squares and nonlinear simultaneous equations; see [8,13].

The problem discussed in this study is the following unconstrained optimization problem

$$\min f(x) \quad x \in \mathbb{R}^n,$$

where f is a nonlinear function from $\mathbb{R}^n$ into $\mathbb{R}$ and $x \in \mathbb{R}^n$. For the function $f(x)$, the Nelder–Mead method begins with $n+1$ vertices as $y_1, y_2, \ldots, y_{n+1}$ and then evaluates the objective function value of each vertices. We assign to y_1 as the best vertex where $f(x)$ is the lowest and to y_{n+1} as the worst point where $f(x)$ is the highest. The most common Nelder–Mead iterations perform a succession of primary geometric transformations including reflection, expansion, contraction, and shrinkage to find a better point and make it replace the worst point. Geometric transformations are controlled by four parameters including coefficients of reflection ρ, expansion χ, contraction γ, and shrinkage σ. The introduced parameters should assure the following consternations:

$$\rho > 0, \quad \chi > 1, \quad 0 < \gamma < 1, \quad \text{and} \quad 0 < \sigma < 1.$$

The common values, exerted in this paper, are

$$\rho = 1, \quad \chi = 2, \quad \gamma = \frac{1}{2}, \quad \text{and} \quad \sigma = \frac{1}{2}.$$

We set $\epsilon = 10^{-10}$ for the termination criterion in the Nelder–Mead method. The principal foundation of the Nelder–Mead method is described in Algorithm 2. Readers are referred to [8,19] for a detailed study of the Nelder–Mead method.

4 Curve approximation

We partition the outline of a 2D shape into curve segments based on its corner points. It means that all the outline points between two consecutive corner points constitute one curve segment. In this literature, cubic Bézier curves have been used to fit each segment separately using the Nelder–Mead algorithm.

4.1 Bézier curve

The Bézier curve is a parametric curve $C(t)$ that widely used to describe and model curves and surfaces in computer aided designs and computer graphics. The Bézier curve of degree n is given as

Algorithm 2 The Nelder–Mead algorithm [8]

1. Choose an initial $n+1$ vertex points $\{y_1, y_2, \ldots, y_{n+1}\}$ and choose a stopping criterion ϵ;
2. Order. Order and re-label the $n+1$ vertices to assure $f(y_1) \leq f(y_2) \leq \cdots \leq f(y_{n+1})$;
3. Reflect. calculate the reflection point y_r by $y_r = \bar{y} + \rho(\bar{y} - y_{n+1})$, where $\bar{y} = \sum_{i=1}^{n}(\frac{y_i}{n})$ is the centroid of the n best points;
if $f(y_1) \leq f(y_r) < f(y_n)$ **then**
replace y_{n+1} with the reflected point y_r and go to Step 7;
end if
4. **Expand.**
if $f(y_r) < f(y_1)$ **then**
calculate the expansion point y_e by $y_e = \bar{y} + \chi(y_r - \bar{y})$;
end if
if $f(y_e) < f(y_r)$ **then**
exchange y_{n+1} with y_e and go to Step 7;
else
exchange y_{n+1} with y_r and go to Step 7;
end if
5. Contract.
if $f(y_r) \geq f(y_n)$ **then**
a contraction is performed among $\bar{y}$ and the better of y_{n+1} and y_r;
end if
a. Outside.
if $f(y_n) \leq f(y_r) < f(y_{n+1})$ **then**
apply an outside contraction: compute $y_{oc} = \bar{y} + \gamma(y_r - \bar{y})$;
end if
if $f(y_{oc}) \leq f(y_r)$ **then**
exchange y_{n+1} with y_{oc} and go to Step 7;
else
go to Step 6 (perform a shrink);
end if
b. Inside.
if $f(y_r) \geq f(y_{n+1})$ **then**
apply an inside contraction: compute $y_{ic} = \bar{y} + \gamma(y_{n+1} - \bar{y})$;
end if
if $f(y_{ic}) \geq f(y_{n+1})$ **then**
exchange y_{n+1} with y_{ic} and go to Step 7;
else
go to Step 6 (perform a shrink);
end if
6. Shrink. Measure the n new vertices
$y' = y_1 + \sigma(y_i - y_1), i = 2, \ldots, n+1.$
exchange the vertices $y_2, \ldots, y_{n+1}$ with the new vertices $y'_2, \ldots, y'_{n+1}$;
7. Termination Criterion. Arrange and re-label the vertices of the new simplex as $y_1, y_2, \ldots, y_{n+1}$ such that $f(y_1) \leq f(y_2) \leq \cdots \leq f(y_{n+1})$;
if $f(y_{n+1}) - f(y_1) < \epsilon$ **then**
stop;
else
go to Step 3.
end if

$$C(t) = \sum_{i=0}^{n} P_i B_{i,n}(t), \quad 0 \le t \le 1,$$

where P_i refers to the control points and $B_{i,n}(t)$ is the Bernstein basis polynomials of index i and degree n expressed as

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \ldots, n.$$

From the defining equation of the Bézier curve, it can be seen that the properties of Bernstein polynomials are passed on to the Bézier curve; see [20]. These properties are as follows:

- Partition of unity.
- Affine invariance.
- Convex hull property.
- Endpoint interpolation.
- Value of Bernstein polynomials, for all $0 \le t \le 1$, does not appertain to the location of any control point(s); see [17].
- If the location of any control point is specified, its efficacy all over the curve can be deprived; see [17].

4.2 Curve fitting with the cubic Bézier curve

Let $\{X_i\}_{i=1}^{N} \subset \mathbb{R}^2$ denote an ordered set of contour points of a segment. Our goal is to find control points $P_i \quad (i = 0, 1, 2, 3)$ so that the cubic Bézier curve

$$C_3(t) = P_0 B_{0,3}(t) + P_1 B_{1,3}(t) + P_2 B_{2,3}(t) + P_3 B_{3,3}(t), \qquad 0 \le t \le 1,$$

can be a good representation of $\{X_i\}_{i=1}^{N}$. For this purpose, we minimize the linear least squares fitting error, E, specified as the sum of the squares of the deviations:

$$E = \sum_{k=1}^{N} ||C_3(t_k) - X_k||^2 = \sum_{k=1}^{N} \left\| \sum_{i=0}^{3} P_i B_{i,3}(t_k) - X_k \right\|^2,$$

where the parameter values t_i are assigned to X_i using the uniform parameterization method; see [20]. The control points P_0 and P_3 are the two corner

points of the segment. So, the effect of P_0 and P_3 is removed from a given cubic Bézier curve as

$$E = \sum_{k=1}^{N} ||C_3'(t_k) - X_k'||^2, \tag{1}$$

where

$$C_3'(t) = P_1 B_{1,3}(t) + P_2 B_{2,3}(t)$$

and

$$X_k' = X_k - (P_0 B_{0,3}(t_k) + P_3 B_{3,3}(t_k)), \quad k = 1, 2, ..., N.$$

Therefore, we have to find the control points P_1 and P_2 by minimizing the problem defined in (1).

Let $P_i = (P_{x_i}, P_{y_i}), i = 1, 2$, and write all the intermediate control points in one vector:

$$x = \begin{bmatrix} P_{x_1} \\ P_{y_1} \\ P_{x_2} \\ P_{y_2} \end{bmatrix}.$$

With this vector, the fitting problem is formulated as

$$\min_x \sum_{k=1}^{N} ||C_3'(t_k) - X_k'||^2. \tag{2}$$

That is, there are four parameters to estimate. We employ the Nelder–Mead method to solve the least square fitting problem defined in (2) and find these parameters.

4.3 Initialization

Before Algorithm 2 is implemented, a starting point should be selected for the intermediate control points. The initial vertex points can be chosen arbitrarily, but using the appropriate initial situation of the control points increases the speed of convergence and quality of solutions. Duo to $x \in \mathbb{R}^4$ in the objective function (2), five vertices have to be chosen. The details on the selection of the initial vertex points are explained as follows.

Let

$$P_1 B_{1,3}(t) + P_2 B_{2,3}(t) = \overline{C}_3(t), \tag{3}$$

where

$$\overline{C}_3(t) = C_3(t) - P_0 B_{0,3}(t) - P_3 B_{3,3}(t).$$

Based on the properties of the cubic Bézier curves, we get

$$B = B_{1,3}(0.5) = B_{2,3}(0.5). \tag{4}$$

From equations (3) and (4), we have

$$P_1 + P_2 = \widetilde{C}_3, \tag{5}$$

where

$$\widetilde{C}_3 = \frac{\overline{C}_3(0.5)}{B}.$$

By solving equations (3) and (5), we have

$$\begin{cases} P_1 = \dfrac{\overline{C}_3(t) - B_{2,3}(t)\widetilde{C}_3}{B_{1,3}(t) - B_{2,3}(t)}, \\ P_2 = \widetilde{C}_3 - P_1. \end{cases}$$

Suppose ${P_1}^i$ and ${P_2}^i$ are the computed intermediate control points at $(0 \leq t_i \leq 1, t_i \neq 0, 0.5, 1,)$, and

$$\begin{cases} {P_1}^i = \dfrac{\overline{C}_3(t_i) - B_{2,3}(t_i)\widetilde{C}_3}{B_{1,3}(t_i) - B_{2,3}(t_i)}, \\ {P_2}^i = \widetilde{C}_3 - {P_1}^i. \end{cases}$$

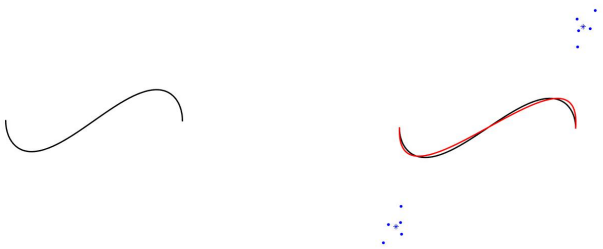
where $i = 1, 2, \ldots, 5$.

Let $P_1^i = (P_{x_1}^i, P_{y_1}^i)$ and $P_2^i = (P_{x_2}^i, P_{y_2}^i)$. The vector x_i can be identified as

$$x_i = \begin{bmatrix} P^i_{x_1} \\ P^i_{y_1} \\ P^i_{x_2} \\ P^i_{y_2} \end{bmatrix}, \quad i = 1, 2, \ldots, 5.$$

Hence, these vectors can be used for the initial vertex points of the Nelder–Mead method.

In order to illustrate the curve fitting using the Nelder–Mead algorithm, we consider a curve as shown in Figure 1(a). Figure 1(b) shows the fitting of a given curve (black) using a cubic Bézier curve (red). The initial vertex points are marked by $\bullet$, and the intermediate control points computed by the Nelder–Mead algorithm are marked by $*$. The fitting error versus the number of iterations for this curve with different initial vertex points is demonstrated in Figure 2. As shown in Figure 2, the Nelder–Mead method based on the proposed initial vertex points converges faster than the one based on random initial vertex points.



(a) Given curve. (b) Cubic Bézier approximation.

Figure 1: Fitting a given curve (black) using a cubic Bézier curve (red) with the proposed algorithm (NMSM).

4.4 Segment subdivision

When curve fitting is done by means of the proposed method (i.e. NMSM), there is a possibility for a generated complex curve not be acceptable and for the deviation error to be high. In such a case, the segment subdivision is applied to decrease the fitting deviation. In this paper, the complex segment of the outline is partitioned into two parts at the point of the maximum deviation error if the maximum deviation error oversteps the given error threshold limit (ETL). The process of subdivision will continue constantly and recursively until the fitting error reaches a value under the given error threshold. An example of subdivision into five curves is demonstrated in

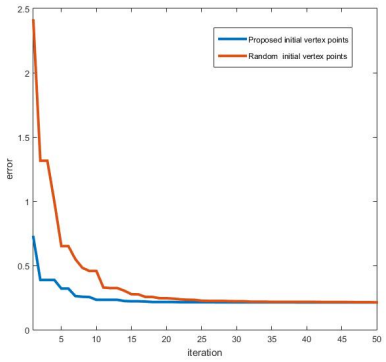
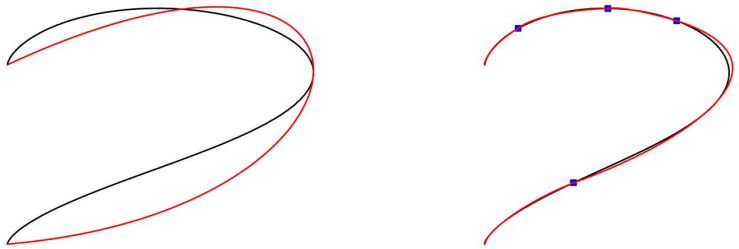


Figure 2: Error vs iteration.

Figure 3. Figure 3(a) is the curve fitting without segment subdivision and Figure 3(b) is curve fitting after segment subdivision. The black and red lines display the original and the approximated curves, respectively. The subdivided points are marked by ■. All the steps of the outline capturing system are shown in Algorithm 3.



(a) Without subdivision. (b) After subdivision.

Figure 3: Curve fitting with recursive subdivision.

Algorithm 3 The outline capturing system

```
Get a digitized image and choose an error threshold limit (ETL) ;
Extract an outline;
Detect the corner points and divide the outline into segments;
for each segment do
    Compute the initial vertex points;
    Perform curve fitting over the segments using the Nelder–Mead algorithm;
    Compute the maximum deviation error (MDE);
    if MDE ≤ ETL then
        break
    else
        Find the subdivision point and partition into two segments;
    end if
end for
```

5 Experimental results and comparative study

The proposed algorithm explained in the Section 4 has been used to some explanatory examples corresponding to generic shapes. Then, the outputs have been compared with those of algorithms [17, 29, 34]. In order to allow a fair comparison, we have tested the same shapes of the criterion as in other papers. In this study, the results are measured based on the following parameters:

- Number of segments: The outline of the shape is partitioned into a number of segments using a corner detection algorithm and subdivision points. An increase in the number of segments, leads to an increase in the number of control points. Therefore, the number of segments must be decreased.

- Compression ratio (CR): This is a significant criterion in the evaluation of the amount of compression accomplished by the method. A large compression ratio is favorable. This criterion indicates the ratio of the number of data points in an actual outline (n) to the obtained data points in curve fitting (nDP). It can be computed as:

$$CR = \frac{n}{nDP}.$$

- Average error: This parameter refers to all the errors generated in the fitted outline of the shape and is given as:

$$\text{Average error } = \frac{1}{n}\sum_{i=1}^{n} e_i,$$

where e_i is the Euclidean distance between the ith point of the outline and the corresponding point of the parametric curve.

- Maximum deviation: This parameter calculates the maximum deviation of the fitted outline from the original shape. It is calculated by the following equation:

$$\text{Maximum deviation } = \max_{i=1}^{n}\{e_i\}$$

- Computation time: This parameter indicates the amount of time taken by the algorithm for capturing the outline of the shape. It appertains to the performance approach and the processor used.

The algorithm has been performed and compared for Figures 4 and 5, which are Arabic words "Sabr", and "Kanji" characters, respectively. Figure 4(a) shows the image of the Arabic word "Sabr" and its boundary is given in Figure 4(b). Figure 4(c) shows the boundary along with the corner points marked with a square (■). Figure 4(d) shows the captured outline based on our approach, and the subdivision points are denoted with a circle (•). The outline computed by means of the method presented by Masood and Sarfraz [17] is shown in Figure 4(e). Figures 4(f) and 4(g), respectively, represent the outline evaluated by Sarfraz and Razzak [34] at a threshold of 3 and 2. The quantitative comparison of the different methods in Figure 4 can be observed in Table 1.

The results obtained for the "Kanji" character are presented in Figure 5 and tabulated in Table 2. The image of "Kanji" character and its boundary are shown in Figures 5(a) and 5(b), respectively. Figure 5(c) shows the boundary along with the corner points which are marked with a square (■). The results of the presented method for this shape are given in Figure 5(d). The outline captured with algorithm [17] is shown in Figure 5(e). The result obtained by Sarfraz and Khan [29] can be seen in Figure 5(f). The outline generated by means of method [34] is given in Figure 5(g). It should be noted that the results of Tables 1 and 2 are achieved by different systems.

Through a visual inspection of the results in Tables 1 and 2, one may come to the following conclusions:

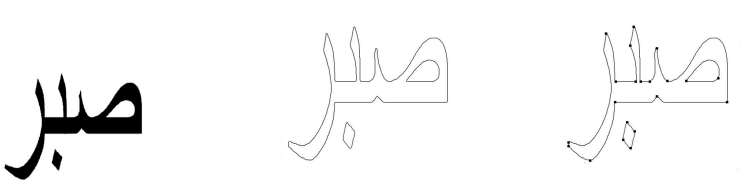
Table 1: Quantitative comparison for the Arabic word "Sabr".

Algorithm	Number of segments	Compression ratio	Maximum deviation	Average error	Computation time
Proposed (NMSM)	28	46.43	1.92	0.44	0.79
Masood and Sarfraz [17]	27	48.15	2.21	1.38	0.83
Sarfraz and Razzak [34] at $\tau = 3$	66	19.70	1.71	1.07	3.79
Sarfraz and Razzak [34] at $\tau = 2$	98	13.27	1.39	0.91	2.58

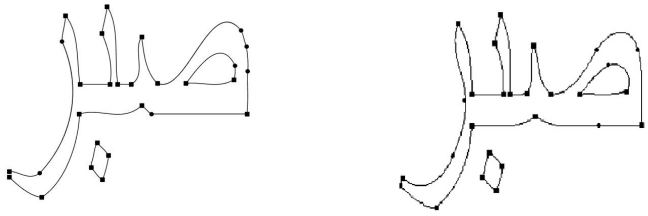
Table 2: Quantitative comparison for the "Kanji" character.

Algorithm	Number of segments	Compression ratio	Maximum deviation	Average error	Computation time
Proposed (NMSM)	33	64.81	1.53	0.40	0.55
Masood and Sarfraz [17]	33	64.81	1.55	0.73	0.59
Sarfraz and Khan [29]	31	69	3.78	1.92	3.61
Sarfraz and Razzak [34]	66	33.42	1.40	1.13	3.83

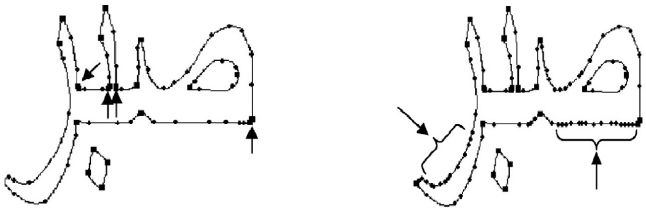
- The method proposed in this study protects an appropriate equilibrium among the compression ratio and the fitting error.
- The proposed method avoids time consuming procedures, such as least square fitting, and uses one of the popular derivative-free free operations (i.e., the NelderMead simplex method) that is simple and fast.
- The average error, maximum deviation, and the computation time are lesser than the results obtained by other methods in the current literature [17, 29, 34].
- The number of segments generated by method [34] is very high because the corner detection and the subdivision algorithms are suboptimal. The suboptimal corner points are indicated by arrows in Figures 4(f), 4(g), and 5(g). This method also suffers from a long computation time. This is due to the use of least squares fitting and control parameters.
- Computation of the outline by means of method [29] takes too long and has errors. The least squares curve fitting and the noise filtering procedure are the main causes for these disadvantages. It is shown with an arrow in Figure 5(f).
- The average error with algorithm [17] is higher than that in the proposed method, because this method does not use the standard optimization process.



(a) Original image. (b) Boundary of original image. (c) Boundary with corner points.

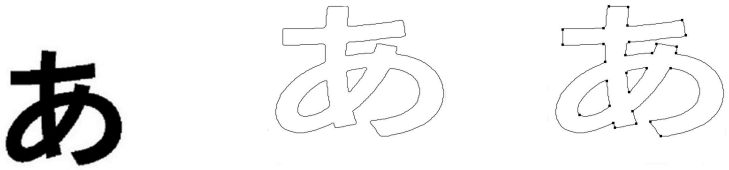


(d) Proposed method (NMSM). (e) Masood and Sarfraz [17].

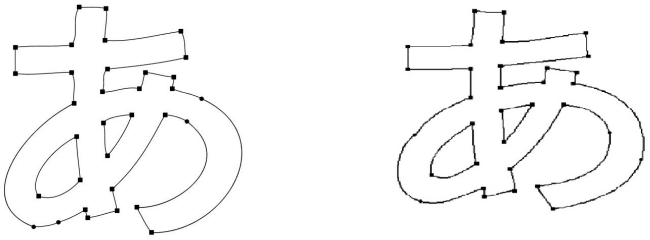


(f) Sarfraz and Razzak [34] at $\tau = 3$. (g) Sarfraz and Razzak [34] at $\tau = 2$.

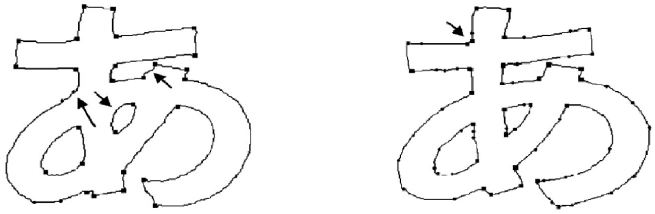
Figure 4: Captured shape of the Arabic word "Sabr" with different methods.



(a) Original image. (b) Outline of original image. (c) Boundary with corner points.



(d) Proposed method (NMSM). (e) Masood and Sarfraz [17].



(f) Sarfraz and Khan [29]. (g) Sarfraz and Razzak [34].

Figure 5: Captured shape of the Kanji character with different methods.

6 Conclusion

In this study, we introduced a novel outline capturing scheme for 2D shapes based on the Nelder–Mead simplex method. The Nelder–Mead simplex algorithm is straight to perform, easy to calculate, and does not call for much time to compute the optimization problem. The unique feature of this method is that it does not require to consider time consuming least squares fitting. The initial situation of the control points is obtained through the properties of

the cubic Bézier curve. It increases the speed of convergence and the quality of curve fitting. As a result, the introduced approximation scheme is much faster than traditional approaches and helps to capture a better fit with a desirable precision. Through a comparison of our method with previous approaches and considering the simulation results, it emerges that the method has such advantages as appropriate equivalency between the compression ratio and the fitting error, low deviation error, and low computation time.

References

1. Bézier, P. *Numerical control; mathematics and applications*, John Wiley & Sons, 1972.

2. Biswas, S. and Lovell, B.C. *Bézier and splines in image processing and machine vision*, Springer Science & Business Media, 2007.

3. Cabrelli, C.A. and Molter, U.M. *Automatic representation of binary images,* IEEE Trans. Pattern. Anal. Mach. Intell., 12 (1990), 1190–1196.

4. Chetverikov, D. and Szabo, Z. *A simple and efficient algorithm for detection of high curvature points in planar curves,* proc. of 23rd workshop of Australian Pattern Recognition Group, Steyr, (1999), 175–184.

5. Conn, A.R., Scheinberg, K. and Vicente, L.N. *Introduction to derivative-free optimization*, MOS-SIAM Series on Optimization, 2009.

6. Farin, G. *Curves and surfaces for computer-aided geometric design: a practical guide*, Academic Press, 1997.

7. Freeman, H. *On the encoding of arbitrary geometric configurations,* IEEE Trans. Comput., 2 (1961), 260–268.

8. Hassanien, A.E., Grosan, C. and Tolba, M.F. *Applications of intelligent optimization in biology and medicine: Current trends and open problems*, Springer, 2015.

9. Hussain, M., Hussain, M.Z., and Sarfraz, M. *Shape-preserving polynomial interpolation scheme,* Iran. J. Sci. Technol. Trans. A Sci., 40 (2016), no. 1, 9–18.

10. Hussain, M.Z., Hussain, F. and Sarfraz, M. *Shape-preserving positive trigonometric spline curves,* Iran. J. Sci. Technol. Trans. A Sci. 42 (2018), no. 2, 1–13.

11. Itoh, K. and Ohno, Y.*A curve fitting algorithm for character fonts,* Electronic publishing , 6 (1993), no. 3, 195–205.

12. Khan, M.S., Ayob, A. F.M., Isaacs, A. and Ray, T. *A novel evolutionary approach for 2d shape matching based on B-spline modeling,* Proc. Congr. Evol. Comput., (2011), 655–661.

13. Klein, K. and Neira, J. *Nelder–Mead simplex optimization routine for large-scale problems: A distributed memory implementation,* Comput. Econ., 4 (2014), 447–461.

14. Lagarias, J.C., Reeds, J.A., Wright, M.H. and Wright, P.E. *Convergence properties of the nelder–mead simplex method in low dimensions,* SIAM. J. Optim. , 9 (1998), no. 1, 112–147.

15. Marji, M. and Siy, P. *A new algorithm for dominant points detection and polygonization of digital curves,* Pattern. Recognit., 10 (2003), 2239–2251.

16. Masood, A. and Haq, S.A. *Object coding for real time image processing applications,* Pattern Recognition and Image Analysis. ICAPR 2005. Lecture Notes in Computer Science, vol 3687. Springer, Berlin, Heidelberg.

17. Masood, A. and Sarfraz, M. *An efficient technique for capturing 2d objects,* Comput. Graph., 32 (2008), no 1, 93–104.

18. Masood, A. and Sarfraz, M. *Capturing outlines of 2d objects with bézier cubic approximation,* Image Vis. Comput., 6 (2009), 704–712.

19. Nelder, J. A., and Mead, R. *A simplex method for function minimization,* Comput. J., 7 (1965), no. 4, 308–313.

20. Piegl, L. and Tiller, W. *The NURBS book*. Springer Science & Business Media, 2012.

21. Plass, M. and Stone, M. *Curve-fitting with piecewise parametric cubics*, Comput. Graph., 17 (1983), no 3, 229–239.

22. Powell, S. *Applications and enhancements of aircraft design optimization techniques*, PhD thesis, University of Southampton, 2012.

23. Salomon, D. *Curves and surfaces for computer graphics*, Springer Science & Business Media, 2007.

24. Sarfraz, M. *Some algorithms for curve design and automatic outline capturing of images,* Int. J. Image. Graph., 2 (2004), 301–324.

25. Sarfraz, M. *Computer-aided intelligent recognition techniques and applications*, Wiley Online Library, 2005.

26. Sarfraz, M. *Vectorizing outlines of generic shapes by cubic spline using simulated annealing,* Int. J. Comput. Math., 8 (2010), 1736–1751.

27. Sarfraz, M. *Capturing image outlines using simulated annealing approach with conic splines,* The Proceedings of the International Conference on Information and Intelligent Computing, (2011), 152–157.

28. Sarfraz, M. and Khan, M. *Automatic outline capture of arabic fonts,* Inf. Sci., 3 (2002), 269–281.

29. Sarfraz, M. and Khan, M. *An automatic algorithm for approximating boundary of bitmap characters,* Future. Gener. Comput. Syst., 8 (2004), 1327–1336.

30. Sarfraz, M. and Masood, A. *Capturing outlines of planar images using bézier cubics,* Comput. Graph., 5 (2007), 719–729.

31. Sarfraz, M., Masood, A. and Asim, M.R. *A new approach to corner detection,* Computer Vision and Graphics 32 (2006), 528533

32. Sarfraz, M. and Raza, A. *Visualization of data with spline fitting: a tool with a genetic approach,* CISST, 97(2002), 99–105.

33. Sarfraz, M. and Raza, S.A. *Capturing outline of fonts using genetic algorithm and splines,* Proceedings of IEEE International Conference on Information Visualization-IV'2001-UK, USA:IEEE Computer Society Press, (2001), 738–743.

34. Sarfraz, M. and Razzak, M. *An algorithm for automatic capturing of the font outlines,* Comput. Graph., 5 (2002), 795–804.

35. Sarfraz, M., Riyazuddin, M. and Baig, M. *Capturing planar shapes by approximating their outlines,* J. Comput. Appl. Math., 1 (2006), 494–512.

36. Sohel, F.A., Karmakar, G.C., Dooley, L.S. and Arkinstall, J. *Enhanced bezier curve models incorporating local information,* Proc. IEEE. Int. Conf. Acoust. Speech. Signal. Process., 4 (2005), 253–256.

37. Tang, Y. Y., Yang, F., and Liu, J. *Basic processes of chinese character based on cubic b-spline wavelet transform,* IEEE. Trans. Pattern. Anal. Mach. Intell., 12 (2001), 1443–1448.

38. Zheng, W., Bo, P., Liu, Y., and Wang, W. *Fast B-spline curve fitting by L-BFGS,* Comput. Aided. Geom. Des., 7 (2012), 448–462.

39. Zhu, F. *Geometric Parameterisation and Aerodynamic Shape Optimisation*, PhD thesis, University of Sheffield, 2014.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 123–139
DOI:10.22067/ijnao.v9i2.75991

Research Article

An extension of the quasi-Newton method for minimizing locally Lipschitz functions

Z. Akbari *

Abstract

We present a method to minimize locally Lipschitz functions. At first, a local quadratic model is developed to approximate a locally Lipschitz function. This model is constructed by using the ϵ-subdifferential. We minimize this local model and compute a search direction. It is shown that this direction is descent. We generalize the Wolfe conditions for finding an adequate step length along this direction. Next, the method is equipped with a quasi-Newton approach to update the local model and its globally convergence is proposed. Finally, the proposed algorithm is implemented in MATLAB environment on some standard nonsmooth optimization test problems and compared with some algorithms in the literature.

AMS(2010): 49J52; 90C56; 90C26.

Keywords: Quasi-Newton method; Quadratic model; Line search algorithm; Locally Lipschitz functions.

1 Introduction

In this paper, we consider the following unconstrained nonsmooth optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz continuous function. There exist several methods for solving this problem, for example, the subgradient method [25], the bundle methods [7,10,19,24], algorithms based on smoothing

*Corresponding author
Received 16 September 2018; revised 6 April 2019; accepted 10 April 2019
Z. Akbari
Faculty of Mathematical Sciences, University of Mazandaran, Babolsar, Iran. e-mail: z.akbari@umz.ac.ir

techniques [22], the derivative free methods [3], the bundle and quasi-Newton combining methods [13,15,17], the gradient sampling method [5,12], the limited memory gradient bundle method [11], and the trust region method [2,23].

In the most smooth optimization problems, the objective function is approximated by a quadratic model as follows:

$$Q(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{d^T B_k d}{2}, \tag{2}$$

where $\nabla f(x_k)$ and B_k are the gradient and Hessian of f at x_k; see [21]. In fact, the objective function is locally approximated at x_k. In quasi-Newton methods, the quadratic model (2) is minimized and its solution is a descent direction at x_k. Afterward a line search method is applied along this direction. This process is repeated until the optimal solution is achieved. Also, in trust region methods, the model (2) is minimized on a given region, called trust region, and a search direction is obtained. If this direction does not decrease the objective function adequately, then the trust region is updated.

In the nonsmooth optimization same as the smooth one, the objective function is approximated. Han et al. [9], presented a quadratic model to approximate the locally Lipschitz function f as follows:

$$Q(x_k + d) = f(x_k) + \phi(x_k, d) + \frac{d^T B_k d}{2}, \tag{3}$$

where $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a given iteration function and B_k is a given $n \times n$ symmetric matrix. The iteration function ϕ must have some properties to guarantee the convergence of produced minimization algorithm.

The presented iteration function in [9] is not practical. So, in this paper, we propose a practical iteration function and solve the following problem:

$$\min_{d \in \mathbb{R}^n} Q_k(x_k + d) = f(x_k) + \phi(x_k, d) + \frac{d^T B_k d}{2}. \tag{4}$$

Suppose that d_k is the solution of (4). We show that d_k is a descent direction and develop a line search method to reduce objective function along this direction.

In this paper, we propose a computable quadratic model to approximate the locally Lipschitz function and develop a minimization algorithm based on this model. In Section 2, some preliminary concepts of the nonsmooth analysis are reviewed. In Section 3, an iteration function is introduced for the locally Lipschitz objective function. After that, a descent algorithm is proposed. In Section 4, the global convergence of the presented algorithm is proved. The line search algorithm is described in Section 5. Numerical results are given in Section 6.

2 Preliminaries

In this section, we state the basic concepts and definitions of nonsmooth analysis from [4]. The Clarke generalized directional derivative of the locally Lipschitz function f at the point x in the direction d is defined by

$$f^{\circ}(x,d) := \limsup_{y \to x, t \downarrow 0} \frac{f(y+td) - f(x)}{t}.$$

The Clarke generalized gradient of f at x is the set $\partial f(x)$ defined by

$$\partial f(x) = \{\xi \in \mathbb{R}^n | f^{\circ}(x;d) \geq \xi^T d \text{ for all } d \in \mathbb{R}^n\}.$$

Each vector $\xi \in \partial f(x)$ is called a subgradient of f at x. For $\epsilon > 0$, The Clarke generalized ϵ-directional derivative of f at x in the direction d is defined by

$$f_{\epsilon}^{\circ}(x,d) := \limsup_{y \to x, t \downarrow 0} \frac{f(y+td) - f(x) + \epsilon}{t},$$

and the Goldstein ϵ-subdifferential of f at the point x is the set

$$\partial_{\epsilon} f(x) := \mathsf{cl}\ \mathsf{conv}\{\partial f(y),\ \|x-y\|_2 \leq \epsilon\},$$

where conv and cl are used to denote the convex hull and the closure, respectively. Each element $\xi \in \partial_{\epsilon} f(x)$ is called an ϵ-subgradient of the function f at x; see [4]. It can be seen that $f_{\epsilon}^{\circ}(x,d) = \sup_{\xi \in \partial_{\epsilon} f(x)} \xi^T d$. Let L be the Lipschitz constant of f in a neighborhood of x. Then, we have

$$\|v\|_2 \leq L, \quad \text{for all } v \in \partial f(x).$$

Throughout the paper, the used norm is the Euclidean norm. If f is differentiable at x, then $\nabla f(x) \in \partial f(x)$. Furthermore, if f is continuously differentiable at x, then

$$\partial f(x) = \{\nabla f(x)\}.$$

If $0 \in \partial_{\epsilon} f(x)$, then x is called as an ϵ-stationary point.

3 Generalized quasi-Newton method

In this section, we develop a computable iteration function and suppose that B_k is positive definite and has bounded norm, $m \leq \|B_k\| \leq M$. Since B_k^{-1} is positive definite, then we consider its Cholesky decomposition as follows:

$$B_k^{-1} = L^T L,$$

where L is an upper triangular matrix with real and positive diagonal entries, and L^T denotes the transpose of L. To define the iteration function, we consider the following problem:

$$v_k = \arg \min_{\xi \in \partial_\epsilon f(x_k)} \|L\xi\|^2, \tag{5}$$

where ϵ is a small enough positive scalar, and define $\phi(x_k, d) = v_k^T d$. Therefore the problem (4) is rewritten as follows:

$$\min_{d \in \mathbb{R}^n} q_k(d) = v_k^T d + \frac{d^T B_k d}{2}. \tag{6}$$

By this modification, $-B_k^{-1} v_k$ is the unique solution of (6). Let $d_k = -\frac{B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2}$ be a search direction. The line search is applied along this direction and the step length α_k is returned. The next iteration is computed as follows:

$$x_{k+1} = x_k + \alpha_k d_k.$$

At the point x_{k+1}, either B_k is updated by one of the quasi-Newton methods or is set the identity matrix for all k. Now, we are ready to express the generalized quasi-Newton algorithm for solving the problem (1):

Algorithm 1 Generalized quasi-Newton algorithm

Step 0 (Initialization): Let $B_1 = I_{n\times n}$, $\epsilon > 0$, $c_1 \in (0, 1)$, $x_1 \in \mathbb{R}^n$, $k = 1$.

Step 1 (Creating an iteration function): Set L as an upper triangular matrix of Cholesky decomposition for the matrix B_k^{-1}. Solve (5) at the point x_k and denote its solution by v_k.

Step 2 (Stopping condition): **If** $\|v_k\|_2 = 0$, then **stop**, **else** set $d_k = -\frac{B_k^{-1} v_k}{\|B_k^{-1} v_k\|}$ and compute $q_k = q_k(d_k)$.

Step 3 (Line search method): Apply the line search algorithm and find the step length $\alpha_k \in [\epsilon, 1]$ such that

$$\alpha_k = \max \left\{ \alpha \mid f(x_k + \alpha d_k) - f(x_k) \leq c_1 \alpha v_k^T d_k \right\}.$$

Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 4 (Updating): **If** it is necessary, **then** update B_k, set $k = k + 1$ and **go to step 1**.

As mentioned before, we can set $B_k = I$ for all k in Step 4, which means that the matrix B_k^{-1} is a fixed and identity matrix. Therefore, Algorithm 1 is converted to the steepest descent method for minimizing the locally Lipschitz function [18]. If B_k is updated by the BFGS formula (see [21]), then Algorithm 1 is a generalization of the quasi-Newton method for minimizing the locally Lipschitz continuous function. In Section 5, we describe a line search algorithm how to find a step length along the search direction such that the Wolfe conditions are satisfied.

The following lemma shows that the optimal value of (6) is nonpositive. Also it ensures that if the optimal value of (6) is nonzero, then the corresponding solution provides a descent direction for the function f at the point x_k. Else x_k is the optimal solution.

Lemma 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function and let $c_1 \in (0, 1)$. Suppose that v_k is the solution of problem (5) at x_k such that $v_k \neq 0$. If $d_k = -\frac{B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2}$, then*

(a) $v_k^T d_k \leq 0$ and $q_k \leq 0$.

(b) if $v_k^T d_k > 0$, then there exists a scalar $\bar{\alpha} > 0$ such that for all $\alpha \in [0, \bar{\alpha}]$, we have

$$f(x_k + \alpha d_k) - f(x_k) \leq c_1 \alpha v_k^T d_k.$$

Proof. Part (a) is clear. To prove part (b), at the contrary, we suppose there exists a positive sequence $\{\alpha_i\}$ such that $\alpha_i \to 0$ and we have

$$f(x_k + \alpha_i d_k) - f(x_k) > c_1 \alpha_i v_k^T d_k.$$

Both sides of the above inequality are divided by α_i, then, with passing to the $\limsup_{i\to\infty}$, we deduce

$$f_\epsilon^\circ(x_k, d_k) \geq f^\circ(x_k, d_k) \geq \limsup_{i\to\infty} \frac{f(x_k + \alpha_i d_k) - f(x_k)}{\alpha_i} \geq c_1 v_k^T d_k.$$

Since $d_k = -\frac{B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2}$, then $v_k^T d_k = -\frac{v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} = -\frac{\|L v_k\|_2^2}{\|B_k^{-1} v_k\|_2}$. Thus we have

$$f_\epsilon^\circ(x_k, d_k) \geq -c_1 \frac{\|L v_k\|^2}{\|B_k^{-1} v_k\|_2}. \tag{7}$$

On the other hand, we have

$$\begin{aligned} f_\epsilon^\circ(x_k, d_k) &= \max_{\xi \in \partial_\epsilon f(x_k)} \xi^T d_k = \max_{\xi \in \partial_\epsilon f(x_k)} \frac{-\xi^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2}, \\ &= \frac{-1}{\|B_k^{-1} v_k\|_2} \min_{\xi \in \partial_\epsilon f(x_k)} \xi^T L^T L v_k = \frac{-1}{\|B_k^{-1} v_k\|_2} \min_{\xi \in \partial_\epsilon f(x_k)} (L\xi)^T L v_k. \end{aligned}$$

According to (5), we have $\|Lv_k\|^2 \leq \|L\xi\|^2$, for all $\xi \in \partial_\epsilon f(x)$, and since the set $\partial_\epsilon f(x_k)$ is a convex compact set, referring to [20, Lemma 5.2.6.], reader can easily find out

$$f_\epsilon^\circ(x_k, d_k) = -\frac{\|Lv_k\|_2^2}{\|B_k^{-1}v_k\|_2}. \tag{8}$$

By (7) and (8), we have

$$-\frac{\|Lv_k\|_2^2}{\|B_k^{-1}v_k\|_2} = f_\epsilon^\circ(x_k, d_k \geq -c_1\frac{\|Lv_k\|_2^2}{\|B_k^{-1}v_k\|_2}.$$

Since $c_1 \in (0,1)$, the above equation is false. Thus, part (b) is proved. □

The following lemma shows that the step length $\alpha_k = \epsilon$ is satisfied the Arimijo condition along d_k for all k.

Lemma 2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function and let $c_1 \in (0,1)$. Then the step length ϵ satisfies the Armijo condition along d_k at x_k.*

Proof. Since f is a locally Lipschitz continuous function, then according to the mean-value theorem in [20], there exists $\eta \in (0,1)$ such that $z = x_k + \eta\epsilon d_k$ and

$$f(x_k + \epsilon d_k) - f(x_k) \in \partial f(z)^T(\epsilon d_k).$$

So, there exists $u \in \partial f(z)$ such that

$$f(x_k + \epsilon d_k) - f(x_k) = \epsilon u^T d_k = -\frac{\epsilon u^T B_k^{-1} v_k}{\|B_k^{-1}v_k\|_2} = -\frac{\epsilon (Lu)^T (Lv_k)}{\|B_k^{-1}v_k\|_2}. \tag{9}$$

Since $v_k \in \partial_\epsilon f(x_k)$ is the solution of problem (5), and $\partial_\epsilon f(x_k)$ is a convex compact set, then by [20, Lemma 5.2.6], we have

$$(Lu)^T(Lv_k) \geq \|Lv_k\|_2^2. \tag{10}$$

So, by (9) and (10), we have

$$f(x_k + \epsilon d_k) - f(x_k) \leq -\frac{\epsilon\|Lv_k\|_2^2}{\|B_k^{-1}v_k\|_2} = -\frac{\epsilon v_k^T B_k^{-1} v_k}{\|B_k^{-1}v_k\|_2} = \epsilon v_k^T d_k.$$

Since $v_k^T d_k < 0$ and $c_1 \in (0,1)$, then

$$f(x_k + \epsilon d_k) - f(x_k) \leq c_1 \epsilon v_k^T d_k.$$

Thus the proof is completed. □

4 Optimality condition and global convergence

In this section, we provide the necessary optimality condition and prove the global convergence of Algorithm 1. If $\|v_k\|_2 = 0$ at some iteration k, then $0 \in \partial_\epsilon f(x_k)$. This shows that x_k is an ϵ-stationary point. Now, we show that if Algorithm 1 dose not terminate after finite many iterations, then each accumulation point of the sequence $\{x_k\}$ is an ϵ-stationary point of f. To prove this, we first prove the following lemma.

Lemma 3. *Suppose that the level set $\mathcal{L} := \{x : f(x) \leq f(x_1)\}$ is bounded, and there exist positive constants M and m such that $m\|d\|_2^2 \leq d^T B_k d \leq M\|d\|_2^2$, for all k and $d \in \mathbb{R}^n$. If Algorithm 1 does not terminate after finite many iterations, then*

$$\lim_{k\to\infty} \|v_k\| = 0.$$

Proof. Suppose that Algorithm 1 does not terminate after finite many iterations. Since $\|v_k\| \neq 0$ for each k, then by Lemma 1, we have

$$f(x_k + \alpha_k d_k) - f(x_k) \leq c_1 \alpha_k v_k^T d_k. \tag{11}$$

Since $\{f(x_k)\}$ is a strictly decreasing and bounded below sequence, then it converges and we have

$$f(x_k + \alpha_k d_k) - f(x_k) \to 0.$$

So by (11), we have

$$0 \leq c_1 \lim_{k\to\infty} \alpha_k v_k^T d_k.$$

By Lemma 1, we know $v_k^T d_k = -\frac{v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} \leq 0$, so

$$0 \leq c_1 \lim_{k\to\infty} \alpha_k v_k^T d_k = -c_1 \lim_{k\to\infty} \frac{\alpha_k v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} \leq 0.$$

On the other hand, by Lemma 2, $\alpha_k \geq \epsilon$ Thus

$$\lim_{k\to\infty} \frac{v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} = 0. \tag{12}$$

Let $0 < \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$ be the eigenvalues of B_k^{-1}. Since B_k^{-1} is a positive definite matrix, then we have $\lambda_1^k \geq m$ and $\lambda_n^k \leq M$. Therefore

$$\frac{m}{M}\|v_k\|_2 \leq \frac{\lambda_1^k}{\lambda_n^k}\|v_k\|_2 \leq \frac{v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} \leq \frac{\lambda_n^k}{\lambda_1^k}\|v_k\| \leq \frac{M}{m}\|v_k\|_2. \tag{13}$$

Then, by (12) and (13), we conclude $\|v_k\|_2 \to 0$. □

In the following theorem, we show that each accumulation point of the sequence $\{x_k\}$ is an ϵ-stationary point.

Theorem 1. *Suppose that the level set $\mathcal{L} := \{x : f(x) \leq f(x_1)\}$ is bounded and that $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function on $\mathcal{L}$. If x^* is an accumulation point of the sequence $\{x_k\}$, then x^* is an ϵ-stationary point of f.*

Proof. Since $\{x_k\} \subset \mathcal{L}$, then the sequence $\{x_k\}$ is bounded and it has a convergent subsequence. Let $\{x_{k_i}\}$ be a subsequence of the sequence $\{x_k\}$ convergent to x^*. We have $v_{k_i} \in \partial_\epsilon f(x_{k_i})$ and by Lemma 3, $v_{k_i} \to 0$. On the other hand, $\partial_\epsilon f(\cdot)$ is upper semicontinuous. Thus $0 \in \partial_\epsilon f(x^*)$ and this shows that x^* is an ϵ-stationary point of f. □

5 Line search algorithm

In this section, we present a line search algorithm to update the matrix B_k by the BFGS method. To do this, we need two subgradients $\xi_k \in \partial f(x_k)$ and $\xi_{k+1} \in \partial f(x_{k+1})$, such that the curvature condition is satisfied, that is,

$$\xi_{k+1}^T d_k \geq c_2 \xi_k^T d_k,$$

where $c_2 \in (c_1, 1)$. To find ξ_{k+1}, we use the line search strategy same as [21, 26]. The step length α must satisfy the Wolfe conditions at x_k along d_k. The Wolfe conditions are the Armijo and curvature conditions, as

$$f(x_k + \alpha d_k) - f(x_k) \leq c_1 \alpha f_\epsilon^\circ(x_k, d_k), \text{ (Armijo condition)}$$
$$\exists \xi_{k+1} \in \partial f(x_k + \alpha d_k), \text{ s.t. } \xi_{k+1}^T d_k \geq c_2 f_\epsilon^\circ(x_k, d_k). \text{ (curvature condition)}$$

In this paper, we present an algorithm to find a step length, which satisfies the Armijo condition. Now we present the following algorithm. This algorithm finds an interval including a step length satisfies the Wolfe conditions at the point x_k along d_k. Algorithm 2 is started with $\alpha_0 = \epsilon$, because the step length ϵ satisfies the Armijo condition. The following proposition shows that Algorithm 2 terminates after finite many iterations.

Proposition 1. *If $f_\epsilon^\circ(x_k, d_k) < 0$ and f is bounded below at x_k along d_k, then Algorithm 2 terminates after finite many iterations.*

Proof. Since $\phi(\alpha) = f(x_k + \alpha d_k)$ is bounded below and the line $l(\alpha) = f(x_k) + c_1 \alpha f_\epsilon^\circ(x_k, d_k)$ is unbounded below, then there exists $\overline{\alpha}$ such that $\phi(\alpha) > l(\alpha)$ for all $\alpha > \overline{\alpha}$. Thus Algorithm 2 terminates when $\alpha_1 > \overline{\alpha}$. □

Either Algorithm 2 finds a step length that it satisfies the Wolfe conditions or it returns an interval. We prove that there exists a step length in this interval, which satisfies the Wolfe conditions.

Algorithm 2 Line search algorithm

set $\alpha_1 = 1, \alpha_0 = \epsilon$;
repeat
  if $f(x_k + \alpha_1 d_k) - f(x_k) > c_1 \alpha_1 f_\epsilon^\circ(x_k, d_k)$ **then**
    α^*=wolfe(α_0, α_1)
    return α^*
  else if $\exists \xi \in \partial f(x_k + \alpha_1 d_k)$ s.t. $\xi^T d_k \geq c_2 f_\epsilon^\circ(x_k, d_k)$ **then**
    $\alpha^* = \alpha_1$
    return α^*
  end if
  $\alpha_0 = \alpha_1$;
  $\alpha_1 = 2\alpha_1$
until

Proposition 2. *In Algorithm 2, suppose that* $f(x_k + \alpha_1 d_k) - f(x_k) > c_1 \alpha_1 f_\epsilon^\circ(x_k, d_k)$. *Then the interval* (α_0, α_1) *contains step lengths satisfying the Wolfe conditions.*

Proof. Let $\psi(\alpha) = f(x_k + \alpha d_k) - f(x_k) - c_2 \alpha f_\epsilon^\circ(x_k, d_k)$. It is obvious that $\psi(\alpha_0) \leq \psi(\alpha_1)$. Since the curvature condition is not satisfied at α_0, we have

$$\xi^T d_k < c_2 f^\circ f(x_k, d_k), \ \forall \xi \in \partial f(x_k + \alpha_0 d_k).$$

Thus $0 \notin \partial\phi(\alpha_0)$. Therefore α_0 is not the local minimizer of ψ on $[\alpha_0, \alpha_1]$. So the minimum point of ψ must be in (α_0, α_1). Suppose that $\alpha^* \in (\alpha_0, \alpha_1)$ is the minimizer of ψ. Thus $0 \in \partial\phi(\alpha^*)$. Since

$$\partial\phi(\alpha^*) \subset \partial f(x_k + \alpha^* d_k)^T d_k + c_2 f_\epsilon^\circ(x_k, d_k),$$

there exists $\xi \in \partial f(x_k + \alpha^* d_k)$ such that $\xi^T d_k - c_2 f_\epsilon^\circ(x_k, d_k) = 0$. On the other hand, $\psi(\alpha_0) \geq \psi(\alpha^*)$ and α_0 satisfies the Armijo condition. Thus α^* satisfies the Armijo condition. Therefore, α^* is satisfied in the Wolfe conditions. □

Now we present an algorithm to find a step length, which satisfies the Wolfe conditions on $[\alpha_0, \alpha_1]$. The following proposition indicates the convergence of Algorithm 3.

Proposition 3. *If Algorithm 3 does not terminate after finite many iterations, then it converges to* α^* *such that* $0 \in \partial f(x_k + \alpha^* d_k)^T d_k - c_2 f_\epsilon^\circ(x_k, d_k)$.

Proof. Let ψ be the function defined in Proposition 2. Since $\psi(a_i) < \psi(b_i)$ and $0 \notin \partial f(x_k + a_i d_k)^T d_k - c_2 f_\epsilon^\circ(x_k, d_k)$, then $0 \notin \partial\psi(a_i)$. Thus, ψ takes its minimum on (a_i, b_i). Suppose r_i is the minimum of ψ on (a_i, b_i). So, we have $0 \in \partial\psi(r_i)$. On the other hand, $\{a_i\}$ and $\{b_i\}$ are monotone and bounded sequences. Therefore, these sequences are convergent. Also, we have

Algorithm 3 Wolfe Algorithm

$i = 1$
$a_i = \alpha_1$
$b_i = \alpha_0$
$t_i = \frac{a_i+b_i}{2}$
repeat
  if $f(x_k + t_i d_k) - f(x_k) \geq c_1 t_i f_\epsilon^\circ(x_k, d_k)$ **then**
    $a_{i+1} = t_i$ and $b_{i+1} = b_i$
  else if $\exists \xi \in \partial f(x_k + t_i d_k)$ such that $\xi^T d_k \geq c_2 f_\epsilon^\circ(x_k, d_k)$ **then**
    return t_i
  else
    $b_{i+1} = t_i$ and $a_{i+1} = a_i$.
  end if
  $t_i = \frac{a_i+b_i}{2}$
  $i = i + 1$
until

$$\lim_{i\to\infty} a_i - b_i = 0,$$

thus

$$\lim_{i\to\infty} a_i = \lim_{i\to\infty} b_i = \lim_{i\to\infty} r_i = \alpha^*.$$

Since $\partial\psi(\cdot)$ is upper semicontinuous and $0 \in \partial\psi(r_i)$, then $0 \in \partial\psi(\alpha^*)$. This shows that $0 \in \partial f(x_k + \alpha^* d_k)^T d_k - c_2 f_\epsilon^\circ(x_k, d_k)$. □

Now, we go back to problem (5). Solving this problem is impractical, because the structural of the set $\partial_\epsilon f(x)$ is unclear. So the set $\partial_\epsilon f(x)$ is approximated and problem (5) is approximately solved. In [1], the generalized h-increasing point algorithm is proposed for computing an approximation solution of problem (5). Here, we summarize the generalized h-increasing point algorithm. Let W_l be a finite subset of $\partial_\epsilon f(x_k)$. The convex hull of W_l is considered as an approximation of $\partial_\epsilon f(x_k)$. We solve the following problem instead of (5):

$$\min_{v\in \mathsf{conv}W_k} v^T B_k v,$$

and consider w_k as its solution. Let $g_k = -\frac{w_k}{\|w_k\|}$. If the inequality

$$f(x_k + \epsilon g_k) - f(x_k) \leq -c_1 \epsilon \|w_k\|, \tag{14}$$

is satisfied, then $\mathsf{conv}W_k$ is an acceptable approximation of $\partial_\epsilon f(x_k)$. In this case, we consider g_k and $-\|w_k\|$ as an approximation of d_k and $f_\epsilon^\circ(x_k, d_k)$. Else a new element from $\partial_\epsilon f(x_k)$, such as v_l, must be added into W_k such that $v_l \notin \mathsf{conv}W_k$. To find such an element, we apply the generalized h-increasing point algorithm in [1]. In [1], it is proved that this procedure is terminated after finite many iterations. Either the direction g_k is returned

such that (14) is satisfied or we have

$$\|w_k\| \leq \delta, \tag{15}$$

where δ is a predefined threshold. If (14) is satisfied, then the Armijo condition is satisfied along direction g_k at x_k with step length ϵ. Also when (15) holds, we can assume that x_k is an ϵ-stationary point. Finding such a direction is given in Algorithm 3.1 in [1].

6 Numerical experiments

In the numerical experiments, as computing d_k by (5) is impractical, we use Algorithm 3.1 in [1] and approximate it with g_k. In this case, $f_\epsilon^\circ(x_k, d_k)$ is approximated with $-\|w_k\|$. We set parameters as follows: $\epsilon = 10^{-6}$, $\delta = 10^{-6}$, $c_1 = 10^{-4}$, and $c_2 = 0.9$. In Algorithms 2 and 3, the curvature condition is checked just by one subgradient from $\partial_\epsilon f(x_k + td_k)$, say ξ, that is, the following condition is checked:

$$\xi^T d_k \geq c_2 f_\epsilon^\circ(x_k, d_k).$$

In this section, we used some test problems in [8, 14] and report the computational and numerical results of Algorithm 1 denoted by "NQSN". Also Algorithm 1 is compared with some algorithms existing in the literatures. Some of the selected algorithms are implemented in MATLAB environment and other ones in Fortran. The measurement of algorithm efficiency is the number of function evaluations. To have a better comparison of the implemented algorithms, the performance profile of Dolan and More in [6] is used.

Two classes of test functions are applied. The first class is taken from [8] and the second one is the TEST29 in [14]. The test problems are introduced in Table 1. We compare the smooth BFGS method [21], the variable metric bundle method (PVAR) [16,17], a method presented in [18] (MY), the limited-memory BFGS method (LBFGS) [21], the limited memory bundle method (LMBM) [8], and the gradient sampling method (GS) with the presented method.

Each algorithm is terminated after 10000 iterations. In the performance profile, an algorithm solves a problem successfully when

$$\frac{|f_{\min} - f^*|}{1 + |f^*|} \leq thr,$$

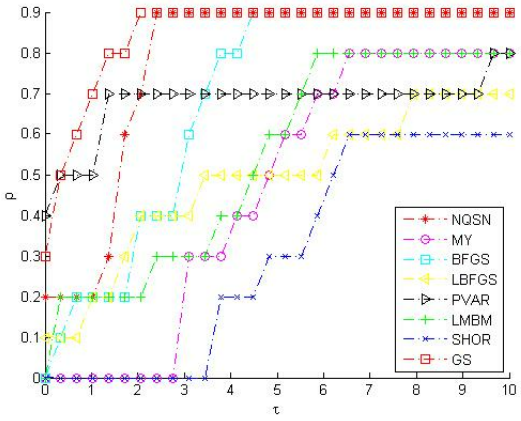
where $thr \in (0, 1)$ is a predefined threshold and $f_{\min}$ and f^* are the optimal value and achieved optimal value, respectively. In this paper, we set thr equal to 10^{-4}.

Table 1: Test problems and their optimal value for $n = 1000$.

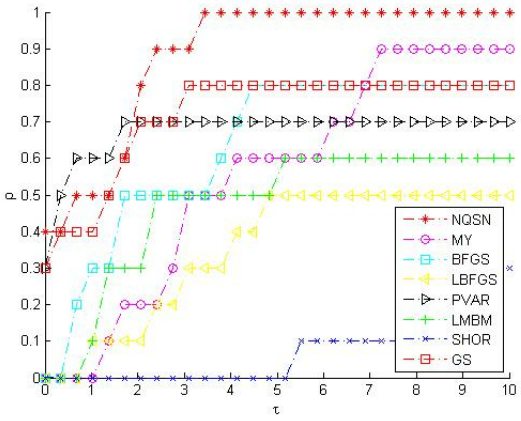
First class of problems			Second class of problems		
No.	problem	optimal value	No.	problem	optimal value
1	MAXQ	0	11	problem 2 from TEST29	0
2	MAXHILB	0	12	problem 5 from TEST29	0
3	LQ	-1.412799e+003	13	problem 6 from TEST29	0
4	CB3I	1998	14	problem 11 from TEST29	1.203128e+004
5	CB3II	1998	15	problem 13 from TEST29	5.661313e+002
6	NACTFACES	0	16	problem 17from TEST29	0
7	Brown 2	0	17	problem 19 from TEST29	0
8	Mifflin 2	-7.065034e+002	18	problem 20 from TEST29	0
9	Crescent I	0	19	problem 22 from TEST29	0
10	Crescent II	0	20	problem 24 from TEST29	0

Let n be the size of the test problem. The algorithms are tested with 3 sizes of test problems, the small size $n = 10$, middle size $n = 100$, and large scale $n = 1000$. Since the GS algorithm is very time consuming for large scale problems, then we do not run this algorithm for large scale problems. In Figures 1 and 2, we report the results for the first and second class of test problems as the performance profile, respectively.

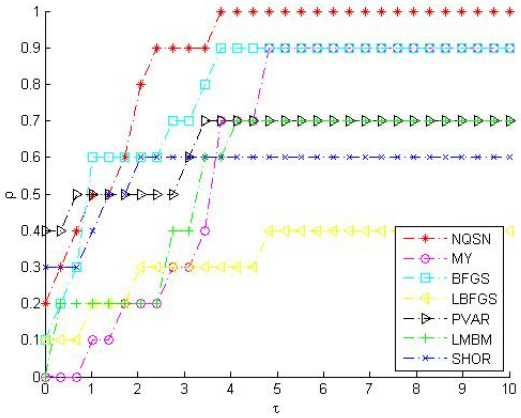
Figure 1 shows that the first class of the test problems is simpler than the second one. The proposed algorithm is more efficient than other algorithms for middle and large scale problems and is the only algorithm can solve all these problems. Also, the number of function evaluations of NQSN method is less than other methods. In the second class of test problems, for small size problems, the GS method is more efficient than other ones. But the BFGS and NQSN have similar results. But in the middle size, the MY method can just solve all problems. Performance profiles show that the NQSN method solves problems with significantly less number of function evaluations. For large scale test problems, just MY and NQSN can solve 6 problems. While the number of function evaluations in the NQSN method is very less than the number of function evaluations in MY method. The results show that the NQSN method is more efficient than other algorithms for large scale problems. Also, this algorithm solves problems with less number of function evaluations.



(a). Numerical results for $n = 10$.

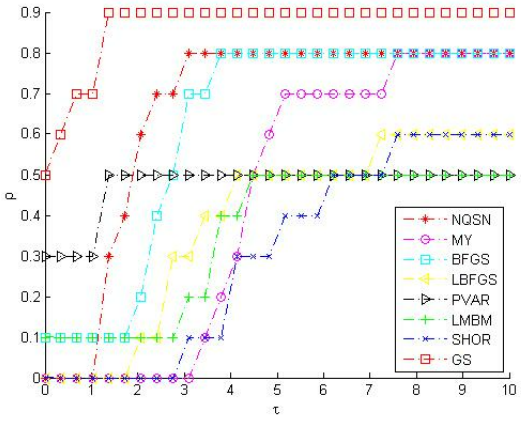


(b). Numerical results for $n = 100$.

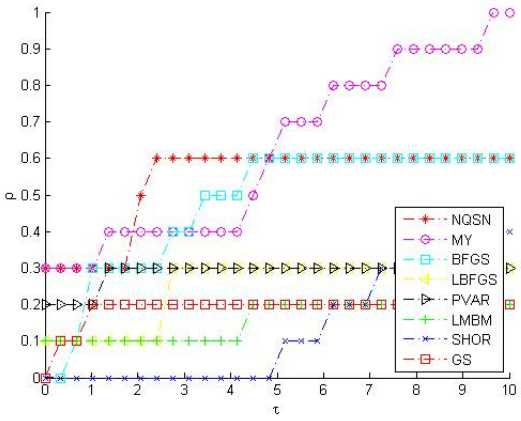


(c). Numerical results for $n = 1000$.

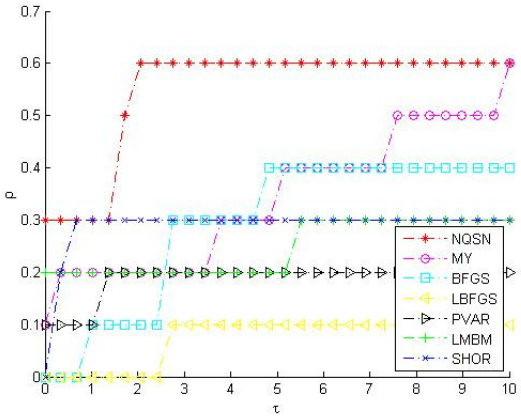
Figure 1: The performance profiles of the first class of test problems



(a). Numerical results for $n = 10$.



(b). Numerical results for $n = 100$.



(c). Numerical results for $n = 1000$.

Figure 2: The performance profiles of the second class of test problems

7 Conclusion

In this paper, the generalized quasi-Newton algorithm is presented for minimizing locally Lipschitz functions. Same as the iterative optimization methods, first we introduced a descent direction for the locally Lipschitz function in each iteration. Then, we computed an adequate step length along this direction satisfies the generalized Wolfe conditions. After that, we showed that the generalized quasi-Newton algorithm is convergent. The presented algorithm was applied to some standard test functions and compared with the MY method, the smooth BFGS method, the gradient sampling method, the limited-memory BFGS method and the limited memory bundle method. In future work, we will develop this algorithm for minimizing locally Lipschitz functions with nonlinear constraints.

Acknowledgements

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

References

1. Akbari, Z. and Yousefpour, R. *A trust region method for solving linearly constrained locally Lipschitz optimization problems*, Optimization, 66(9) (2017), 1519–1529.

2. Akbari, Z., Yousefpour, R. and Peyghami, M.R.*A new Nonsmooth trust region Algorithm for locally Lipschitz unconstrained optimization problems*, J. Optim. Theory Appl. 164(3) (2015), 733–754.

3. Bagirov, A.M. *Continuous subdifferential approximations and their applications*, Optimization and related topics, 2. J. Math. Sci. (N.Y.) 115 (2003), 2567–2609.

4. Bagirov, A.M., Karmitsa, N. and Mäkelä, M. M. *Introduction to nonsmooth optimization: Theory, practice and software*, Springer Publishing Company, Incorporated, 2014.

5. Burke, J.V. and Overton, M.L. *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM J. Optim. 15 (2005), 571–779.

6. Dolan, E.D. and More, J.J. *Benchmarking optimization software with performance profiles*, Math. Program. 91 (2002), 201–213.

7. Frangioni, A. *Generalized bundle methods*, SIAM J. Optim. 113 (2003), 117–156.

8. Haarala, N. and Miettinen, K. and Mäkelä, M.M. *Globally convergent limited memory bundle method for large-scale nonsmooth optimization*, Math. Program. 109 (2007), 181–205.

9. Han, S., Pang, J. and Rangaraj, N. *Globally Convergent Newton Methods for Nonsmooth Equations*, Math. Oper. Res.,17(3) (1992), 586–607.

10. Hiriart-Urruty, J-B. and Lemaréchal, C. *Convex analysis and minimization algorithms II: Advanced theory and bundle methods*, Springer-Verlag, 1993.

11. Karmitsa, N. and Bagirov, A.M. *Limited memory discrete gradient bundle method for nonsmooth derivative-free optimization*, Optimization, 61(12) (2012), 1491-1509.

12. Kiwiel, K.C. *Convergence of the Gradient Sampling Algorithm for Nonsmooth Nonconvex Optimization*, SIAM J. Optim. 18(2) (2007), 379–388.

13. Lewis, A.S. and Overton, M.L. *Nonsmooth optimization via quasi-Newton methods*, Math. Program. 141(1-2) (2012) Ser. A, 135–163. .

14. Luksan, L. , Tuma, M. , Siska, M., Vlcek, J. and Ramesova, N. *UFO 2002: interactive system for universal functional optimization*, Academy of Sciences of the Czech Republic, (2012). Also available as http://www.cs.cas.cz/luksan/ufo.pdf.

15. Luksan, L. and Vlcek, J. *A bundle-Newton method for nonsmooth unconstrained minimization*, Math. Program. 83 (1998), 373–391.

16. Luksan, L. and Vlcek, J. *Globally convergent variable metric method for convex nonsmooth unconstrained minimization*, J. Optim. Theory Appl. 102 (1999), 593–613.

17. Luksan, L. and Vlcek, J. *Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization*, J. Optim. Theory Appl., 111 (2001), 407–430.

18. Mahdavi-Amiri, M. and Yousefpour, R. *An effective nonsmooth optimization algorithm for locally Lipschitz functions*, J. Optim. Theory Appl. 155 (2012), 180–195.

19. Mäkelä, M. M. *Survey of bundle methods for nonsmooth optimization*, Optim. Methods Softw. 17 (1) (2002), 1–29.

20. Mäkelä, M.M. and Neittaanmaki, P. *Nonsmooth optimization: Theory and algorithms with applications to optimal control*, World Scientific, 1992.

21. Nocedal, J. and Wright, S.J. *Numerical Optimization*, Springer, 1999.

22. Polak, E. and Royset, J.O. *Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques*, J. Optim. Theory Appl. 119(421) (2003), 421–457.

23. Qi, L. and Sun, J. *A trust region algorithm for minimization of locally Lipschitzian functions*, Math. Program. 66 (1994), 25–43.

24. Schramm, H. and Zowe, J. *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optim. 2 (1992), 121–152.

25. Shor, N. Z. *Minimization methods for non-differentiable functions*, Springer-Verlag, 1985.

26. Yousefpour, R. *Combination of steepest descent and BFGS methods for nonconvex nonsmooth optimization*, Numer. Algorithms, 72(1) (2016), 57–90.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 141–163
DOI:10.22067/ijnao.v9i2.77371

Research Article

A new regularization term based on second order total generalized variation for image denoising problems

E. Tavakkol, S.M. Hosseini* and A.R. Hosseini

Abstract

Variational models are one of the most efficient techniques for image denoising problems. A variational method refers to the technique of optimizing a functional in order to restore appropriate solutions from observed data that best fit the original image. This paper proposes to revisit the discrete total generalized variation (TGV) image denoising problem by redefining the operations via the inclusion of a diagonal term to reduce the staircasing effect, which is the patchy artifacts usually observed in slanted regions of the image. We propose to add an oblique scheme in discretization operators, which we claim is aware of the alleviation of the staircasing effect superior to the conventional TGV method. Numerical experiments are carried out by using the primal-dual algorithm, and numerous real-world examples are conducted to confirm that the new proposed method achieves higher quality in terms of relative error and the peak signal to noise ratio compared with the conventional TGV method.

AMS(2010): 68Q25; 68R10; 68U05.

Keywords: Image denoising; Total variation; Staircasing effect; Total generalized variation; Peak signal to noise ratio.

*Corresponding author
Received 15 December 2018; revised 13 March 2019; accepted 29 April 2019
E. Tavakkol
Department of Applied Mathematics, Tarbiat Modares University, P. O. Box 14115-175, Tehran, Iran. e-mail: e.tavakkol@modares.ac.ir

S.M. Hosseini
Department of Applied Mathematics, Tarbiat Modares University, P. O. Box 14115-175, Tehran, Iran. e-mail: hossei_m@modares.ac.ir

A.R. Hosseini
School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, P. O. Box 14115-175, Tehran, Iran. e-mail: hosseini.alireza@ut.ac.ir

1 Introduction

Digital image processing (DIP) deals with performing operations on digital images. A digital image is a numerical representation of a physical scene, which is composed of a finite number of pixels. Digital images are produced by means of imaging machines that cover the electromagnetic spectrum. Synthetic images, electron microscopy images, and ultra-sound images are examples of digital images. Digital images were first used in the newspaper industry in the 1920s. These digital images were produced from a coded tape by a telegraph printer. The field of DIP is enriched with various applications, including image restoration [3,17,23,25], artistic effects [19], medical visualization [9,27,28,30], industrial inspection [22], law enforcement [32,33], and so on. Image restoration is one of the most widespread applications of DIP techniques that implements processes on digital images in order to estimate the original image from the corrupted one. The image distortion is caused due to different types of noise, such as Gaussian noise, white noise, salt and pepper noise, and speckle noise. In recent decades, variational approaches have been used as an efficient tool for image denoising problems.

A variational model is an optimization problem in which the criterion is defined as a functional (energy), which consists of a regularization term and a data fidelity term. Total variation (TV) regularization is a variational model that uses total variation as a regularization term. TV regularization was first proposed by Rudin, Osher, and Fatemi (ROF model) for imaging problems; see [26]. The ROF method is edge-preserving and has a fast numerical algorithm. Many different papers have shown the efficiency of TV minimization for image restoration [1,4,5,8,10,11,13–15,18,20,21,24,29,31]. The TV regularization has been widely used in various applications, such as image deblurring, inpainting, image zooming, segmentation problems, interpolation, spectral extrapolation, and stereovision. In these methods, the TV semi-nom is defined as

$$TV(u) = sup \left\{ \int_{\Omega} u \; div \; \nu \; dx \;\middle|\; \nu \in C_c^1(\Omega, \mathbf{R}^n), \; ||\nu||_\infty \leq 1 \right\},$$

where u is a function defined on a bounded region $\Omega \subset \mathbf{R}^n$. TV based regularization models have been proved to be efficient in image denoising problems. However, these models suffer from the staircasing effect, which appears as undesired patchy artifacts in slanted regions (see Figure 1). The total generalized variation (TGV) regularization model [6] is one technique to overcome this shortcoming, which acts as a regularization functional that incorporates higher-order derivatives and regularizes independently on various regularity levels. The main idea of TGV is a generalization of TV, which is defined as

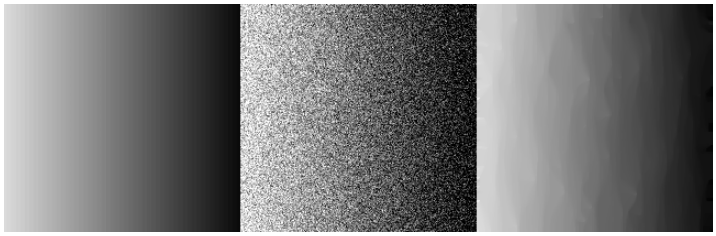


Figure 1: The implementation results of 1500 iterations of the ROF TV model [26]. From left to right: The reference image, the noisy image, and the restored image using the ROF TV. The original image has been corrupted by additive white Gaussian noise of standard deviation 0.18 and the regularization parameter is set to $\lambda = 0.17$. The ROF denoising model leads to staircasing effect, which is observed as patchy artifacts.

$$TGV_{\lambda}^{k}(u) = sup \left\{ \int_{\Omega} u \; div^{k}\nu \; dx \;\middle|\; \nu \in C_{c}^{k}(\Omega, Sym^{k}(\mathbf{R}^{n})),\; ||div^{l}\nu||_{\infty} \leq \lambda_{l}, \right.$$
$$\left. l = 0, \ldots, k-1 \right\},$$

where $k \geq 1$, $n \geq 1$, and λ_l are fixed positive parameters, and $Sym^k(\mathbf{R}^n)$ denotes the space of symmetric tensors of order k with arguments in $\mathbf{R}^n$. This paper proposes to revisit the TGV image denoising model by redefining the gradient operations via the inclusion of diagonal terms to reduce the staircasing effect. We propose to add an oblique scheme in classical image derivatives discretization, which we claim is aware of the alleviation of the staircasing effect superior to the conventional TGV method. Numerical experiments are carried out using the primal-dual algorithm, and numerous real-world experiments are conducted to confirm the effectiveness of the new approach.

The remainder of this paper is organized as follows. First, a brief explanation of some essential concepts regarding the TV and TGV schemes is presented in section 2. In section 3, the new proposed regularization model for image denoising problems and the corresponding theoretical results are presented. Section 4 is devoted to numerical experiments and comparisons that demonstrate the efficiency of the proposed method. Finally, Section 5 contains some concluding remarks.

2 Background

In this section, we present a brief review on essential concepts of TV and TGV approaches.

2.1 TV concept

Assume that u is a function defined on a bounded region $\Omega \subset \mathbf{R}^n$. The function u is said of bounded variation (BV function) if it is integrable and there exists a Radon measure Du such that

$$\int_\Omega u \; div \; \nu \; dx = - \int_\Omega \nu \; Du \; dx,$$

for all $\nu \in C_c^1(\mathbf{R}^n, \mathbf{R}^n)$, where $C_c^1(\mathbf{R}^n, \mathbf{R}^n)$ is the space of continuously differentiable vector functions ν of compact support and Du is the distributional (weak) derivative of u. The TV seminorm of u is defined as

$$TV(u) := sup\Big\{ \int_\Omega u \; div \; \nu \; dx \;\Big|\; \nu \in C_c^1(\Omega, \mathbf{R}^n), \; ||\nu||_\infty \leq 1 \Big\} = \int_\Omega |Du| dx,$$

where $|.|$ is the Euclidean norm. In the case that u is a smooth function, we have $Du = \nabla u$; therefore $TV(u)$ is the integral of its gradient magnitude

$$TV(u) = \int_\Omega |\nabla u| dx.$$

2.2 TGV concept

In this subsection, we review the essential concepts of TGV regularization from [6].

Definition 1. Let $k \geq 1$, $\Omega \subset \mathbf{R}^n$ be a bounded region, and $\lambda_0, \ldots, \lambda_{k-1}$ be fixed positive parameters. For $u \in L_{loc}^1(\Omega)$, the total generalized variation of order k with weight $\lambda \in \mathbf{R}^k$ is defined by the functional

$$TGV_\lambda^k(u) = sup \; \Big\{ \int_\Omega u \; div^k \nu \; dx \;\Big|\; \nu \in C_c^k(\Omega, Sym^k(\mathbf{R}^n)), \; ||div^l \nu||_\infty \leq \lambda_l,$$

$$l = 0, \ldots, k-1 \; \Big\},$$

where $\lambda = (\lambda_0, \ldots, \lambda_{k-1})$, $C_c^k(\Omega, Sym^k(\mathbf{R}^n))$ is the space of k-times continuously differentiable functions with compact support from Ω to $Sym^k(\mathbf{R}^n)$ and $Sym^k(\mathbf{R}^n)$ is the vector space of symmetric k-tensors defined as

$$Sym^k(\mathbf{R}^n) = \Big\{ \eta : \underbrace{\mathbf{R}^n \times \cdots \times \mathbf{R}^n}_{\text{k times}} \to \mathbf{R} \;\Big|\; \eta \; is \; k\text{-}linear \; and \; symmetric \Big\}.$$

For the case when $k = 2$, the TGV_λ^2 functional is a special case of TGV_λ^k functional, which is defined as

$$TGV_\lambda^2(u) = sup \; \Big\{ \int_\Omega u \; div^2 \nu \; dx \;\Big|\; \nu \in C_c^2(\Omega, Sym^2(\mathbf{R}^n)), \; ||\nu||_\infty \leq \lambda_0,$$

$$||div\ \nu||_\infty \leq \lambda_1\Big\}, \tag{1}$$

where $\lambda = (\lambda_0, \lambda_1)$ and

$$(div\ \nu)_i = \sum_{j=1}^{d} \frac{\partial \nu_{ij}}{\partial x_i},\ div^2\nu = \sum_{i=1}^{d} \frac{\partial^2 \nu_{ii}}{\partial x_i^2} + 2\sum_{i<j} \frac{\partial \nu_{ij}}{\partial x_i \partial x_j}.$$

Remark 1. The Euclidean space $\mathbf{R}^{N_1\times N_2}$ is equipped with the inner product

$$\langle u, s\rangle_{\mathbf{R}^{N_1\times N_2}} = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} u_{i,j}s_{i,j}, \quad u, s \in \mathbf{R}^{N_1\times N_2}.$$

For $u \in \mathbf{R}^{N_1\times N_2}$, the second order TGV semi-norm (1) can be discretized as

$$TGV_\lambda^2(u) = \max\ \Big\{\langle u, div^2\nu\rangle_{\mathbf{R}^{N_1\times N_2}} \ \Big|\ \nu \in (\mathbf{R}^4)^{N_1\times N_2},\ \nu = \begin{pmatrix} \nu_{11} & \nu_{12} \\ \nu_{12} & \nu_{22} \end{pmatrix},$$

$$\nu_{ij} \in \mathbf{R}^{N_1\times N_2}\ (i,j=1,2),\ ||\nu||_\infty \leq \lambda_0, ||div\ \nu||_\infty \leq \lambda_1\Big\}, \tag{2}$$

where $\lambda = (\lambda_0, \lambda_1)$. The discrete version of the infinity norm for the vector field z, where $z = div\ \nu$, is defined as

$$||z||_\infty = sup\left\{ \left(\sum_{i=1}^{n} z_i(x)^2 \right)^{1/2} \ \Big|\ x \in \Omega \right\}.$$

Moreover, the discrete version of the infinity norm for matrix ν is defined as

$$||\nu||_\infty = sup\left\{ \left(\sum_{i=1}^{n} \nu_{ii}(x)^2 + 2\sum_{i<j} \nu_{ij}(x)^2 \right)^{1/2} \ \Big|\ x \in \Omega \right\}.$$

3 The new proposed method

In this section, we present a new variant of the TGV model for the alleviation of the staircasing effect in image denoising problems by redefining the operators via the inclusion of a diagonal term. In the conventional TGV model presented in [6], the discretization operators are based on finite differences in the direction of the horizontal and vertical axes. Here, we reconstruct the discretization operators in [6] via the inclusion of a diagonal term. For this purpose, we introduce the versions of the discretization operators in parts (**a**-**e**). The indexes x, y, and o indicate that the corresponding finite-differences

are established in the direction of the horizontal, vertical, and diagonal axis, respectively.

a. For $u \in \mathbf{R}^{N_1 \times N_2}$, we define the discretization operator $\delta : \mathbf{R}^{N_1 \times N_2} \longrightarrow (\mathbf{R}^3)^{N_1 \times N_2}$ as

$$\delta(u) = \begin{bmatrix} \partial_x^+(u) \\ \partial_y^+(u) \\ \partial_o^+(u) \end{bmatrix},$$

$(i = 1, \ldots, N_1,\ j = 1, \ldots, N_2)$, where

$$(\partial_x^+ u)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } \ 1 \le i < N_1, \\ 0 & \text{if } \ i = N_1, \end{cases}$$

$$(\partial_y^+ u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } \ 1 \le j < N_2, \\ 0 & \text{if } \ j = N_2, \end{cases}$$

$$(\partial_o^+ u)_{i,j} = \begin{cases} u_{i+1,j+1} - u_{i,j} & \text{if } \ 1 \le i < N_1, 1 \le j < N_2, \\ 0 & \text{if } \ i = N_1, j = N_2. \end{cases}$$

b. For $p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$, $p_i \in \mathbf{R}^{N_1 \times N_2}$ $(i = 1, 2, 3)$, the discretization operator $\zeta : (\mathbf{R}^3)^{N_1 \times N_2} \longrightarrow \mathbf{R}^{N_1 \times N_2}$ is defined as

$$\zeta(p) = \partial_x^-(p_1) + \partial_y^-(p_2) + \partial_o^-(p_3),$$

where

$$(\partial_x^- p_1)_{i,j} = \begin{cases} (p_1)_{i,j} - (p_1)_{i-1,j} & \text{if } \ 1 < i < N_1, \\ (p_1)_{i,j} & \text{if } \ i = 1, \\ -(p_1)_{i-1,j} & \text{if } \ i = N_1, \end{cases}$$

$$(\partial_y^- p_2)_{i,j} = \begin{cases} (p_2)_{i,j} - (p_2)_{i,j-1} & \text{if } \ 1 < j < N_2, \\ (p_2)_{i,j} & \text{if } \ j = 1, \\ -(p_2)_{i,j-1} & \text{if } \ j = N_2, \end{cases}$$

$$(\partial_o^- p_3)_{i,j} = \begin{cases} (p_3)_{i,j} - (p_3)_{i-1,j-1} & \text{if } \ 1 < i \le N_1, 1 < j \le N_2, \\ (p_3)_{i,j} & \text{if } \ i = 1, j = 1. \end{cases}$$

c. For $p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$, $p_i \in \mathbf{R}^{N_1 \times N_2}$ $(i = 1, 2, 3)$, the discretization operator $\xi : (\mathbf{R}^3)^{N_1 \times N_2} \longrightarrow (\mathbf{R}^9)^{N_1 \times N_2}$ is defined as

$$\xi(p) = \begin{bmatrix} \partial_x^-(p_1) & \frac{1}{2}(\partial_x^-(p_2) + \partial_y^-(p_1)) & \frac{1}{2}(\partial_x^-(p_3) + \partial_o^-(p_1)) \\ \frac{1}{2}(\partial_x^-(p_2) + \partial_y^-(p_1)) & \partial_y^-(p_2) & \frac{1}{2}(\partial_y^-(p_3) + \partial_o^-(p_2)) \\ \frac{1}{2}(\partial_x^-(p_3) + \partial_o^-(p_1)) & \frac{1}{2}(\partial_y^-(p_3) + \partial_o^-(p_2)) & \partial_o^-(p_3) \end{bmatrix},$$

where ∂_x^-, ∂_y^-, and ∂_o^- are defined as in part (**b**).

d. For $\nu = \begin{bmatrix} \nu_{11}\ \nu_{12}\ \nu_{13} \\ \nu_{12}\ \nu_{22}\ \nu_{23} \\ \nu_{13}\ \nu_{23}\ \nu_{33} \end{bmatrix}$, $\nu_{ij} \in \mathbf{R}^{N_1 \times N_2}$ $(i,j = 1,2,3)$, the discretization operator $\zeta : (\mathbf{R}^9)^{N_1 \times N_2} \longrightarrow (\mathbf{R}^3)^{N_1 \times N_2}$ is defined as

$$\zeta(\nu) = \begin{bmatrix} \partial_x^+(\nu_{11}) + \partial_y^+(\nu_{12}) + \partial_o^+(\nu_{13}) \\ \partial_x^+(\nu_{12}) + \partial_y^+(\nu_{22}) + \partial_o^+(\nu_{23}) \\ \partial_x^+(\nu_{13}) + \partial_y^+(\nu_{23}) + \partial_o^+(\nu_{33}) \end{bmatrix},$$

where ∂_x^+, ∂_y^+, and ∂_o^+ are defined as in part (**a**).

e. For $\nu = \begin{bmatrix} \nu_{11}\ \nu_{12}\ \nu_{13} \\ \nu_{12}\ \nu_{22}\ \nu_{23} \\ \nu_{13}\ \nu_{23}\ \nu_{33} \end{bmatrix}$, $\nu_{ij} \in \mathbf{R}^{N_1 \times N_2}$ $(i,j = 1,2,3)$, the discretization operator $\zeta^2 : (\mathbf{R}^9)^{N_1 \times N_2} \longrightarrow \mathbf{R}^{N_1 \times N_2}$ is defined as

$$\zeta^2(\nu) = \partial_x^-\partial_x^+(\nu_{11}) + \partial_y^-\partial_y^+(\nu_{22}) + \partial_o^-\partial_o^+(\nu_{33}) + \partial_x^-\partial_y^+(\nu_{12}) + \partial_x^-\partial_o^+(\nu_{13})$$
$$+\partial_y^-\partial_x^+(\nu_{12}) + \partial_y^-\partial_o^+(\nu_{23}) + \partial_o^-\partial_x^+(\nu_{13}) + \partial_o^-\partial_y^+(\nu_{23}).$$

where ∂_x^+, ∂_y^+, ∂_o^+, ∂_x^-, ∂_y^-, and ∂_o^- are defined as in parts (**a**) and (**b**).

The next step, we aim to solve the following variational image denoising problem

$$\min_u\ \tfrac{1}{2}||u - u_0||_2^2 + L_\lambda(u), \tag{3}$$

where $u_0 \in \mathbf{R}^{N_1 \times N_2}$ is the noisy image, $u \in \mathbf{R}^{N_1 \times N_2}$ is the image to be reconstructed, $\lambda = (\lambda_0, \lambda_1)$ (λ_0, λ_1 are the regularization parameters), and $L_\lambda(u)$ is the new proposed regularization functional, which is defined as

$$L_\lambda(u) = \max\ \Big\{ \langle u\ ,\ \zeta^2(\nu) \rangle_{\mathbf{R}^{N_1 \times N_2}} \ \Big|\ \nu \in (\mathbf{R}^9)^{N_1 \times N_2},\ \nu = \begin{bmatrix} \nu_{11}\ \nu_{12}\ \nu_{13} \\ \nu_{12}\ \nu_{22}\ \nu_{23} \\ \nu_{13}\ \nu_{23}\ \nu_{33} \end{bmatrix},$$
$$\nu_{i,j} \in \mathbf{R}^{N_1 \times N_2}\ (i,j = 1,2,3),\ ||\nu||_\infty \le \lambda_0,\ ||\zeta(\nu)||_\infty \le \lambda_1 \Big\}. \tag{4}$$

3.1 Theoretical results

We apply the over-relaxed Chambolle–Pock algorithm described in [16] for solving (3). Since this algorithm solves jointly the primal and dual formulations of minimization problem (3), we need to obtain the Fenchel dual formulation of functional (4). The analytical process for establishing this formulation of (4) is stated in Theorem 2, in which we follow the steps of [7] for its proof. Before studying this theorem, the reader needs to be familiar with the concept of Legendre–Fenchel duality and Fenchel duality theorem from [2].

Definition 2. Given some convex, proper, and lower semi-continuous function $f(p)$ defined for $p \in H$, where H is a Hilbert space with inner product $\langle ., . \rangle_H$, its Legendre–Fenchel dual function is defined as

$$f^*(q) = \max \{ \langle p, q \rangle_H - f(p) \Big| p \in H \}$$

for all $q \in H$.

Theorem 1. *Assume that X and Y are real Banach spaces, that $f_1 : X \longrightarrow (-\infty, +\infty)$ and $f_2 : Y \longrightarrow (-\infty, +\infty)$ are proper, convex, and lower semi-continuous functions, and that $A : X \longrightarrow Y$ is a linear continuous operator. If there exists $x_0 \in X$ such that $f_1(x_0) < \infty$ and f_2 is continuous at Ax_0, then*

$$\min \{ f_1(x) + f_2(A(x)) \Big| x \in X \} = \max \{ -f_2^*(y^*) - f_1^*(-A^* y^*) \Big| y^* \in Y^* \}.$$

Remark 2. For $p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \in (\mathbf{R}^3)^{N_1 \times N_2}$ and $w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \in (\mathbf{R}^3)^{N_1 \times N_2}$, the Euclidean space $(\mathbf{R}^3)^{N_1 \times N_2}$ is equipped with the inner product

$$\langle p, w \rangle_{(\mathbf{R}^3)^{N_1 \times N_2}} = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (p_1)_{i,j} (w_1)_{i,j} + (p_2)_{i,j} (w_2)_{i,j} + (p_3)_{i,j} (w_3)_{i,j}.$$

Theorem 2. *The discrete second order total generalized variation functional (4) is equivalent to the following Fenchel dual formulation*

$$\begin{aligned} L_\lambda(u) = \min \ & \lambda_0 ||\xi(p)||_1 + \lambda_1 ||\delta(u) - p||_1, \\ & p \in (\boldsymbol{R}^3)^{N_1 \times N_2}, \end{aligned}$$

where ξ and δ are defined as in parts $(\boldsymbol{c})$ and $(\boldsymbol{a})$, respectively.

Proof. We prove in the lines of [7]. Based on (4), we have

$$L_\lambda(u) = \max\left\{ \langle u, \zeta^2(\nu)\rangle_{\mathbf{R}^{N_1\times N_2}} \;\middle|\; \nu \in (\mathbf{R}^9)^{N_1\times N_2},\ ||\nu||_\infty \le \lambda_0,\ ||\zeta(\nu)||_\infty \le \lambda_1 \right\}$$
$$= \max\left\{ \langle u, \zeta^2(\nu)\rangle_{\mathbf{R}^{N_1\times N_2}} - I_{\{||.||_\infty \le \lambda_0\}}(\nu) - I_{\{||.||_\infty \le \lambda_1\}}(\zeta(\nu)) \;\middle|\; \nu \in (\mathbf{R}^9)^{N_1\times N_2} \right\}$$
$$= -\min\left\{ I_{\{||.||_\infty \le \lambda_0\}}(\nu) + I_{\{||.||_\infty \le \lambda_1\}}(\zeta(\nu)) - \langle u, \zeta^2(\nu)\rangle_{\mathbf{R}^{N_1\times N_2}} \;\middle|\; \nu \in (\mathbf{R}^9)^{N_1\times N_2} \right\},$$

where

$$I_{\{||.||_\infty \le \lambda_0\}}(\nu) = \begin{cases} 0 & \text{if } ||\nu||_\infty \le \lambda_0, \\ \infty & \text{if } ||\nu||_\infty > \lambda_0, \end{cases}$$

$$I_{\{||.||_\infty \le \lambda_1\}}(\zeta(\nu)) = \begin{cases} 0 & \text{if } ||(\zeta(\nu))||_\infty \le \lambda_1, \\ \infty & \text{if } ||(\zeta(\nu))||_\infty > \lambda_1. \end{cases}$$

We choose

$$f_1(\nu) = I_{\{||.||_\infty \le \lambda_0\}}(\nu),\ f_2(\zeta(\nu)) = I_{\{||.||_\infty \le \lambda_1\}}(\zeta(\nu)) - \langle u, \zeta^2(\nu)\rangle_{\mathbf{R}^{N_1\times N_2}}.$$

Based on the principles of Theorem 1, it follows that

$$L_\lambda(u) = -\min\left\{ f_1(\nu) + f_2(\zeta(\nu)) \;\middle|\; \nu \in (\mathbf{R}^9)^{N_1\times N_2} \right\}$$
$$= \min\left\{ f_1^*(-\xi(p)) + f_2^*(p) \;\middle|\; p \in (\mathbf{R}^3)^{N_1\times N_2} \right\},$$

where $p = \zeta(\nu)$. Based on the duality principle of definition 2, it yields $f_1^*(-\xi(p)) = \lambda_0||\xi(p)||_1$, and

$$f_2^*(p) = \max\left\{ \langle p, w\rangle_{(\mathbf{R}^3)^{N_1\times N_2}} - f_2(w) \;\middle|\; w \in (\mathbf{R}^3)^{N_1\times N_2} \right\}$$
$$= \max\left\{ \langle p, w\rangle_{(\mathbf{R}^3)^{N_1\times N_2}} - I_{\{||.||_\infty \le \lambda_1\}}(w) + \langle u, \zeta(w)\rangle_{\mathbf{R}^{N_1\times N_2}} \;\middle|\; w \in (\mathbf{R}^3)^{N_1\times N_2} \right\}.$$

Since $\zeta^* = -\delta$, it yields

$$f_2^*(p) = \max\left\{ \langle \delta(u), w\rangle_{(\mathbf{R}^3)^{N_1\times N_2}} - \langle p, w\rangle_{(\mathbf{R}^3)^{N_1\times N_2}} \;\middle|\; ||w||_\infty \le \lambda_1,\ w \in (\mathbf{R}^3)^{N_1\times N_2} \right\}$$
$$= \max\left\{ \langle \delta(u) - p, w\rangle_{(\mathbf{R}^3)^{N_1\times N_2}} \;\middle|\; ||w||_\infty \le \lambda_1,\ w \in (\mathbf{R}^3)^{N_1\times N_2} \right\}.$$

Choosing $k = \delta(u) - p$, it follows that

$$f_2^*(p) = \max\left\{ \sum_{s=1}^{2}\sum_{j=1}^{N_2}\sum_{i=1}^{N_1} w_s(i,j)\ k_s(i,j) \;\middle|\; ||w||_\infty \le \lambda_1,\ w \in (\mathbf{R}^3)^{N_1\times N_2} \right\}$$
$$\lambda_1\ ||\delta(u) - p||_1.$$

□

3.2 Primal-dual algorithm

This section contains the primal-dual algorithm described in [16] for solving (3). Primal-dual methods apply proximity operators, which can be defined for proper, lower semi-continuous, convex, and extended real-valued functions.

The followings are the proximity operators, which are used in this algorithm,

$$prox_{\tau F_1}(u) = \frac{u + \tau u_0}{1 + \tau},\ prox_{\tau F_2}(p) = p - \frac{p}{\max(\frac{|p|}{\tau\lambda_1}, 1)},$$

$$prox_{\sigma g}(\nu) = \frac{\nu}{\max(\frac{|\nu|}{\lambda_0}, 1)}.$$

Algorithm 1 to solve (3)

1. Set $k = 0$, choose parameters τ, σ, ρ, and the initial estimates $u^{(0)} \in \mathbf{R}^{N_1 \times N_2}$, $p^{(0)} \in (\mathbf{R}^3)^{N_1 \times N_2}$, $\nu^{(0)} \in (\mathbf{R}^9)^{N_1 \times N_2}$.
2. Calculate $u^{(k+1)}$, $p^{(k+1)}$ and $\nu^{(k+1)}$ using the following equations:
$P_1 := prox_{\tau F_1}(u^{(k)} - \zeta(\tau\zeta(\nu^{(k)})))$,
$P_2 := prox_{\tau F_2}(p^{(k)} + \tau\zeta(\nu^{(k)}))$,
$P_3 := prox_{\sigma g}(\nu^{(k)} + \sigma\ \xi(\delta(2\ p_1 - u^{(k)}) - (2\ p_2 - p^{(k)})))$,
$u^{(k+1)} := u^{(k)} + \rho\ (p_1 - u^{(k)})$,
$p^{(k+1)} := p^{(k)} + \rho\ (p_2 - p^{(k)})$,
$\nu^{(k+1)} := \nu^{(k)} + \rho\ (p_3 - \nu^{(k)})$.
3. Stop or set $k := k + 1$ and go back to step 2.

4 Numerical experiments

In this section, we test the performance of the proposed method on several sample images to remove noise (see Figure 2). Each image is corrupted by additive white Gaussian noise of standard deviation 0.18. We compare the new proposed method with the anisotropic TV, the isotropic TV, the upwind TV in [12], and the TGV method in [6] (see Figures 3,4,5,6,7,8,9,10,11,12,13,14 for the restored images). For the selection of the optimal regularization parameter, the algorithms corresponding to the anisotropic TV, the isotropic TV, the upwind TV, the TGV method, and the proposed method are implemented with many different choices for λ, and the λ value corresponding to the best peak signal to noise ratio (PSNR) (least relative error) is chosen as the optimal λ value.

Using the anisotropic TV, it performs well in removing noise but small details become unclear, and this TV scheme suffers from the staircasing effect, which appears as undesired patchy artifacts in slanted regions. The upwind TV has nice performance in preserving details but it has a remarkable drawback because some small white particles of noise will be remained in restored images, which indicates that the upwind TV is not capable of removing white noise. Moreover, the upwind TV suffers from the staircasing effect. The isotropic TV performs to some extent better than the anisotropic

TV and the upwind TV in noise removal but still some details are not reconstructed during the denoising process, and this TV scheme is not capable of alleviating the staircasing effect. Using the TGV method, it still suffers from the remaining of the staircasing effect, which indicates that it is not powerful enough to handle the severe noise. The proposed method outperforms the anisotropic TV, the isotropic TV, the upwind TV, and the TGV method both in the reconstruction of fine structures and the elimination of the staircasing effect. The numerical experiments illustrate that the new proposed method achieves higher quality in terms of PSNR and relative error (see Table 1). For example in the case of bird image, Table 1 illustrates that the proposed method achieves a PSNR value, which is about 0.187585 higher than the PSNR value of TGV model. If we notice the other quantities in Table 1, we observe that the PSNR value corresponding to the TGV model is about 0.129112 higher than the PSNR value corresponding to isotropic TV, and this PSNR difference is less than the PSNR difference of the proposed method and the TGV method.

The iteration numbers of optimization algorithm for both bird and bike images in each model is 1500. Figures 15,16,17, and 18 illustrate the PSNR and the relative error of various methods versus iteration numbers. For example, in the case of bike image, Figures 17 and 18 illustrate that after 600 iterations, the proposed method has smaller PSNR and higher relative error in comparison with the TGV model (TGV: PSNR=23.038355, Relative error=0.092436, proposed: PSNR=22.980077, Relative error=0.093058). If we increase the number of iterations to 1500, the TGV model achieves better quantities (TGV: PSNR=23.060928, Relative error=0.092196) in comparison with 600 iterations, and the proposed model achieves the best PSNR value (proposed: $PSNR = 23.080060$, Relative error=0.091993) in comparison with other methods. Even if we increase the iteration numbers again, very few changes are observed, and the proposed method will still have the best PSNR value and the least relative error in comparison with other methods.

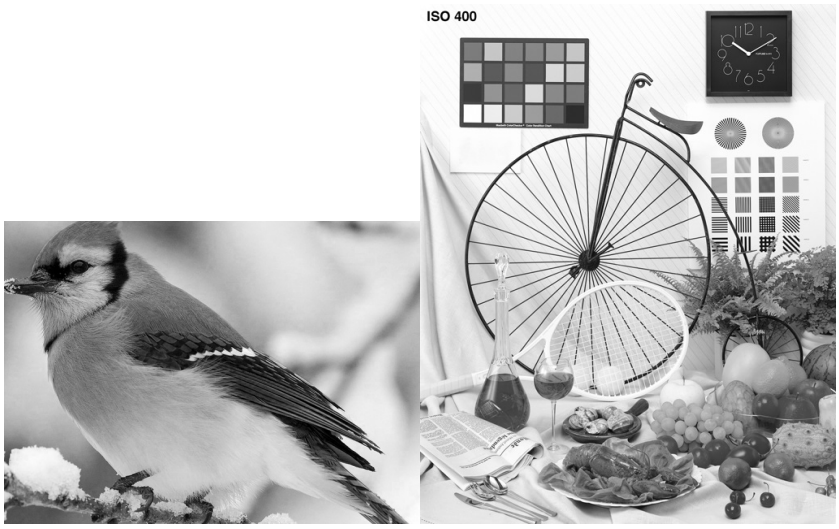


Figure 2: The test images used in our numerical experiments. Left: bird (490×674 in experiments); right: bike (640×512 in experiments).

Figure 3: From left to right: noisy bird image, restored bird image by anisotropic TV.

Figure 4: From left to right: restored bird images by isotropic TV and upwind TV.

Figure 5: From left to right: restored bird images by TGV and the new proposed method.

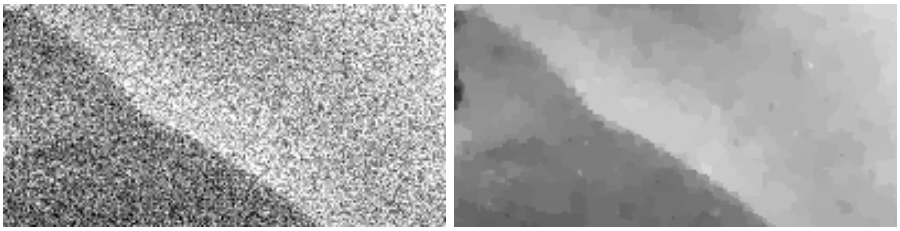


Figure 6: Zoomed-in regions of bird image. From left to right: noisy image, restored by anisotropic TV.

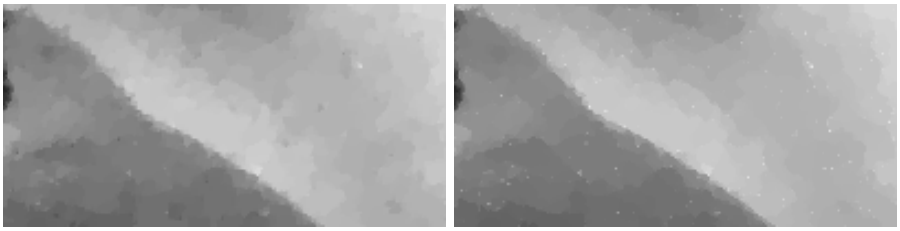


Figure 7: Zoomed-in regions of bird image. From left to right: restored by isotropic TV and upwind TV.

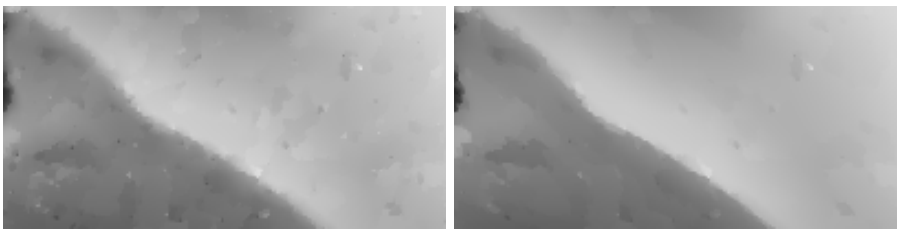


Figure 8: Zoomed-in regions of bird image. From left to right: restored by TGV and the new proposed method.

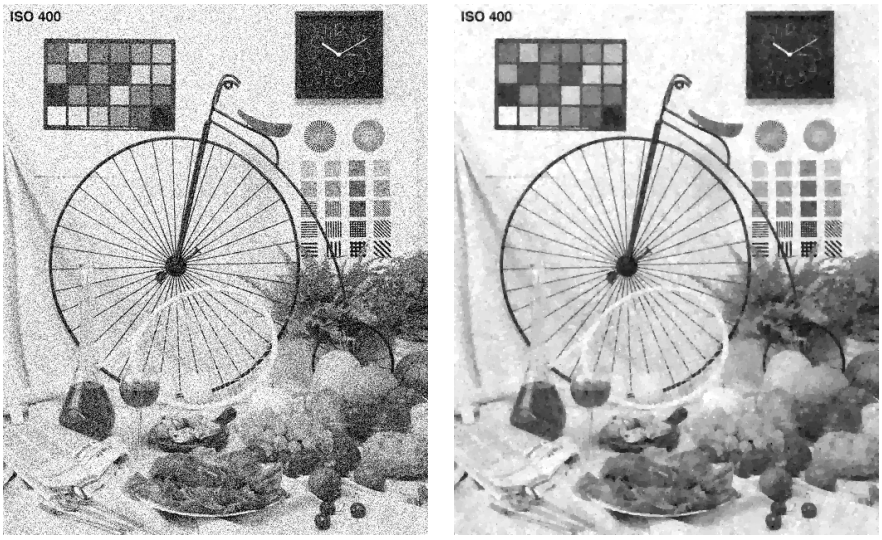


Figure 9: From left to right: noisy bike image, restored bike image by anisotropic TV.

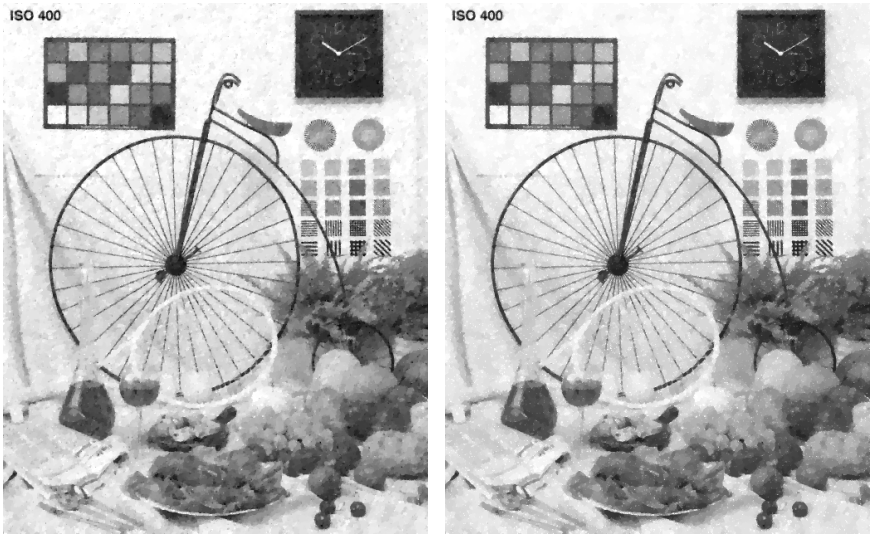


Figure 10: From left to right: restored bike images by isotropic TV and upwind TV.

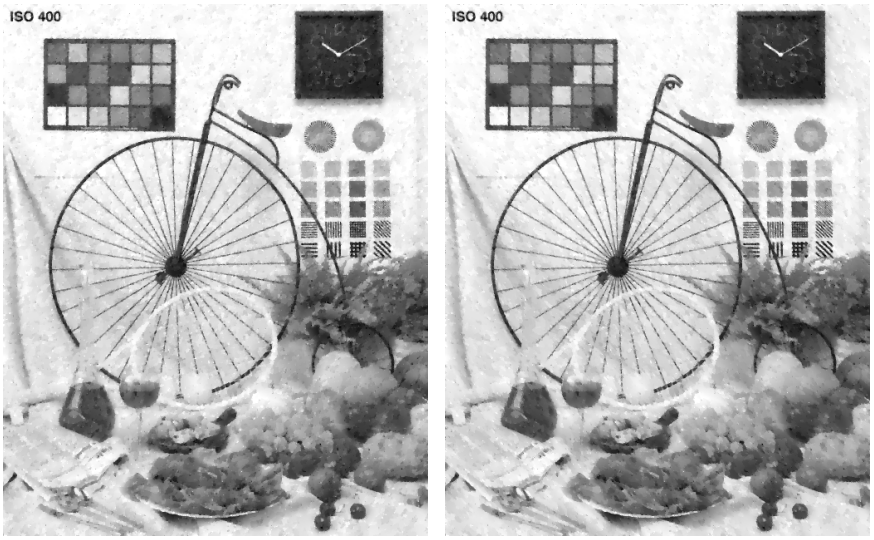


Figure 11: From left to right: restored bike images by TGV and the new proposed method.

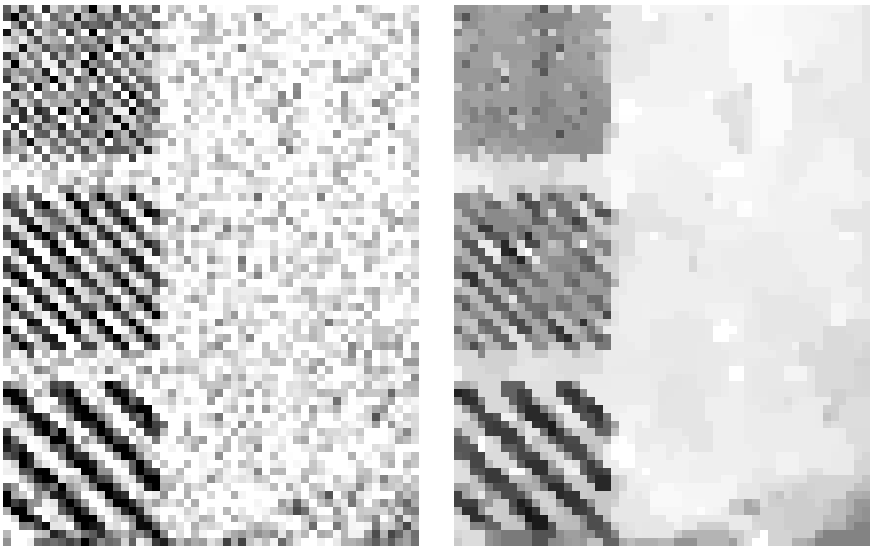


Figure 12: Zoomed-in regions of bike image. From left to right: noisy image, restored by anisotropic TV.

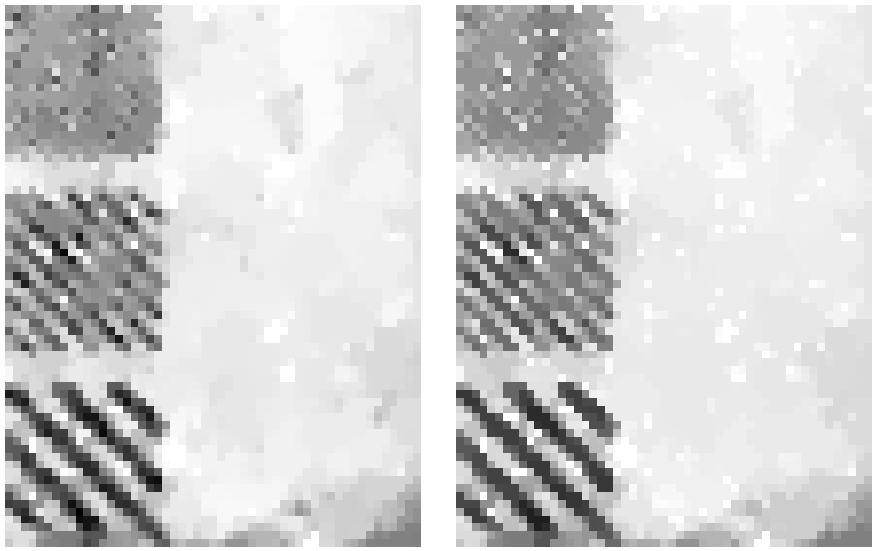


Figure 13: Zoomed-in regions of bike image. From left to right: restored by isotropic TV and upwind TV.

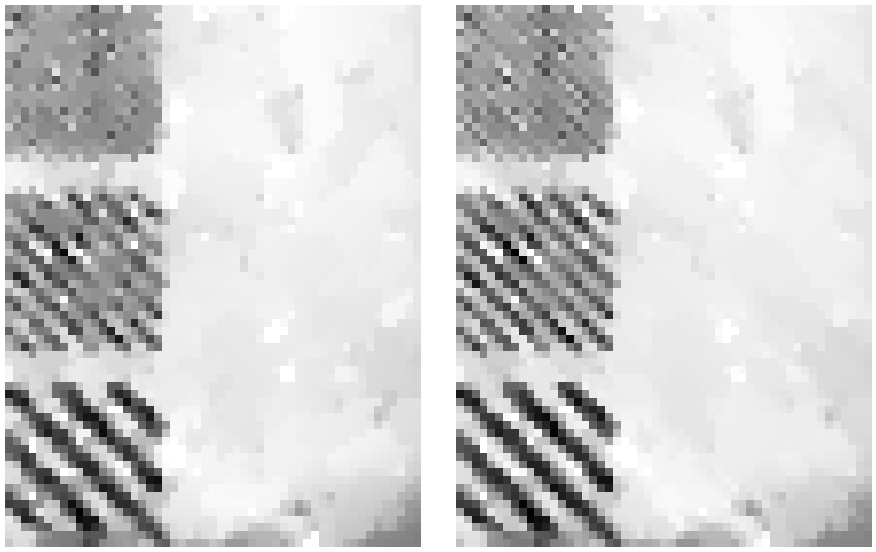


Figure 14: Zoomed-in regions of bike image. From left to right: restored by TGV and the new proposed method.

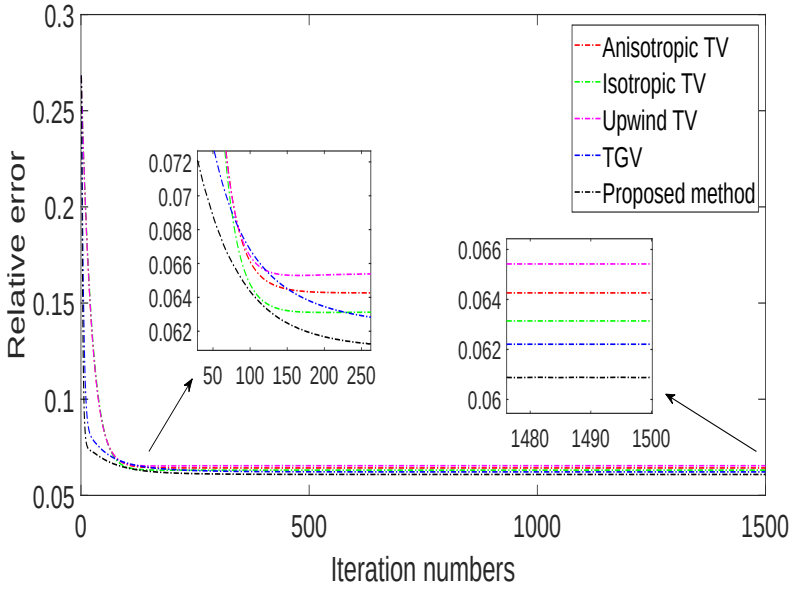


Figure 15: The sequence of relative error of bird image versus iteration numbers.

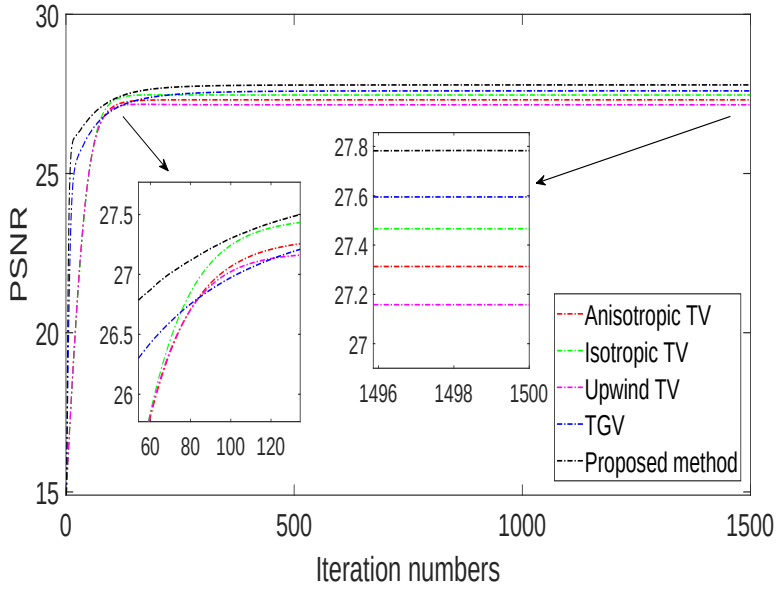


Figure 16: The sequence of PSNR of bird image versus iteration numbers.

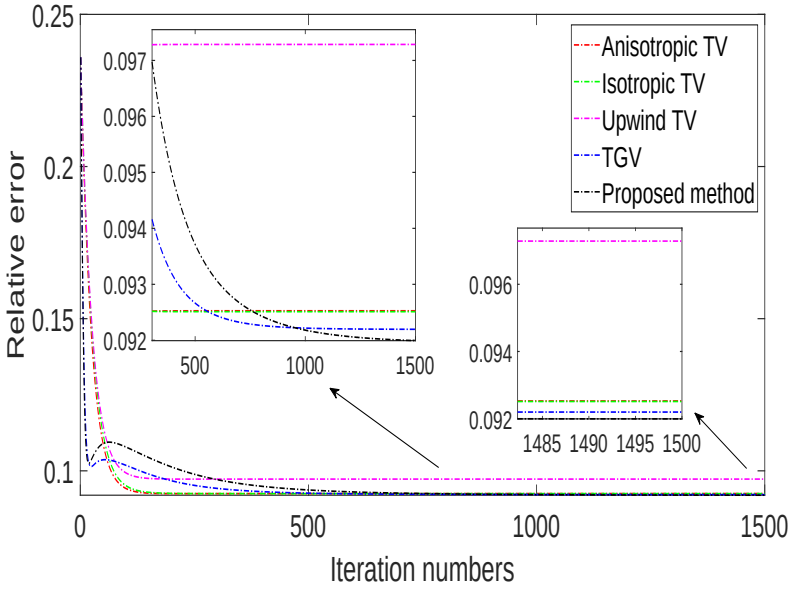


Figure 17: The sequence of relative error of bike image versus iteration numbers.

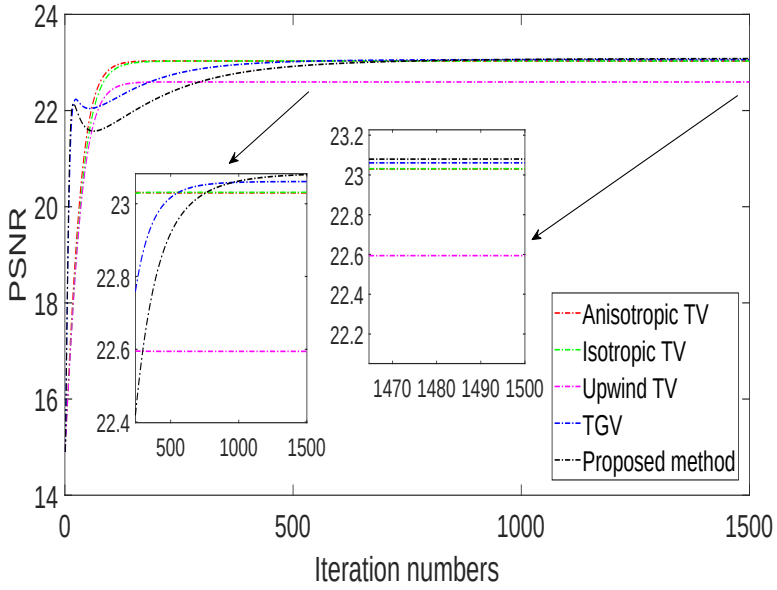


Figure 18: The sequence of PSNR of bike image versus iteration numbers.

Table 1: Performance comparison of restoration results in terms of relative error and the PSNR. The new proposed method achieves higher quality in terms of the PSNR and relative error.

Image	Method	Iterations	Optimal λ	Relative error	PSNR
Bird	Anisotropic TV	1500	0.14	0.064262	27.313069
Bird	Isotropic TV	1500	0.17	0.063141	27.466009
Bird	Upwind TV	1500	0.21	0.065420	27.158004
Bird	TGV	1500	(0.3, 0.16)	0.062209	27.595121
Bird	proposed	1500	(0.3, 0.13)	0.060880	27.782706
Bike	Anisotropic TV	1500	0.11	0.092527	23.029824
Bike	Isotropic TV	1500	0.13	0.092511	23.031338
Bike	Upwind TV	1500	0.155	0.097285	22.594246
Bike	TGV	1500	(0.21, 0.13)	0.092196	23.060928
Bike	proposed	1500	(0.14, 0.1)	0.091993	23.080060

5 Conclusion

This paper proposes to revisit the discreteTGV image denoising problem by redefining the operations via the inclusion of a diagonal term to reduce the staircasing effect, which is the patchy artifacts usually observed in slanted regions of the image. Numerical experiments confirm that the new proposed method achieves higher quality in terms of relative error and the PSNR compared with the conventional TGV method. The more direction analysis of first-order-derivative by using more oblique terms is considered as our future researches.

References

1. Aujol, J.F., Gilboa, G. and Papadakis, N. *Fundamentals of non-local total variation spectral theory*, In International Conference on Scale Space and Variational Methods in Computer Vision, Springer, Cham, (2015), 66–77.
2. Bauschke, H.H. and Combettes, P.L. *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, New York, 2011.
3. Bioucas–Dias, J.M. and Figueiredo, M.A. *A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration*, IEEE Transactions on Image processing, 16(12) (2007), 2992–3004.
4. Blomgren, P. and Chan, T.F. *Color TV: total variation methods for restoration of vector-valued images*, IEEE transactions on image processing, 7(3) (1998), 304–309.

5. Blomgren, P., Chan, T.F., Mulet, P. and Wong, C.K. *Total variation image restoration: numerical methods and extensions*, In Proceedings of International Conference on Image Processing, IEEE, 3 (1997), 384–387.

6. Bredies, K., Kunisch, K. and Pock, T. *Total generalized variation*, SIAM Journal on Imaging Sciences, 3(3) (2010), 492–526.

7. Bredies, K. and Valkonen, T. *Inverse problems with second-order total generalized variation constraints*, Proceedings of SampTA 201 (2011).

8. Bresson, X. and Chan, T.F. *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse problems and imaging, 2(4) (2008), 455–484.

9. Burns, M., Haidacher, M., Wein, W., Viola, I. and Groeller, E. *Feature emphasis and contextual cutaways for multimodal medical visualization*, In EuroVis, 7 (2007), 275–282.

10. Caselles, V., Chambolle, A. and Novaga, M. *The discontinuity set of solutions of the TV denoising problem and some extensions*, Multiscale modeling & simulation, 6(3) (2007), 879–894.

11. Caselles, V., Chambolle, A. and Novaga, M. *Regularity for solutions of the total variation denoising problem*, Revista Matematica Iberoamericana, 27(1) (2011), 233–252.

12. Chambolle, A., Levine, S.E. and Lucier, B.J. *An upwind finite difference method for total variation-based image smoothing*, SIAM Journal on Imaging Sciences, 4(1) (2011), 277–299.

13. Chan, T.F., Osher, S. and Shen, J. *The digital TV filter and nonlinear denoising*, IEEE Transactions on Image processing, 10(2) (2001), 231–241.

14. Chan, T.F. and Wong, C.K. *Total variation blind deconvolution*, IEEE transactions on Image Processing, 7(3) (1998), 370–375.

15. Chan, T.F., Yip, A.M. and Park, F.E. *Simultaneous total variation image inpainting and blind deconvolution*, International Journal of Imaging Systems and Technology, 15(1) (2005), 92–102.

16. Condat, L. *A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms*, Journal of optimization theory and applications, 158(2) (2013), 460–79.

17. Elad, M. and Feuer, A. *Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images*, IEEE transactions on image processing, 6(12) (1997), 1646–1658.

18. Fadili, J.M. and Peyr, G. *Total variation projection with first order schemes*, IEEE Transactions on Image Processing, 20(3) (2011), 657–669.

19. Georgiev, T.G. and Chunev, G.N. *Methods and apparatus for rendering output images with simulated artistic effects from focused plenoptic camera data*, U.S. Patent 8,665,341 (2014).

20. Guichard, F. and Malgouyres, F. *Total variation based interpolation*, In 9th European Signal Processing Conference, IEEE, (1998), 1–4.

21. Lou, Y., Zeng, T., Osher, S. and Xin, J. *A weighted difference of anisotropic and isotropic total variation model for image processing*, SIAM Journal on Imaging Sciences, 8(3) (2015), 1798–1823.

22. Nacereddine, N., Zelmat, M., Belaifa, S.S. and Tridi, M. *Weld defect detection in industrial radiography based digital image processing*, Transactions on Engineering Computing and Technology 2 (2005), 145–148.

23. Richardson, W.H. *Bayesian-based iterative method of image restoration*, Journal of the Optical Society of America, 62(1) (1972), 55–59.

24. Ring, W. *Structural properties of solutions to total variation regularization problems*, ESAIM: Mathematical Modelling and Numerical Analysis, 34(4) (2000), 799–810.

25. Rudin, L.I. and Osher, S. *Total variation based image restoration with free local constraints*, In Proceedings of 1st International Conference on Image Processing, IEEE, 1 (1994), 31–35.

26. Rudin, L.I., Osher, S. and Fatemi, E. *Nonlinear total variation based noise removal algorithms*, Physica D: nonlinear phenomena, 60(1-4) (1992), 259–268.

27. Santhanam, A.P., Fidopiastis, C.M., Hamza-Lup, F.G., Rolland, J.P. and Imielinska, C.Z. *Physically-based deformation of high-resolution 3D lung models for augmented reality based medical visualization*, MICCAI AMI-ARCS, (2004), 21–32.

28. Svakhine, N.A., Ebert, D.S. and Andrews, W.M. *Illustration-inspired depth enhanced volumetric medical visualization*, IEEE Transactions on Visualization and Computer Graphics, 15(1) (2009), 77–86.

29. Vogel, C.R. and Oman, M.E. *Fast, robust total variation-based reconstruction of noisy, blurred images*, IEEE transactions on image processing, 7(6) (1998), 813–824.

30. Wachs, J., Stern, H., Edan, Y., Gillam, M., Feied, C., Smith, M. and Handler, J. *A real-time hand gesture interface for medical visualization applications*, In Applications of Soft Computing, Springer, Berlin, Heidelberg, (2006), 153–162.

31. Zach, C., Pock, T. and Bischof, H. *A duality based approach for realtime TV-L1 optical flow*, In Joint Pattern Recognition Symposium, Springer, Berlin, Heidelberg, (2007), 214–223.

32. Zhang, D.D., Kong, W.K.A., You, J. and Wong, M. *Online palmprint identification*, IEEE Transactions on pattern analysis and machine intelligence, 25(9) (2003), 1041–1050.

33. Zhang, Z. and Blum, R.S. *Region-based image fusion scheme for concealed weapon detection*, In Proceedings of the 31st Annual Conference on Information Sciences and Systems, (1997), 168–173.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 165–183
DOI:10.22067/ijnao.v9i2.74000

Research Article

Solving linear optimal control problems of the time-delayed systems by Adomian decomposition method

S.M. Mirhosseini-Alizamini*

Abstract

We apply the Adomian decomposition method (ADM) to obtain a suboptimal control for linear time-varying systems with multiple state and control delays and with quadratic cost functional. In fact, the nonlinear two-point boundary value problem, derived from Pontryagin's maximum principle, is solved by ADM. For the first time, we present here a convergence proof for ADM. In order to use the proposed method, a control design algorithm with low computational complexity is presented. Through the finite iterations of algorithm, a suboptimal control law is obtained for the linear time-varying multi-delay systems. Some illustrative examples are employed to demonstrate the accuracy and efficiency of the proposed methods.

AMS(2010): 49N05; 93C05.

Keywords: Multiple time-delay systems; Pontryagin's maximum principle; Adomian decomposition method.

1 Introduction

Optimal control of time-delay systems is one of the most challenging mathematical problems in control theory. Indeed, the presence of delay makes analysis and control design much more complicated. Delays frequently occur in mechanics, physics, population dynamics, biological, chemical, electronic, and transformation systems; see [14]. The theory and the application of optimal control for linear time-delay systems have been developed perfectly. However, as for nonlinear systems, synthesis problems that are solved by classical control theory lead to difficult computations. It is well known

*Corresponding author
Received 8 July 2018; revised 18 June 2019; accepted 24 June 2019
S.M. Mirhosseini-Alizamini
Department of Mathematics, Payame Noor University (PNU), Tehran, Iran. e-mail: m_mirhosseini@pnu.ac.ir

that the nonlinear optimal control time-delay systems can be reduced to a two-point boundary value problem (TPBVP) involving both delay and advance delay terms, implementing Pontryagin's maximum principle (PMP); see [15]. In general, this TPBVP cannot be solved exactly and most researches have been devoted to finding an approximate solution, for nonlinear TPBVP. We briefly review some recent papers that are relevant to the method developed in the current work for time-delay optimal control problem. An averaging approximations for time-delay optimal control problems [5], the B-spline approximation scheme [8,24], the PMP [9], variational iteration method (VIM) [25–28], a novel feedforward-feedback suboptimal control of linear time-delay systems [13], Haar wavelets approach [29], hybrid of block-pulse functions and orthonormal basis [7, 11, 19, 22, 30], composite Chebyshev finite difference method [20], the Hamilton-Jacobi-Bellman equation [6], a delay-dependent stability of neutral systems [4], an interior-point algorithm [34,35], and an embedding process that transfers the problem to a new optimal measure problem [17].

The topic of the Adomian decomposition method (ADM) has been rapidly growing in recent years. It was first proposed by Adomian [1, 2]. In this method, the solution of functional equations is considered as the sum of an infinite series usually converging to the solution. A lot of research works have been conducted recently in applying this method to a class of linear and nonlinear partial differential equations; see [33]. The Adomian's decomposition has many advantages: It does not require any kind of discretization, linearization, or perturbation of the variables and the equation, therefore it does not need any modification of the actual model that could change the solution; it is efficient on providing an approximate or even exact solution in a closed form, to linear and nonlinear problems and provides a fast and accurate convergent series, and therefore it is only necessary to calculate a few terms of the series in order to obtain a reliable approximate solution; the method depends only on the known function $u_0(t)$ and the algorithm is of simple implementation. The method, has been widely applied to solve nonlinear problems, and different modifications are suggested to overcome the demerits arising in the solution procedure [3,32].

This paper concerns with a class of nonlinear quadratic optimal control problem with multi-delay systems. Applying the main ideas of the shooting method to the basic and also an ADM. By applying the necessary optimality conditions, we obtain iterative formulas for the ADM. By using the finite-step iteration of algorithm, we obtain a suboptimal control law. The convergence of the ADM is studied and for illustrating the effectiveness of these methods, some test problems are investigated. Four illustrative examples are given to demonstrate the simplicity and efficiency of the proposed method.

The structure of this paper is arranged as follows: Section 2 is devoted to Pontryagin's maximum principle used for solving linear time-varying multi-delay system. Section 3 is dedicated to the proposed design approach to solve a closed loop optimal control problem based on the ADM and conver-

gence of the method is demonstrated. Section 4 is devoted to the suboptimal control strategy and algorithm for the proposed method. In Section 5, the numerical examples are simulated to show the reasonableness of our theory and demonstrate the performance of our network. Finally, we end this paper with conclusions in Section 6.

2 Problem statement and optimality conditions

Consider the following linear time-varying multi-delay system:

$$\begin{cases} \dot{x}(t) = A(t)x(t) + A_1(t)x(t-\tau_x) + B(t)u(t) + B_1(t)u(t-\tau_u), \\ x(t) = \phi(t), \qquad t_0 - \tau_x \leqslant t \leqslant t_0, \\ u(t) = \psi(t), \qquad t_0 - \tau_u \leqslant t \leqslant t_0, \end{cases} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$, are the state and control vectors, respectively; $A(t)$, $A_1(t)$, $B(t)$, and $B_1(t)$ are real, piecewise continuous matrices of appropriate dimensions defined on the appropriate intervals; $\phi(t)$ and $\psi(t)$ are specified initial functions; τ_x and τ_u are constant positive scalars. Here, it is assumed that system (1) is controllable and that $\tau_u < \tau_x$. Find the control signal $u(t)$ that minimizes the cost functional:

$$J = \frac{1}{2}x^T(t_f)Q_f x(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left(x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)\right) dt, \tag{2}$$

where, the matrix $Q_f \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite, $Q(t) \in \mathbb{R}^{n\times n}$ and $R(t) \in \mathbb{R}^{m\times m}$ are chosen to be positive semidefinite and positive definite matrices, respectively.

The Hamiltonian function for the problem is

$$\begin{aligned} \mathcal{H}(x,u,\lambda,t) = {} & \frac{1}{2}x^T(t)Q(t)x(t) + \frac{1}{2}u^T(t)R(t)u(t) \\ & + \lambda^T(t)[A(t)x(t) + A_1(t)x(t-\tau_x) + B(t)u(t) + B_1(t)u(t-\tau_u)], \end{aligned} \tag{3}$$

where $\lambda(t) \in \mathbb{R}^n$ is the vector of the Lagrange multiplier. According to the necessary conditions for optimality, we can obtain the following nonlinear TPBVP [15,31]:

$$\dot{x}(t) = \begin{cases} A(t)x(t) + A_1(t)x(t-\tau_x) - (S_1(t) + S_2(t))\lambda(t) \\ -S_3(t)\lambda(t+\tau_u) - S_4(t)\lambda(t-\tau_u), \quad t_0 \leqslant t < t_f - \tau_u, \\ A(t)x(t) + A_1(t)x(t-\tau_x) - S_1(t)\lambda(t) \\ -S_4(t)\lambda(t-\tau_u), \quad t_f - \tau_u \leqslant t \leqslant t_f \end{cases} \tag{4}$$

and

$$\dot{\lambda}(t) = \begin{cases} -Q(t)x(t) - A^T(t)\lambda(t) \\ -A_1^T(t+\tau_x)\lambda(t+\tau_x), \quad t_0 \leqslant t < t_f - \tau_x, \\ -Q(t)x(t) - A^T(t)\lambda(t), \quad t_f - \tau_x \leqslant t \leqslant t_f \end{cases} \tag{5}$$

with initial conditions

$$\begin{cases} x(t) = \phi(t), \qquad t_0 - \tau_x \leq t \leqslant t_0, \\ u(t) = \psi(t), \qquad t_0 - \tau_u \leqslant t \leqslant t_0, \\ \lambda(t_f) = Q_f x(t_f), \end{cases} \tag{6}$$

where

$$\begin{aligned} S_1(t) &= B(t)R^{-1}(t)B^T(t), \\ S_2(t) &= B_1(t)R^{-1}(t-\tau_u)B_1^T(t), \\ S_3(t) &= B(t)R^{-1}(t)B_1^T(t+\tau_u), \\ S_4(t) &= B_1(t)R^{-1}(t-\tau_u)B^T(t-\tau_u), \end{aligned}$$

$x(t-\tau)$ is the time-delay term and $\lambda(t+\tau)$ is the time-advance term. Also, the optimal control law is obtained by

$$u^*(t) = \begin{cases} -R^{-1}(t)B^T(t)\lambda(t) \\ -R^{-1}(t)B_1^T(t+\tau_u)\lambda(t+\tau_u), \quad t_0 \leqslant t < t_f - \tau_u, \\ -R^{-1}(t)B^T(t)\lambda(t), \quad t_f - \tau_u \leqslant t \leqslant t_f. \end{cases} \tag{7}$$

The optimal can be implemented as a closed loop optimal if the co-state vector obtained consists of linear function of the states and a nonlinear term, which is the adjoint vector sequence, in the form

$$\lambda(t) = P(t)x(t) + g(t), \qquad \lambda(t_f) = Q_f x(t_f), \tag{8}$$

where $P(t) \in \mathbb{R}^{n\times n}$ is an unknown positive-semidefinite function matrix and $g(t) \in \mathbb{R}^n$ is the adjoint vector.

Substituting (8) into equation (4) yields

$$\begin{aligned} \dot{x}(t) &= [A(t) - S_1(t)P(t)]\, x(t) - S_1(t)g(t) + A_1(t)x(t-\tau_x) + F(t), \\ x(t) &= \phi(t), \qquad t_0 - \tau \leq t \leqslant t_0, \end{aligned} \tag{9}$$

where

$$F(t) = \begin{cases} -S_2(t)\,[P(t)x(t) + g(t)] - S_3(t)\,[P(t+\tau_u)x(t+\tau_u) + g(t+\tau_u)] \\ -S_4(t)\,[P(t-\tau_u)x(t-\tau_u) + g(t-\tau_u)]\,, \ t_0 \leqslant t < t_f - \tau_u, \\ -S_4(t)\,[P(t-\tau_u)x(t-\tau_u) + g(t-\tau_u)]\,, \ \ t_f - \tau_u \leqslant t \leqslant t_f. \end{cases} \tag{10}$$

Computing the derivatives to the both sides with respect to t of equation (8), we have

$$\begin{aligned}\dot{\lambda}(t) &= \dot{P}(t)x(t) + P(t)\dot{x}(t) + \dot{g}(t), \qquad t_0 \leqslant t \leqslant t_f \\ &= \left[\dot{P}(t) + P(t)A(t) - P(t)S_1(t)P(t)\right] x(t) - P(t)S_1(t)g(t) \\ &+ P(t)A_1(t)x(t-\tau_x) + P(t)F(t) + \dot{g}(t). \end{aligned} \tag{11}$$

Putting (8) into equation (5), we get

$$\dot{\lambda}(t) = \begin{cases} -Q(t)x(t) - A^T(t)P(t)x(t) - A^T(t)g(t) \\ -A_1^T(t+\tau_x)[P(t+\tau_x)x(t+\tau_x) + g(t+\tau_x)], t_0 \leqslant t < t_f - \tau_x, \\ -Q(t)x(t) - A^T(t)P(t)x(t) - A^T(t)g(t), t_f - \tau_x \leqslant t \leqslant t_f. \end{cases} \tag{12}$$

Thus, from (11) and (12), we can obtain the following Riccati matrix differential equation:

$$-\dot{P}(t) = P(t)A(t) + A^T(t)P(t) - P(t)S_1(t)P(t) + Q(t), \; P(t_f) = Q_f, \tag{13}$$

and the following adjoint vector differential equation:

$$\dot{g}(t) = -\left[A(t) - S_1(t)P(t)\right]^T g(t) - P(t)A_1(t)x(t-\tau_x) + G(t), \; g(t_f) = 0, \tag{14}$$

where

$$G(t) = \begin{cases} -A_1^T(t+\tau_x)\left[P(t+\tau_x)x(t+\tau_x) + g(t+\tau_x)\right] + P(t)S_2(t)\left[P(t)x(t) + g(t)\right] \\ -P(t)S_3(t)\left[P(t+\tau_u)x(t+\tau_u) + g(t+\tau_u)\right] \\ P(t)S_4(t)\left[P(t-\tau_u)x(t-\tau_u) + g(t-\tau_u)\right], \qquad t_0 \leqslant t < t_f - \tau_x, \\ P(t)S_2(t)\left[P(t)x(t) + g(t)\right] + P(t)S_3(t)\left[P(t+\tau_u)x(t+\tau_u) + g(t+\tau_u)\right] \\ P(t)S_4(t)\left[P(t-\tau_u)x(t-\tau_u) + g(t-\tau_u)\right], \qquad t_f - \tau_x \leqslant t < t_f - \tau_u, \\ P(t)S_4(t)\left[P(t-\tau_u)x(t-\tau_u) + g(t-\tau_u)\right], \qquad t_f - \tau_u \leqslant t \leqslant t_f. \end{cases} \tag{15}$$

Substituting (8) into (7) yields

$$u^*(t) = \begin{cases} -R^{-1}(t)B^T(t)[P(t)x(t) + g(t)] \\ -R^{-1}(t)B_1^T(t+\tau_u)[P(t+\tau_u)x(t+\tau_u) + g(t+\tau_u)], t_0 \leqslant t < t_f - \tau_u, \\ -R^{-1}(t)B^T(t)[P(t)x(t) + g(t)], t_f - \tau_u \leqslant t \leqslant t_f. \end{cases}$$

For the sake of simplicity, let us define the right hand sides of (9) and (14) as follows:

$$f_1(t,x,g) = \left[A(t) - S_1(t)P(t)\right] x(t) - S_1(t)g(t) + A_1(t)x(t-\tau_x) + F(t), \tag{16}$$

$$f_2(t,x,g) == -\left[A(t) - S_1(t)P(t)\right]^T g(t) - P(t)A_1(t)x(t-\tau_x) + G(t), \tag{17}$$

where $F(t)$ and $G(t)$ are relations (10) and (15), respectively.
Thus the TPBVP in (9) and (14) changes to

$$\begin{cases} \dot{x}(t) = f_1(t, x, g), \\ \dot{g}(t) = f_2(t, x, g), \\ x(t_0) = x_0, \quad g(t_f) = 0. \end{cases} \tag{18}$$

Note that, relations (18) form a nonlinear TPBVP with time-varying coefficient involving both delay and advance terms. The exact solution of this problem is, in general, extremely difficult, if not impossible. In the next section, we propose another analytic approximate method based on ADM, for this purpose.

3 Adomian decomposition method

In order to illustrate the basic concepts of the ADM, we consider the following equation:

$$\mathcal{L}(u) + \mathcal{R}(u) + \mathcal{N}(u) = h(t), \tag{19}$$

where $u(t)$ is an unknown function, $\mathcal{L}$ is a linear operator, that is assumed to be invertible, $\mathcal{R}$ is another linear differential operator, $\mathcal{N}(u)$ represents the nonlinear terms, and h is a continuous function. Applying the inverse operator $\mathcal{L}^{-1}$ to both sides of (19) and using the given conditions, we obtain

$$u = f - \mathcal{L}^{-1}(\mathcal{R}(u)) - \mathcal{L}^{-1}(\mathcal{N}(u)), \tag{20}$$

where the function $f(t)$ represents the terms arising from integrating the function $h(t)$ and using the initial condition.

The standard Adomian method defines the solution $u(t)$ of (19) as a series

$$u(t) = \sum_{n=0}^{\infty} u_n(t), \tag{21}$$

where the components $u_n(t)$ are usually determined recurrently. Substituting this infinite series into (20) leads to

$$\sum_{n=0}^{\infty} u_n(t) = f(t) - \mathcal{L}^{-1}\left(\mathcal{R}(\sum_{n=0}^{\infty} u_n(t))\right) - \mathcal{L}^{-1}\left(\mathcal{N}(\sum_{n=0}^{\infty} u_n(t))\right). \tag{22}$$

The nonlinear term in (21) can be computed by substituting

$$\mathcal{N}(u) = \sum_{n=0}^{\infty} A_n(u_0, u_1, \ldots, u_n), \tag{23}$$

where A_n is the Adomian polynomials, which can be determined by

$$A_n = \frac{1}{n!}\frac{\partial^n}{\partial q^n}\mathcal{N}\left[\sum_{n=0}^{\infty} q^k u_k\right]_{q=0}, \quad n = 1, 2, 3, \ldots, . \tag{24}$$

Now, substituting (23) into (22) leads to

$$\sum_{n=0}^{\infty} u_n(t) = f(t) - \mathcal{L}^{-1}\left(\mathcal{R}(\sum_{n=0}^{\infty} u_n(t))\right) - \mathcal{L}^{-1}\left(\sum_{n=0}^{\infty} A_n\right).$$

Each term of series (21) is given by the recurrent relation

$$\begin{aligned} u_o &= f(t) \\ u_n &= -\mathcal{L}^{-1}(\mathcal{R}(u_{n-1})) - \mathcal{L}^{-1}(A_{n-1}), \quad n \geq 1. \end{aligned}$$

Now, we briefly describe how to apply the ADM to systems (18). For this purpose, we use a shooting method like procedure combine with the ADM for solving TPBVP in (18).

Remark. It is necessary to transform the boundary value problem into a initial value problem. Therefore, we must find $\alpha \in \mathbb{R}$ such that the condition $g(t_f) = 0$ can be replaced by the condition $g(t_0) = \alpha$. Thus we rewrite the TPBVP (18) as follows:

$$\begin{cases} \dot{x}(t) = f_1(t, x, g), \\ \dot{g}(t) = f_2(t, x, g), \\ x(t_0) = x_0, \quad g(t_0) = \alpha, \end{cases} \tag{25}$$

where $\alpha \in \mathbb{R}$ is an unknown parameter. This parameter will be identified after sufficient iterations of ADM.

Based on the ADM, we seek the solution $\{x, g\}$ as follows:

$$x = \lim_{N\to\infty} \sum_{n=0}^{N} x_n, \qquad g = \lim_{N\to\infty} \sum_{n=0}^{N} g_n,$$

and hence the recursive relationship is found as

$$\begin{cases} x_{n+1} = \mathcal{L}^{-1} A_{1,n}, \quad n \geq 0, \\ g_{n+1} = \mathcal{L}^{-1} A_{2,n}, \quad n \geq 0, \\ x(t_0) = x_0, \quad g(t_0) = \alpha, \end{cases}$$

with inverse $\mathcal{L}^{-1}(.) = \int_0^t (.)dt$ and

$$f_k(t, x_n, g_n) = \sum_{n=0}^{\infty} A_{k,n}, \quad k = 1, 2,$$

where $A_{k,n}$ are the Adomian polynomials and are calculated by

$$A_{k,n} = \frac{1}{n!} \frac{\partial^n}{\partial q^n} f_k \left(t, \sum_{n=0}^{\infty} q^k x_k, \sum_{n=0}^{\infty} q^k g_k \right)_{q=0}, \quad n = 0, 1, 2, \ldots. \tag{26}$$

Take the first $n+1$ terms of the nth approximation of x and g as follows:

$$\begin{cases} \Phi_n = x_0 + \sum_{i=1}^{n} \mathcal{L}^{-1}(A_{1,i-1}), \\ \Psi_n = g_0 + \sum_{i=1}^{n} \mathcal{L}^{-1}(A_{2,i-1}). \end{cases} \tag{27}$$

Find the sequences $\Phi_n = x_0 + \cdots + x_n$ and $\Psi_n = g_0 + \cdots + g_n$ such that

$$\begin{cases} \Phi_n = x_0 + \mathcal{L}^{-1}(f_1(t, \Phi_{n-1}, \Psi_{n-1})), & n \geq 1, \\ \Psi_n = g_0 + \mathcal{L}^{-1}(f_2(t, \Phi_{n-1}, \Psi_{n-1})), & n \geq 1, \end{cases} \tag{28}$$

where $x_0(t) = x(t_0) = x_0, \;\; g_0(t) = g(t_0) = \alpha$.

Theorem 1. *Assume that* $\{\sum_{k=0}^{n} x_k(t)\}$ *and* $\{\sum_{k=0}^{n} g_k(t)\}$ *are the solution sequences produced by ADM formula* (28)*, which converge, respectively, to* $\widehat{x}(t, \alpha)$ *and* $\widehat{g}(t, \alpha)$*, as* $n \to \infty$*. Then* $\widehat{x}(t, \alpha), \widehat{g}(t, \alpha)$ *are the exact solutions of* (25)*. Accordingly,* $\widehat{x}(t, \widehat{\alpha}), \widehat{g}(t, \widehat{\alpha})$ *are the exact solutions of* (18) *when* $\widehat{\alpha}$ *is the real root of* $\widehat{g}(t_f, \alpha) = 0$.

Proof. Let consider TPBVP (25) as follows:

$$\begin{cases} x(t) = x(t_0) + \mathcal{L}^{-1}[f_1(t, x, g)], \\ g(t) = g(t_0) + \mathcal{L}^{-1}[f_2(t, x, g)], \\ x(t_0) = x_0, \;\; g(t_0) = \alpha, \end{cases}$$

where $\mathcal{L} = \dfrac{d}{dt}(\cdot)$ and $\mathcal{L}^{-1} = \int_{t_0}^{t} (\cdot) dt$. We have

$$x_n(t) = \int_{t_0}^{t} [A_{1,n-1}(x_0, x_1, \cdots, x_{n-1}, g_0, g_1, \cdots, g_{n-1})]\, ds, \quad n \geq 1, \tag{29}$$

$$g_n(t) = \int_{t_0}^{t} [A_{2,n-1}(x_0, x_1, \cdots, x_{n-1}, g_0, g_1, \cdots, g_{n-1})]\, ds, \quad n \geq 1, \tag{30}$$

$$x_0(t) = x(t_0) = x_0, \;\; g_0(t) = g(t_0) = \alpha.$$

Taking limits of both sides of (29) and (30) as $n \to \infty$ implies

$$\lim_{n\to\infty}\sum_{k=1}^{n}x_k(t)=\int_{t_0}^{t}\left[\lim_{n\to\infty}\sum_{k=1}^{n}A_{1,k-1}(x_0,x_1,\cdots,x_{k-1},g_0,g_1,\cdots,g_{k-1})\right]ds,$$
$$\lim_{n\to\infty}\sum_{k=1}^{n}g_k(t)=\int_{t_0}^{t}\left[\lim_{n\to\infty}\sum_{k=1}^{n}A_{2,k-1}(x_0,x_1,\cdots,x_{k-1},g_0,g_1,\cdots,g_{k-1})\right]ds.$$

Then

$$\widehat{x}(t,\alpha)=\int_{t_0}^{t}[f_1(s,\widehat{x}(s,\alpha),\widehat{g}(s,\alpha))]ds,$$
$$\widehat{g}(t,\alpha)=\int_{t_0}^{t}[f_2(s,\widehat{x}(s,\alpha),\widehat{g}(s,\alpha))]ds.$$

Differentiating both sides with respect to t yields

$$\dot{\widehat{x}}(t,\alpha)=f_1(t,\widehat{x}(t,\alpha),\widehat{g}(t,\alpha)),$$
$$\dot{\widehat{g}}(t,\alpha)=f_2(t,\widehat{x}(t,\alpha),\widehat{g}(t,\alpha)).$$

Moreover, if $t=t_0$, then from (29) and (30), $x_n(t_0)=0, g_n(t_0)=0$ for every $n\geq 1$. Thus

$$\sum_{k=0}^{n}x_k(t_0)=x_0(t_0)=x_0,\quad \sum_{k=0}^{n}g_k(t_0)=g_0(t_0)=\alpha,$$

or equivalently, $\widehat{x}(t_0,\alpha)=x_0$, $\widehat{g}(t_0,\alpha)=\alpha$. Hence, $\widehat{x}(t,\alpha)$ and $\widehat{g}(t,\alpha)$ are the exact solutions of (25). In addition, they are the exact solutions of (18), only if the condition $g(t_f)=0$ is satisfied. So, it is straightforward to choose the unknown parameter $\alpha\in\mathcal{R}^n$ such that $\widehat{g}(t_f,\alpha)=0$. Denoting this real root of $\widehat{g}(t_f,\alpha)=0$ by $\widehat{\alpha}$ completes the proof. □

4 Suboptimal control design strategy

Consider the linear time-varying multi-delay system (1) with cost functional (2). Then, the Nth order suboptimal trajectory-control pair is obtained as follows:

$$\begin{cases}x_N(t)=\sum_{k=0}^{N}x_k(t),\\ g_N(t)=\sum_{k=0}^{N}g_k(t),\end{cases}\tag{31}$$

and

$$u_N(t) = \begin{cases} -R^{-1}(t)B^T(t)[P(t)x_N(t) + g_N(t)] \\ -R^{-1}(t)B_1^T(t+\tau_u)[P(t+\tau_u)x_N(t+\tau_u) \\ +g_N(t+\tau_u)], \quad t_0 \leqslant t < t_f - \tau_u, \\ -R^{-1}(t)B^T(t)[P(t)x_N(t) + g_N(t)], \quad t_f - \tau_u \leqslant t \leqslant t_f. \end{cases} \tag{32}$$

The integer N in (31) and (32) is generally determined according to a concrete control precision. Then, the following cost functional can be calculated:

$$J_N = \frac{1}{2}x_N^T(t_f)Q_f x_N(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left(x_N^T(t)Q(t)x_N(t) + u_N^T(t)R(t)u_N(t)\right) dt, \tag{33}$$

The Nth order in (31) and (32) has the desirable accuracy, if for given positive constant $\epsilon > 0$, the following condition holds jointly:

$$\left|\frac{J_N - J_{N-1}}{J_N}\right| < \epsilon, \tag{34}$$

If the tolerance error bound is chosen small enough, the Nth order suboptimal control law will be very close to $u^*(t)$, and thus, the value of cost functional in (33) and its optimal value J^* will be almost identical.

Algorithm: Suboptimal control law of system (1):

Step 1: Obtain $P(t)$ from (13). Let $x_0(t) = x(t_0) = \phi(t), g_0(t) = g(t_0)$ and $k = 1$.

Step 2: Compute $x_k(t)$ and $g_k(t)$ from (29) and (30).

Step 3: Let $N = k$ and obtain $x_N(t)$ and $u_N(t)$ from (31) and (32).

Step 4: Calculate J_N according to (33). If $\left|\frac{J_N - J_{N-1}}{J_N}\right| < \epsilon$, then stop and output $u_N(t)$, go to step 5; else, replace k by $k+1$ and go to step 2.

Step 5: Stop the algorithm; $x_N(t)$ and $u_N(t)$ are accurate enough.

5 Numerical examples

In this section, the proposed method is illustrated by some test problems. The calculations are performed by using MATLAB software.

Example 1. Consider the following linear time-varying multi-delay systems [20]:

$$\begin{cases} \dot{x}(t) = x(t-\frac{1}{2}) + tx(t-\frac{3}{4}) + u(t), & 0 \leqslant t \leqslant 1, \\ x(t) = t+1, & -\frac{3}{4} \leqslant t \leqslant 0, \end{cases} \tag{35}$$

with the cost functional

$$J = \frac{3}{2}x^2(1) + \frac{1}{2}\int_0^1 u^2(t)dt. \tag{36}$$

The exact solutions of $x(t)$ and $u(t)$ are given by

$$x(t) = \begin{cases} \frac{239075}{420332}t^3 + \frac{3129081}{1681328}t^2 - \frac{1178769}{611392}t + 1, & 0 \leqslant t < \frac{1}{4}, \\ \frac{1}{3}t^3 + \frac{1119201}{840664}t^2 - \frac{680513}{420332}t + \frac{7039811}{7336704}, & \frac{1}{4} \leqslant t < \frac{1}{2}, \\ \frac{239075}{1681328}t^4 + \frac{3375959}{5043984}t^3 - \frac{20932555}{13450624}t^2 + \frac{1156163}{1222784}t + \frac{56216927}{161407488}, & \frac{1}{2} \leqslant t < \frac{3}{4}, \\ \frac{47815}{420332}t^5 + \frac{2306773}{10087968}t^4 - \frac{2461871}{2521992}t^3 + \frac{14865377}{53802496}t^2 + \frac{17419475}{26901248}t & \\ + \frac{2652913}{14673408}, & \frac{3}{4} \leqslant t \leqslant 1, \end{cases}$$

and

$$u(t) = \begin{cases} \frac{296893}{420332}t^2 + \frac{2078251}{840664}t - \frac{1484465}{611392}, & 0 \leqslant t < \frac{1}{4}, \\ \frac{296893}{210166}t - \frac{890679}{420332}, & \frac{1}{4} \leqslant t < \frac{1}{2}, \\ -\frac{296893}{210166}, & \frac{1}{2} \leqslant t \leqslant 1. \end{cases}$$

Furthermore, the exact value of the cost functional J is given by $J = 1.70648554$.
In order to obtain an accurate enough suboptimal control law, we applied the proposed algorithm with the tolerance bounds $\epsilon = 2\times10^{-6}$. In this case, convergence is achieved after 10 iterations, that is, $\left|\frac{J_{10} - J_9}{J_{10}}\right| = 1.757 \times 10^{-6} < 2 \times 10^{-6}$.

To analysis the accuracy and effectiveness of the ADM and VIM, the relative error of objective value is summarized in Table 1, for several iterations. Accordingly, results show that in $N = 10$, ADM converges to the exact solution. The optimal value of the cost functional is obtained as $J_{10} = 1.706485$.

We compared the results obtained from ADM and VIM with exact solution. Figure 1 shows the simulation curves of $u(t)$ at iteration time 10 and the corresponding state trajectory $x(t)$. Results of both methods are very close to exact solution as shown by Figure 1. This confirms that the proposed method yields excellent results.

Example 2. Consider the following linear time-varying multi-delay systems [7, 16, 21, 36]:

Table 1: Simulation results of Example 1 at different iteration times.

method	ADM	ADM	VIM	VIM
N	J_N	$\left\|\frac{J_N - J_{N-1}}{J_N}\right\|$	J_N	$\left\|\frac{J_N - J_{N-1}}{J_N}\right\|$
6	1.682312	–	1.678321	–
7	1.691205	5.255×10^{-2}	1.685342	4.165×10^{-2}
8	1.703156	7.016×10^{-2}	1.692335	4.132×10^{-3}
9	1.706484	1.961×10^{-2}	1.697346	2.952×10^{-3}
10	1.706485	1.757×10^{-6}	1.697349	1.767×10^{-6}

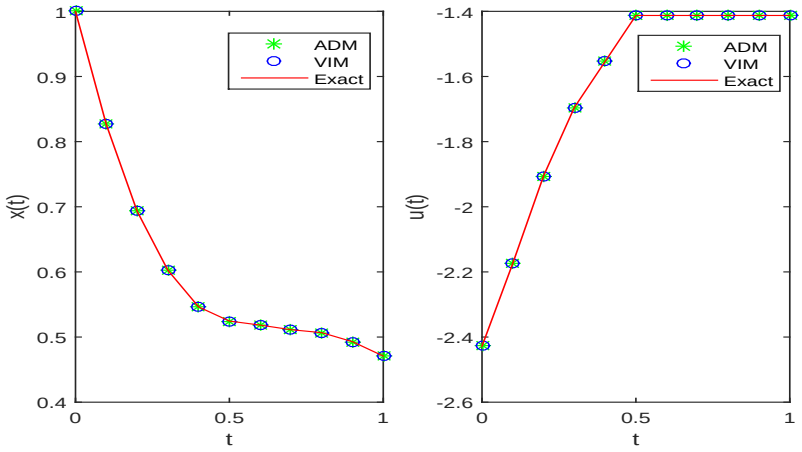


Figure 1: The suboptimal control and state, when $N = 10$ for Example 1.

$$\begin{cases} \dot{x}(t) = -x(t) + x(t - \frac{1}{3}) + u(t) - \frac{1}{2}u(t - \frac{2}{3}), & 0 \leqslant t \leqslant 1, \\ x(t) = 1, & -\frac{1}{3} \leqslant t \leqslant 0, \\ u(t) = 0, & -\frac{2}{3} \leqslant t \leqslant 0, \end{cases} \tag{37}$$

with the cost functional

$$J = \frac{1}{2}\int_0^1 \left(x^2(t) + \frac{1}{2}u^2(t)\right)(t)dt. \tag{38}$$

According to system (1), we have $A = -1, A_1 = B = 1, B_1 = -\frac{1}{2}, Q = 1, R = \frac{1}{2}$.

In order to obtain an accurate enough suboptimal control law, we applied the proposed algorithm with the tolerance bounds $\epsilon = 8.1 \times 10^{-7}$. Simulation

results at different iteration times are summarized in Table 2. From Table 2, it is observed that convergence is achieved after 10 iterations, that is, $\left|\frac{J_{10}-J_9}{J_{10}}\right| = 8.040 \times 10^{-7} < 8.1 \times 10^{-7}$.

In Table 3, the minimums of J using the hybrid of block pulse and Legendre polynomials [21], hybrid of general block pulse and Legendre polynomials [36], orthogonal basis [16], hybrid of block pulse and orthogonal Taylor series [7], and present two methods are listed. Also the suboptimal control and state of the proposed ADM and VIM are demonstrated in Figure 2.

Table 2: Simulation results of Example 2 at different iteration times

method	ADM	ADM	VIM	VIM
N	J_N	$\left\|\frac{J_N - J_{N-1}}{J_N}\right\|$	J_N	$\left\|\frac{J_N - J_{N-1}}{J_N}\right\|$
6	0.36430512	–	0.35935108	–
7	0.36512980	2.258×10^{-2}	0.36431299	1.382×10^{-2}
8	0.37311212	2.139×10^{-2}	0.37511283	2.879×10^{-2}
9	0.37311294	2.519×10^{-6}	0.37311210	5.362×10^{-3}
10	0.37311291	8.040×10^{-7}	0.37311310	2.680×10^{-6}

Table 3: The cost functional values for Example 2

Method	Cost functional values
Marzban and Razzaghi [21]	0.37311241
Wang [36]	0.37312682
Kellat [16]	0.3731123
Dadkhah and Farahi [7]	0.373112935
Proposed method ($N = 10$)	
ADM	0.37311291
VIM	0.37311310

Example 3. Consider the following linear time-varying multi-delay systems [12, 18, 23]:

$$\begin{cases} \dot{x}_1(t) = x_2(t) + x_1(t-1), & t \geq 0, \\ \dot{x}_2(t) = tx_1(t) + 2x_1(t-1) + x_2(t-1) + u(t) - u(t-0.5), \\ x_1(t) = x_2(t) = 1, & -1 \leqslant t \leqslant 0, \\ u(t) = 5(t+1), & -0.5 \leqslant t \leqslant 0, \end{cases} \tag{39}$$

with the cost functional

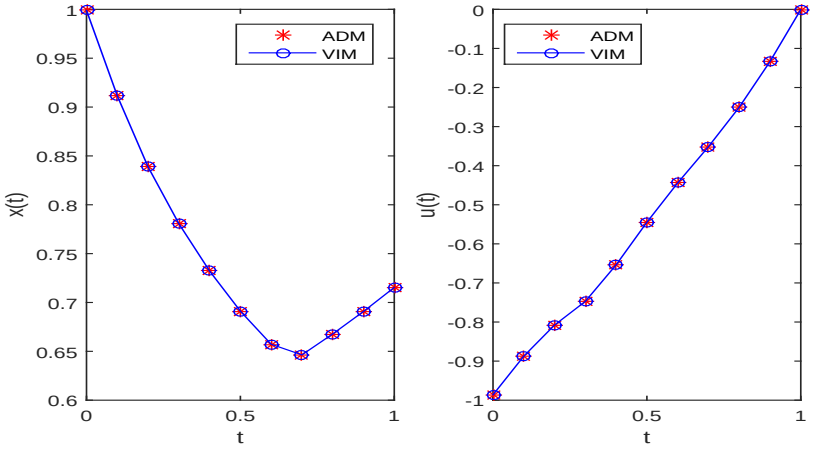


Figure 2: The suboptimal control and state, when $N = 10$ for Example 2.

$$J = \frac{1}{2}x_1^2(3) + x_2^2(3) + \frac{1}{2}\int_0^3 [2x_1^2(t) + 2x_1(t)x_2(t) + x_2^2(t) + \frac{u^2(t)}{(t+2)}]dt. \quad (40)$$

According to system (1), we have

$$A = \begin{pmatrix} 0 & 1 \\ t & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix},$$

and from (2), we get

$$Q_f = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad Q = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \quad R = 1/(t+2).$$

In order to obtain an accurate enough suboptimal control law, we applied the proposed algorithm with the tolerance bounds $\epsilon = 2.3 \times 10^{-6}$. Simulation results at different iteration times are summarized in Table 4. From Table 4, it is observed that convergence is achieved after 12 iterations, that is, $\left|\frac{J_{12} - J_{11}}{J_{12}}\right| = 2.270 \times 10^{-6} < 2.3 \times 10^{-6}$. Table 5, a comparison is made between the value of J obtained by the present method with $N = 12$, together with the value of J reported in the literature: Malek-Zavarei [18], by employing an iterative approach for determining the suboptimal control of the mentioned problem; the method of Hwang and Chen [12], by employing Legendre polynomials of order 20; and the method of Marzban and Pirmoradian [23], by employing a direct approach based on a hybrid of block-pulse functions and Lagrange interpolating polynomials.

Table 4: Simulation results of Example 3 at different iteration times

method	ADM	ADM	VIM	VIM
N	J_N	$\left\|\frac{J_N - J_{N-1}}{J_N}\right\|$	J_N	$\left\|\frac{J_N - J_{N-1}}{J_N}\right\|$
8	23.21056	–	22.17650	–
9	23.05198	6.879×10^{-2}	22.15431	1.001×10^{-2}
10	22.02108	1.681×10^{-3}	22.01430	5.903×10^{-3}
11	22.02230	5.539×10^{-5}	22.02143	3.237×10^{-4}
12	22.02235	2.270×10^{-6}	22.02115	1.271×10^{-5}

Table 5: The cost functional values for Example 3

Method	Cost functional values
Malek-Zavarei [18]	24.0200500
Hwang and Chen [12]	22.0212
Marzban and Pirmoradian [23]	22.0230201080
Proposed method ($N = 12$)	
ADM	22.02235
VIM	22.02115

Example 4. We now consider the following nonlinear time-varying multi-delay systems:

$$\begin{cases} \dot{x}(t) = x(t-1)x(t-2)u(t-2), & 0 \leqslant t \leqslant 3, \\ x(t) = 1, & -2 \leqslant t \leqslant 0, \\ u(t) = 0, & -2 \leqslant t \leqslant 0, \end{cases} \tag{41}$$

with the cost functional

$$J = \int_0^3 (x^2(t) + u^2(t))dt. \tag{42}$$

This optimal control is adopted from [10, 34, 35]. In order to obtain an accurate enough suboptimal control law, we applied the proposed algorithm with the tolerance bounds $\epsilon = 1.2 \times 10^{-6}$. Table 6, a comparison is made between the value of J obtained by the present method with $N = 13$, together with the value of J reported in the literature. The approximate solution of $x(t)$ and $u(t)$, obtained by the proposed method with $N = 13$ and the results of PMP method generated by Gollmann et al. [10] are plotted in Figure 3. Therefore, in view of the results, the present method is quite effective.

Table 6: The cost functional values for Example 4

Method	Cost functional values
Wachter et al. [35](600 grid points)	2.763044
Vanderbei [34](600 grid points)	2.763044
Gollmann et al. [10](600 grid points)	2.761594156
Proposed method ($N = 13$)	
ADM	2.761591012
VIM	2.761592238

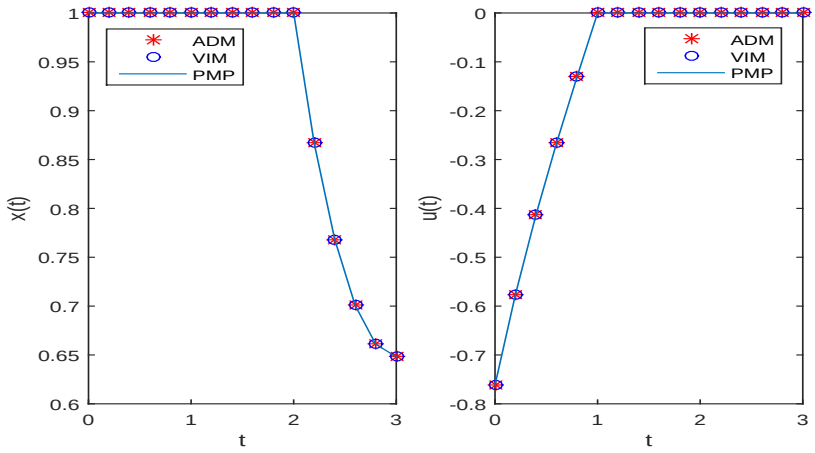


Figure 3: The suboptimal control and state, when $N = 13$ for Example 4.

6 Conclusion

In this work, the ADM has been successfully applied to find the solution of suboptimal control for linear time-varying systems with multiple state and control delays, and quadratic cost functional is presented. By using the ADM and VIM with the finite-step iteration of algorithm, we obtained a suboptimal control law. Some numerical examples have been provided to demonstrate the validity and applicability of the proposed method. The method is general and yields very accurate results.

Acknowledgements

Author is grateful to the anonymous referees and the editors for their constructive comments.

References

1. Adomian, G. *Nonlinear stochastic systems theory and applications to physics*, Kluwer Academic Publishers, 1989, Boston.

2. Adomian, G. *A review of the decomposition method in applied mathematics*, J. Math. Anal. Appl. 135(2) (1989), 501–544.

3. Alizadeh, A. and Effati, S. *Numerical schemes for fractional optimal control problems*, J. Dyn. Sys. Meas. Contr. 4 (2017), 1–14.

4. Balasubramaniam, P., Krishnasamy, R. and Rakkiyappan, R. *Delay-dependent stability of neutral systems with time-varying delays using delay-decomposition approach*, Appl. Math. Model. 36 (2012), 2253–2261.

5. Banks, H.T. and Burns, J.A. *Hereditary control problem: Numerical methods based on averaging approximations*, SIAM J. Contr. Optim. 16(2) (1978), 169–208.

6. Basin, M. and Rodriguez-Gonzalez, J. *Optimal control of linear systems with multiple time delays in control input*, IEEE Trans. Automat. Contr. 51(1) (2006), 91–97.

7. Dadkhah, M. and Farahi, M.H. *Optimal control of time delay systems via hybrid of block-pulse functions and orthogonal Taylor series*, Int. J. Appl. Comput. Math. 2(1) (2016), 137–152.

8. Edrisi-Tabriz, Y., Lakestani, M. and Heydari, A. *Two numerical methods for nonlinear constrained quadratic optimal control problems using linear B-spline functions*, Iranian J. Numer. Anal. Optim. 6(2) (2016), 17–37.

9. Gollmann, L., Kern, D. and Maurer, H. *Optimal control problems with delays in state and control variables subject to mixed control state constraints*, Optim. Contr. Appl. Meth. 30 (2009), 341–365.

10. Gollmann, L. and Maurer, H. *Theory and applications of optimal control problems with mul-tiple time delays*, J. Ind. Manag. Optim. 10(2) (2014), 413–441.

11. Haddadi, N., Ordokhani, Y. and Razzaghi, M.*Optimal control of delay systems by using a hybrid functions approximation*, J. Optim. Theor. Appl. 153 (2012), 338–356.

12. Hwang, G. and Chen, M.Y. *Suboptimal control of linear time-varying multi-delay systems via shifted Legendre polynomials*, Int. J. Syst. Sci. 16(12) (1985), 1517–1537.

13. Jajarmi, A., Dehghan-Nayyeri, M. and Saberi-Nik, H. *A novel feedforward-feedback suboptimal control of linear time-delay systems via shifted Legendre polynomials*, J. Complex. 35 (2016), 46–62.

14. Jamshidi, M. and Wang, C.M. *A computational algorithm for large-scale nonlinear time-delays systems*, IEEE Trans. Syst. Man. Cyber. SMC. 14 (1984), 2–9.

15. Kharatishvili, G.L. *The maximum principle in the theory of optimal process with time-lags*, Doklady Akademii Nauk SSSR. 136 (1961), 39–42.

16. Khellat, F. and Vasegh, N. *Suboptimal control of linear systems with delays in state and input by orthogonal basis*, Int. J. Comput. Math. 88(4) (2011), 781–794.

17. Koshkouei, A.J., Farahi, M.H. and Burnham, K.J. *An almost optimal control design method for nonlinear time-delay systems*, Int. J. Contr. 85(2) (2012), 147–158.

18. Malek-Zavarei, M. *Near-optimum design of nonstationary linear systems with state and control delays*, J. Optim. Theor. Appl. 30 (1980), 73–88.

19. Marzban, H.R. *Optimal control of linear multi-delay systems based on a multi-interval decomposition scheme*, Optim. Contr. Appl. Meth. 37(1) (2016), 190–211.

20. Marzban, H.R. and Hoseini, S.M. *An efficient discretization scheme for solving nonlinear optimal control problems with multiple time delays*, Optim. Contr. Appl. Meth. 37(4) (2016), 682–707.

21. Marzban, H.R. and Razzaghi, M. *Optimal control of linear delay systems via hybrid of block-pulse and Legendre polynomials*, J. Franklin Inst. 341 (2004), 279–293.

22. Marzban, H.R. and Pirmoradian, H. *A direct approach for the solution of nonlinear optimal control problems with multiple delays subject to mixed state-control constraints* , Appl. Math. Model. 53 (2018), 189–213.

23. Marzban, H.R. and Pirmoradian, H. *A novel approach for the numerical investigation of optimal control problems containing multiple delays*, Optim. Contr. Appl. Meth. 39(1) (2018), 302–325.

24. Mehne, H.H., and Farahi, M.H. *Transformation to a fixed domian in LP modelling for a class of optimal shape design problems*, Iranian J. Numer. Anal. Optim. 9(1) (2019), 1–16.

25. Mirhosseini-Alizamini, S.M. *Numerical solution of the controlled harmonic oscillator by homotopy perturbation method*, Contr. Optim. Appl Math. 2(1) (2017), 77–91.

26. Mirhosseini-Alizamini, S.M. and Effati, S. *An iterative method for suboptimal control of a class of nonlinear time-delayed systems*, Int. J. Contr. 92 (12) (2019), 2885–2869.

27. Mirhosseini-Alizamini, S.M., Effati, S. and Heydari, A. *An iterative method for suboptimal control of linear time-delayed systems*, Syst. Contr. Lett. 82 (2015), 40–50.

28. Mirhosseini-Alizamini, S.M., Effati, S. and Heydari, A. *Solution of linear time-varying multi-delay systems via variational iteration method*, J. Math. Comput. Sci. 16 (2016), 282–297.

29. Nazemi, A. and Mansoori, M. *Solving optimal control problems of the time-delayed systems by Haar wavelet*, J. Vib. Contr. 22(11) (2014), 2657–2670.

30. Nazemi, A. and Shabani, M.M. *Numerical solution of the time-delayed optimal control problems with hybrid functions*, IMA J. Math. Contr. Inform. 32(3) (2015), 623–638.

31. Richard, J.P. *Time-delay systems: An overview of some recent advances and open problems*, Automatica, 39 (2003), 1667–1694.

32. Saberi Nik, H., Rebelo, P. and Zahedi, S. *Solution of infinite horizon nonlinear optimal control problems by piecewise Adomian decomposition method*, Math. Model. Anal. 18(4) (2013), 543–560.

33. Shehata, M.M. *A study of some nonlinear partial differential equations by using Adomian decomposition method and variational iteration method*, Am. J. Comput. Math. 5 (2015), 195–203.

34. Vanderbei, R.J. and Shanno, D.F. *An interior-point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl. 13 (1999), 231–252.

35. Wachter, A. and Biegler, L.T. *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program. 106 (2006), 25–57.

36. Wang, X.T. *Numerical solutions of optimal control for linear time-varying systems with delays via hybrid functions*, J. Franklin Inst. 344 (2007), 941–953.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 185–191
DOI:10.22067/ijnao.v9i2.63018

Research Article

Fuzzy endpoint results for Ćirić-generalized quasicontractive fuzzy mappings

B. Mohammadi*

Abstract

We introduce Ćirić-generalized quasicontractive fuzzy mappings and provide the necessary and sufficient conditions of having a unique endpoint for such mappings. Then we introduce β-ψ-quasicontractive fuzzy mappings, establishing an endpoint result for them. Finally, we provide some results as an application.

AMS(2010): 47H10; 54H25.

Keywords: Fuzzy endpoint; Ćirić-generalized; Quasicontractive fuzzy mappings; Fuzzy approximate endpoint property.

1 Introduction and preliminaries

The concept of fuzzy set was introduced initially by Zadeh [12] in 1965. In 1981, Heilpern [6] established the fuzzy contraction and proved a fuzzy fixed point theorem, which was a generalization of Nadler's fixed point theorem for multi-valued mappings (see [9]). In 2001, Estruch and Vidal [5] utilized the result of Heilpern to fuzzy fixed point with fixed degree α for some $\alpha \in [0, 1]$, which was later generalized by many authors (see, for instance, [1,3,11]). Recently, Abbas and Turkoglu [11] proved the existence of a fuzzy fixed point for a generalized contractive fuzzy mapping. On the other hand, In 2010, Amini-Harandi [2] proved that some multi-valued mappings $T : X \to CB(X)$ have a unique endpoint if and only if they have the approximate endpoint property. Afterwards, considering the same properties, Moradi and Khojasteh [8] generalized Amini-Harandi's result. In this paper, in the sense of [8], we prove

*Corresponding author
Received 2 March 2017; revised 8 July 2019; accepted 17 July 2019
Babak Mohammadi
Department of Mathematics, Marand Branch, Islamic Azad University, Marand, Iran. e-mail: bmohammadi@marandiau.ac.ir

that some fuzzy mappings have a unique fuzzy endpoint if and only if they have the fuzzy approximate endpoint property.

Definition 1.(see [6]) Let X be a space of points with generic element x and $I = [0, 1]$. A fuzzy set in X is a function that associates any point of X with a number in interval $[0, 1]$. If A is a fuzzy set in X and $x \in X$, then $A(x)$ is called the grade of membership of x in A.

Definition 2.(see [6]) Let (X, d) be a metric space and let A be a fuzzy set in X. For $\alpha \in [0, 1]$, the α-level set of A denoted by $[A]_\alpha$, is defined as

$$[A]_\alpha = \{x|A(x) \geq \alpha\} \qquad if \qquad \alpha \in (0, 1]$$

and

$$[A]_0 = \overline{\{x|A(x) > 0\}},$$

where $\overline{B}$ denotes the closure of the nonfuzzy set B.

Definition 3.(see [6]) Let X be a nonempty set. For $x \in X$, we write $\mathbf{\{}x\mathbf{\}}$ the characteristic function of the ordinary subset $\{x\}$ of X. For $\alpha \in (0, 1]$, the fuzzy point x_α of X is the fuzzy set in X given by

$$x_\alpha(y) = \begin{cases} \alpha, & y = x, \\ 0, & y \neq x. \end{cases}$$

Define

$$W_\alpha(X) = \{C \in I^X : [C]_\alpha \textit{ is nonempty and compact}\}.$$

Throughout this paper, I^X denotes the collection of all fuzzy sets in X. For $A, B \in I^X$, it is called that A is more accurate than B (denoted by $A \subset B$) whenever $A(x) \leq B(x)$ for all $x \in X$. For $x \in X$, $S \subseteq X$, $A, B \in W_\alpha(X)$, and $\alpha \in (0, 1]$, we define

$$d(x, S) = \inf\{d(x, a) : a \in S\},$$

$$p_\alpha(x, A) = \inf\{d(x, a) : a \in [A]_\alpha\},$$

$$p_\alpha(A, B) = \inf\{d(a, b) : a \in [A]_\alpha, b \in [B]_\alpha\},$$

$$D_\alpha(A, B) = H([A]_\alpha, [B]_\alpha) = \max\{\sup_{x\in A} p_\alpha(x, B), \sup_{y\in B} p_\alpha(y, A)\},$$

where H is the Hausdorff distance. It is easily seen that D_α is the Hausdorff metric on $W_\alpha(X)$ induced by the metric d. Hereafter, we denote by $D_\alpha(x, A)$ the amount $D_\alpha(\mathbf{\{}x\mathbf{\}}, A) = H(\{x\}, [A]_\alpha)$ for all $x \in X$ and $A \in W_\alpha(X)$.

Definition 4.(see [5]) Let X be a nonempty set, let $T : X \to I^X$, and let $\alpha \in (0, 1]$. A fuzzy point x_α is called a fuzzy fixed point of T if $x_\alpha \subset Tx$ (or equally $x \in [Tx]_\alpha$). This means that the fixed degree of x is at least α. If $\mathbf{\{}x\mathbf{\}} \subset Tx$, then it is called that x is a fixed point of T.

2 Main results

Now, we are ready to state and prove the main results of this study. Firstly, we give the following definition:

Definition 5. Let X be a nonempty set, let $T : X \to I^X$, and let $\alpha \in (0, 1]$. We say that a point $x \in X$ is a fuzzy endpoint of T if $\{x\} = [Tx]_\alpha$. This means that x is the only point in X that the fixed degree of x is at least α. If $\{x\} = [Tx]_1$, we say that x is an endpoint of T.

Now, we give the following definition of fuzzy approximate endpoint property in the sense of Amini-Harandi [2].

Definition 6. Let (X, d) be a metric space, let $T : X \to I^X$, and let $\alpha \in (0, 1]$. We say that T has the fuzzy approximate endpoint property whenever

$$\inf_{x \in X} \sup_{y \in [Tx]_\alpha} d(x, y) = 0$$

or equally

$$\inf_{x \in X} D_\alpha(x, Tx) = 0.$$

Definition 7. Let (X, d) be a metric space, let $\alpha \in (0, 1]$, and let $T : X \to W_\alpha(X)$. We say that T is a Ćirić-generalized quasicontractive fuzzy mapping whenever there exists an upper semicontinuous (u.s.c) mapping $\psi : [0, +\infty) \to [0, +\infty)$ such that $\psi(t) < t$, for all $t > 0$ and $\liminf_{t \to \infty}(t - \psi(t)) > 0$ satisfying

$$D_\alpha(Tx, Ty) \leq \psi(M(x, y)) \quad \text{for all } x, y \in X, \tag{1}$$

where

$$M(x, y) = \max\{d(x, y), D_\alpha(x, Tx), D_\alpha(y, Ty), D_\alpha(x, Ty), D_\alpha(y, Tx)\}.$$

Theorem 1. *Let (X, d) be a complete metric space, let $\alpha \in (0, 1]$, and let $T : X \to W_\alpha(X)$ be a Ćirić-generalized quasicontractive fuzzy mapping. Then, T has a unique fuzzy endpoint if and only if T has the fuzzy approximate endpoint property.*

Proof. If T has a fuzzy endpoint, obviously, it has the fuzzy approximate endpoint property. Conversely, let T has the fuzzy approximate endpoint property. Then, there exists a sequence $\{x_n\}$ in X such that $\lim_{n \to \infty} D_\alpha(x_n, Tx_n) = 0$. Now for any $n, m \in \mathbb{N}$, we have

$$\begin{aligned} M(x_n, x_m) &= \max\{d(x_n, x_m), D_\alpha(x_n, Tx_n), \\ &\qquad D_\alpha(x_m, Tx_m), D_\alpha(x_n, Tx_m), D_\alpha(x_m, Tx_n)\} \\ &\leq D_\alpha(x_n, Tx_n) + D_\alpha(x_m, Tx_m) + D_\alpha(Tx_n, Tx_m) \\ &\leq D_\alpha(x_n, Tx_n) + D_\alpha(x_m, Tx_m) + \psi(M(x_n, x_m)). \end{aligned} \tag{2}$$

Therefore, from the above inequality, we have

$$\liminf_{n,m\to\infty} (M(x_n, x_m) - \psi(M(x_n, x_m))) = 0.$$

From the property of ψ, we can conclude that $\limsup_{n,m\to\infty} M(x_n, x_m) < \infty$. Thus from (2) and by upper semicontinuity of ψ, we have

$$\begin{aligned} \limsup_{n,m\to\infty} M(x_n, x_m) &\leq \limsup_{n,m\to\infty} \psi(M(x_n, x_m)) \\ &\leq \psi(\limsup_{n,m\to\infty} M(x_n, x_m)). \end{aligned}$$

So we have $\limsup_{n,m\to\infty} M(x_n, x_m) = 0$ and so $\{x_n\}$ is a Cauchy sequence. Since X is complete, there exists $x^* \in X$ such that $\lim_{n\to\infty} d(x_n, x^*) = 0$. We shall show that $\{x^*\} = [Tx^*]_\alpha$. To see this, we have

$$\begin{aligned} D_\alpha(x^*, Tx^*) &\leq d(x^*, x_n) + D_\alpha(x_n, Tx_n) + D_\alpha(Tx_n, Tx^*) \\ &\leq d(x^*, x_n) + D_\alpha(x_n, Tx_n) + \psi(M(x_n, x^*)). \end{aligned} \tag{3}$$

Limiting from both sides of (3), we get

$$D_\alpha(x^*, Tx^*) \leq \limsup_{n\to\infty} \psi(M(x_n, x^*)). \tag{4}$$

On the other hand,

$$\begin{aligned} M(x_n, x^*) &= \max\{d(x_n, x^*), D_\alpha(x_n, Tx_n), \\ &\qquad D_\alpha(x^*, Tx^*), D_\alpha(x_n, Tx^*), D_\alpha(x^*, Tx_n)\} \\ &\leq d(x_n, x^*) + D_\alpha(x_n, Tx_n) + D_\alpha(x^*, Tx^*), \end{aligned}$$

which implies

$$\limsup_{n\to\infty} M(x_n, x^*) \leq D_\alpha(x^*, Tx^*). \tag{5}$$

Consequently, from right upper semicontinuity of ψ, (4) and (5) yield

$$D_\alpha(x^*, Tx^*) \leq \psi(D_\alpha(x^*, Tx^*))$$

and so $H(\{x^*\}, [Tx^*]_\alpha) = D_\alpha(x^*, Tx^*) = 0$. This means that $\{x^*\} = [Tx^*]_\alpha$. The uniqueness of endpoint is concluded from (1). □

Definition 8. Let (X, d) be a metric space, $\alpha \in (0, 1]$, and $T : X \to W_\alpha(X)$. We say that T is a Ćirić-generalized β-ψ-quasicontractive fuzzy mapping whenever there exists an upper semicontinuous (u.s.c) mapping ψ :

$[0,+\infty) \to [0,+\infty)$ such that $\psi(t) < t$, for all $t > 0$ and $\liminf_{t\to\infty}(t-\psi(t)) > 0$ and a function $\beta : X \times X \to [0,\infty)$ satisfying

$$\beta(x,y)D_\alpha(Tx,Ty) \le \psi(M(x,y)) \quad \text{for all } x,y \in X, \tag{6}$$

where

$$M(x,y) = \max\{d(x,y), D_\alpha(x,Tx), D_\alpha(y,Ty), D_\alpha(x,Ty), D_\alpha(y,Tx)\}.$$

Theorem 2. *Let (X,d) be a complete metric space, let $\alpha \in (0,1]$, and let $T : X \to W_\alpha(X)$ be a Ćirić-generalized β-ψ-quasicontractive fuzzy mapping. Moreover suppose that*

(i) there exists a sequence $\{x_n\}$ in X such that $\beta(x_n,x_m) \ge 1$ for all $n,m \in \mathbb{N}$ with $n < m$ and $\lim_{n\to\infty} D_\alpha(x_n,Tx_n) = 0$,

(ii) for any sequence $\{x_n\}$ in X which $\beta(x_n,x_m) \ge 1$ for all $n,m \in \mathbb{N}$ with $n < m$ and $x_n \to x$, we have $\beta(x_n,x) \ge 1$, for all $n \in \mathbb{N}$.

Then, T has a fuzzy endpoint.

Proof. For any $n,m \in \mathbb{N}$, we have

$$\begin{aligned} M(x_n,x_m) &= \max\{d(x_n,x_m), D_\alpha(x_n,Tx_n), \\ &\qquad D_\alpha(x_m,Tx_m), D_\alpha(x_n,Tx_m), D_\alpha(x_m,Tx_n)\} \\ &\le D_\alpha(x_n,Tx_n) + D_\alpha(x_m,Tx_m) + \beta(x_n,x_m)D_\alpha(Tx_n,Tx_m) \\ &\le D_\alpha(x_n,Tx_n) + D_\alpha(x_m,Tx_m) + \psi(M(x_n,x_m)). \end{aligned} \tag{7}$$

Similar to Theorem 1, we conclude that $\limsup_{n,m\to\infty} M(x_n,x_m) = 0$ and so $\{x_n\}$ is a Cauchy sequence. Let $\lim_{n\to\infty} d(x_n,x^*) = 0$. We show that $\{x^*\} = [Tx^*]_\alpha$. To see this, we have

$$\begin{aligned} D_\alpha(x^*,Tx^*) &\le d(x^*,x_n) + D_\alpha(x_n,Tx_n) + \beta(x_n,x^*)D_\alpha(Tx_n,Tx^*) \\ &\le d(x^*,x_n) + D_\alpha(x_n,Tx_n) + \psi(M(x_n,x^*)). \end{aligned} \tag{8}$$

Consequently, as in Theorem 1, we obtain

$$D_\alpha(x^*,Tx^*) \le \psi(D_\alpha(x^*,Tx^*)),$$

which implies $H(\{x^*\},[Tx^*]_\alpha) = D_\alpha(x^*,Tx^*) = 0$. This means that $\{x^*\} = [Tx^*]_\alpha$. □

Let $\subset$ be the partial order on $W_\alpha(X)$ defined by $A \subset B$ if and only if $A(x) \le B(x)$ for all $x \in X$. In the following result, we restrict the contraction condition only for $x,y \in X$ with $Tx \subset Ty$.

Corollary 1. *Let (X,d) be a complete metric space, $\alpha \in (0,1]$, and $T : X \to W_\alpha(X)$ be a fuzzy mapping such that there exists an upper semicontinuous (u.s.c) mapping $\psi : [0,+\infty) \to [0,+\infty)$ with $\psi(t) < t$, for all $t > 0$ and $\liminf_{t\to\infty}(t - \psi(t)) > 0$ satisfying*

$$D_\alpha(Tx,Ty) \le \psi(M(x,y)) \quad \text{for all } x,y \in X \text{ with } Tx \subset Ty, \tag{9}$$

where

$$M(x,y) = \max\{d(x,y), D_\alpha(x,Tx), D_\alpha(y,Ty), D_\alpha(x,Ty), D_\alpha(y,Tx)\}.$$

Moreover suppose that

(i) there exists a sequence $\{x_n\}$ in X such that $\{Tx_n\}$ is a nondecreasing sequence in $W_\alpha(X)$ and $\lim_{n\to\infty} D_\alpha(x_n,Tx_n) = 0$,

(ii) for any sequence $\{x_n\}$ in X which $\{Tx_n\}$ is a nondecreasing sequence in $W_\alpha(X)$ and $x_n \to x$, we have $Tx_n \subset Tx$, for all $n \in \mathbb{N}$.

Then, T has a fuzzy endpoint.

Proof. Define the mapping $\beta : X \times X \to [0,\infty)$ by $\beta(x,y) = 1$, whenever $Tx \subset Ty$ and $\beta(x,y) = 0$ otherwise. Then apply Theorem 2. □

Corollary 2. *Let (X,d) be a complete metric space, let $x^* \in X$ be a fixed element, let $\alpha \in (0,1]$, and let $T : X \to W_\alpha(X)$ be a fuzzy mapping such that there exists an upper semicontinuous (u.s.c) mapping $\psi : [0,+\infty) \to [0,+\infty)$ with $\psi(t) < t$, for all $t > 0$ and $\liminf_{t\to\infty}(t - \psi(t)) > 0$ satisfying*

$$D_\alpha(Tx,Ty) \le \psi(M(x,y)) \quad \text{for all } x,y \in X \text{ with } Tx(x^*) = Ty(x^*), \tag{10}$$

where

$$M(x,y) = \max\{d(x,y), D_\alpha(x,Tx), D_\alpha(y,Ty), D_\alpha(x,Ty), D_\alpha(y,Tx)\}.$$

Moreover suppose that

(i) there are a sequence $\{x_n\}$ in X and $\lambda \in [0,1]$ such that $Tx_n(x^*) = \lambda$ is fixed for all $n \in \mathbb{N}$ and $\lim_{n\to\infty} D_\alpha(x_n,Tx_n) = 0$,

(ii) for any sequence $\{x_n\}$ in X that $Tx_n(x^*) = \lambda$ is fixed for all $n \in \mathbb{N}$ and $x_n \to x$, we have $Tx(x^*) = \lambda$, for all $n \in \mathbb{N}$.

Then, T has a fuzzy endpoint.

Proof. Define the mapping $\beta : X \times X \to [0,\infty)$ by $\beta(x,y) = 1$, whenever $Tx(x^*) = Ty(x^*)$ and $\beta(x,y) = 0$ otherwise. Then applying Theorem 2 completes the proof. □

Acknowledgement

This study was supported by Marand Branch, Islamic Azad University, Marand, Iran.

References

1. Abbas, M., Turkoglu, D. *Fixed point theorem for a generalized contractive fuzzy mapping*, J. Intell. Fuzzy Systems. 26 (2014), 33–36.

2. Amini-Harandi, A. *Endpoints of set-valued contractions in metric spaces*, Nonlinear Anal. 72 (2010), 132–134.

3. Azam, A., Beg, I. *Common fixed points of fuzzy maps*, Math. Comput. Modelling. 49 (2009), 1331–1336.

4. Ćirić, L.B. *A generalization of Banach's contraction principle*, Proc. Amer. Math. Soc. 45 (1974), 267–273.

5. Estruch, V.D., Vidal, A. *A note on fixed fuzzy points for fuzzy mappings*, Rend Istit. Mat. Univ. Trieste. 32 (2001), 39-45.

6. Heilpern, S. *Fuzzy mappings and fixed point theorems*, J. Math. Anal. Appl. 83 (1981), 566–569.

7. Mohammadi, B., Rezapour, SH., Shahzad, N. *Some results on fixed points of α-ψ-Ciric generalized multifunctions*, Fixed Point Theory Appl. (2013):24.

8. Moradi, S., Khojasteh, F. *Endpoints of multi-valued generalized weak contraction mappings*, Nonlinear Anal. 74 (2011), 2170–2174.

9. Nadler, S.B. *Multi-valued contraction mappings*, Pacific J. Math. 30 (1969), 475–488.

10. Samet, B., Vetro, C., Vetro, P. *Fixed point theorems for α-ψ-contractive type mappings*, Nonlinear Anal. 75 (2012), 2154–2165.

11. Turkoglu, D., Rhoades, B.E. *A fixed fuzzy point for fuzzy mapping in complete metric spaces*, Math. Commun. 10 (2005), 115–121.

12. Zadeh, L.A. *Fuzzy sets*, Inf. Control. 8 (1965), 103–112.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 193–205
DOI:10.22067/ijnao.v9i2.81034

Research Article

Multicriteria allocation model of additional resources based on DEA: MOILP-DEA

J.W. Yougbare

Abstract

We exploit the relationship between multiobjective integer linear problem (MOILP) and data envelopment analysis (DEA) to develop an approach to a resource reallocation problem. The general purpose of the mathematical formulation of this multicriteria allocation model based on DEA is to enable decision-makers to take into account the efficiency of units under control to allocate additional resources for a new period of operation. We develop a formal approach based on DEA and MOILP to find the most preferred allocation plan taken account additional resources. The mathematical model is given, and we illustrate it with a numerical example.

AMS(2010): 90B99.

Keywords: Multicriteria decision aiding; Data envelopment analysis; Multiobjective integer linear programming; Additional resource allocation; Efficient frontier.

1 Introduction

The use of the data envelopment analysis (DEA) approach [1, 4] provides useful information for monitoring and management in a production organization. The essential elements are knowledge reference units, sources of inefficient units, variations of productivity from one period to the next, and variable returns to scale.

Multicriteria decision aiding (MCDA) and DEA are two techniques in operations research, which attract the attention of many researchers. Some

*Corresponding author
Received 30 May 2019; revised 2 June 2019; accepted 3 August 2019
Jacob Wendpanga Yougbare
Department of Mathematics, Faculty of Sciences and Technologies, University Norbert ZONGO, Koudougou, Burkina Faso. e-mail: jwyougby@yahoo.com, ORCID : 0000-0001-8611-337X

authors are interested in their relationship and how to exploit [2, 3, 7–9, 11, 13, 15, 16]. We propose a mathematical model of multiobjective optimization based on DEA that can help managers or decision-makers adjust investments in different units of the control system by additional assignments to bring.

The general purpose of the mathematical formulation of this multicriteria allocation model based on DEA is to enable decision-makers to take into account the performance of units under control to allocate additional resources for a new period of operation. So, it is necessary to determine information intermediaries:
– information on production or performance efficiencies;
– information on the returns to scale presented by each unit;
– optimal targets for each production unit.

The mathematical model, we propose, takes into account all this information intermediaries. The objective is, on the basis of additional quantity of resources and demands expressed by the different DMUs, to allocate optimally and efficiently these resources available to maximize estimates of product returns based on information gathered from performance measures and returns of the ladder. In a given production system, the reference units (the efficient units) are those that use resources (inputs) rationally, at best, to achieve optimal results. This transformation of inputs into outputs is supposed to follow a function the best of which is that used by efficient units. These efficient units give an idea of best practices and thus provide the best possible production plans. On the basis of information on the units, namely the identification of the efficient and the inefficient, it is important to consider how this information can be used to assist the decision-maker in allocating resources for a new production period with the assumption that the system will use the same process of transformation. One of the information that can be provided by the application of the DEA method is the identification of the variable returns to scale associated with each unit of the system on a production function basis. Efficient and inefficient units can be either of increasing returns to scale, constant returns to scale, or decreasing returns to scale.

In this paper, we present the modeling of the problem, the mathematical model and illustrate with a numerical example.

2 Issue and literature review

The DEA methodology is one of the most widely used decision aiding approaches in the literature. Emrouznejad and Yang (2017) [5] listed more than 10,300 published articles. The fields of application are very diverse and broad. “Resource allocation is the generic problem of assigning available resources to users in the best possible way. Usually the resources are limited,

thus sets of activities compete for these resources establishing explicit and implicit dependencies, which may be subject to uncertainty. The resulting decision problem on the best use of the resources can be decomposed in several problems that have been intensively studied in the field of operations research in the last few decades. Examples of such problems are [1]:
Scheduling problem. Scheduling is used to optimally allocate scarce resources to activities and includes problems like the allocation of plant and machinery resources, the organization of human resources, and the control of production processes and materials purchase.
Knapsack problem. The main idea is to consider the capacity of the knapsack as the available amount of resource and the item types as activities to which this resource can be allocated.
Cutting stock problem. It is a general resource allocation problem, where the objective is to cut pieces of stock material into pieces of specified sizes, minimizing the wasted material.
Power generation scheduling problem. Power generation scheduling is required in order to find the optimum allocation of energy such that the annual operating cost of a power system is minimized, or the obtained profit is maximized."

There is a large literature dealing with these problems and several solving techniques have been proposed. Several authors have been interested in the problem of resource allocation. Fang, [6] used a generalized DEA model for centralized resource allocation to uncover the sources of such total input contraction. Wu et al. [14] integrated DEA and MOLP to deal with the resource allocation problem. Several authors have been interested in the linkage of the two methods, the DEA method coupled with the MOILP method (rkc et al. [10] and Sueyoshi [12]). We looked at questions such as: how can we take into account the performance indices of the different units of a production system to reallocate additional resources. To answer this question, we have made certain assumptions. First, the observed efficient production boundary of the system provides the best production function of the moment. Second, the lower expected returns to scale of a unit is an indicator to minimize the risk of resource waste. Indeed, when a unit has a variable returns to scale, the methods give a forecast of the expected yield. This random aspect led us to focus on the lowest expected return. In developing countries, the problem of allocating additional resources concerns more than one. Donors and technical and financial partners, not only do they want to have measurable performance indicators, but they also want to know if an efficient unit is not going to lose performance if we add too many resources to it. With respect to inefficient units, the same questions arise. Our approach is to address these two types of concerns: to take into account the performance indices calculated by DEA and to maximize the expected returns to scale of the units taking into account the risks of waste. For example, if the minimum return to scale

[1] Dimitri Thomopulos, Models and Solutions of Resource Allocation Problems based on Integer Linear and Nonlinear Programming

is zero, then the risk of waste is high. As a precautionary measure, we have to wait for another evaluation period before adding additional resources.

3 Modeling of the problem

The Modeling of the problem requires intermediate steps to calculate, on the one hand, coefficients of improvement of efficiency, according to the increase inputs, and on the other hand, minimum values of variable returns to scale of the different units.

3.1 Determination of reference units and estimation of the "technical" function for improving returns

One of the major information in the process of modeling the problem is knowledge of reference units that are technically effective units of system. These units are determined using the Charnes, Cooper and Rhodes (CCR) model [4]. We note

$$\mathcal{E} = \{j \in \{1, \ldots, N_E\} \text{such as Decision Making Unit (DMU)} : j \text{ is CCR efficient}\},$$

which contains the technical efficient DMUs. To obtain $\mathcal{E}$, we use the DEA techniques [1,4]. Once the set $\mathcal{E}$ is determined, we use the Banker, Charnes and Cooper (BCC) model [1] with input orientation analyzing the technical efficient units by taking turns, individually each output factor associated with all input factors. This step consists of measuring the performance of technical efficient units relative to each output factor obtained with input factors. It is a question of measuring the performance of the N_E units in $\mathcal{E}$ with a single output k and the m input factors. Thus, we estimate the possible maximum efficiency for each output factor in relation to the observed data. This mathematically translates into resolution for each unit $d \in \mathcal{E}$ and for each output factor $k \in \{1, \ldots, s\}$ the following problem.

$$\max ft(k,d) = \mu_k^d y_{kd} + u^{k,d}$$
$$\text{subject to the constraints (s.c)} \begin{cases} \mu_k^d y_{kj} - \sum_{i=1}^{m} \nu_i^{k,d} x_{ij} + u^{k,d} \leq 0, \quad j = 1, \ldots, n, \\ \sum_{i=1}^{m} \nu_i^{k,d} x_{id} = 1, \\ \mu_k^d \geq \varepsilon, \quad \nu_i^{k,d} \geq \varepsilon, \quad \text{for all } i, u^{k,d} \in \mathbb{R}, \end{cases} \tag{1}$$

Remark: Although the units under evaluation are all technically effective, therefore using the full BCC model, they are not necessarily efficient here

since only one output factor is considered and the unit may be less efficient for this single factor. In other words, the optimal value simply checks $\tilde{ft}(k,d) \leq 1$.

To avoid problems related to situations of multiple optimal solutions, we use the following min max model (2).

$$\min \Delta_k^d \quad \text{s.c.} \begin{cases} \mu_k^d(y_{kj} - y_{kd}) - \sum_{i=1}^{m} \nu_i^{k,d}(x_{ij} - x_{id}) - \delta_j^{k,d} \leq 0, j = 1, \ldots, n, \\ \sum_{i=1}^{m} \nu_i^{k,d} x_{id} = 1, \\ \Delta_k^d - \delta_j^{k,d} \geq 0, \quad j = 1, \ldots, n, \\ \mu_k^d \geq \varepsilon, \quad \nu_i^{k,d} \geq \varepsilon, \qquad i = 1, \ldots, m. \end{cases} \tag{2}$$

The above formulation indicates that for a given output factor k, the hyperplane representing the maximum efficiency of that factor with the reference unit d is defined by

$$\tilde{\mu}_k^d(y_k - y_{kd}) - \sum_{i=1}^{m} \tilde{\nu}_i^{k,d}(x_i - x_{id}) = \tilde{\Delta}_k^d.$$

It can therefore be seen that, for the output factor k and for the DMU d, $\tilde{\mu}_k^d(y_k - y_{kd})$ is provided by poor performance

$$\sum_{i=1}^{m} \tilde{\nu}_i^{k,d}(x_i - x_{id}) + \tilde{\Delta}_k^d.$$

The contribution of the various input factors to this increase can therefore be measured by the expression $\sum_{i=1}^{m} \tilde{\nu}_i^{k,d} x_i$, where x_i represents the increase in factor input i. In order to take into account all the reference units $d \in \mathcal{E}$, we consider the average of these values by introducing the quantity

$$\sum_{i=1}^{m} \alpha_i^k x_i, \quad \text{where} \quad \alpha_i^k = \frac{1}{N_E} \sum_{d=1}^{N_E} \tilde{\nu}_i^{k,d}. \tag{3}$$

The coefficients $\alpha_i^k \geq 0$ reflect the relationship between the increase of the input factor i and the maximum average return to which it contributes for the improvement of the output k.

3.2 Determination of returns to scale values

However, it is also necessary to take into account the information on the variable returns to scale of the different units in order to allocate the additional

amount of available input optimally. In order to do this, we determine the minimum values of the returns to scale that each unit can present. These values are calculated for projections obtained on the efficient frontier, that is, if the unit is not efficient, we project it on the efficient border before calculating the minimum value of return to scale it can present. We use the input-oriented BCC model to determine the values of the projections and the minimum values of the returns to scale. We note $\widehat{X}_j$ and $\widehat{Y}_j$ the vectors of the inputs and outputs, respectively, corresponding to the projection of unit j on the efficient frontier. For each unit d, $\quad d = 1, \ldots, n$, we calculate $\rho^{\min,d}$ the finite minimum value of the present returns to scale by solving the following problem:

$$u^{\min,d} = \min\ u^d$$
$$\text{s.c.} \quad \begin{cases} \sum\limits_{r=1}^{s} \mu_r^d y_{rj} - \sum\limits_{i=1}^{m} \nu_i^d x_{ij} + u^d \leq 0, \quad j = 1, \ldots, n, \quad j \neq d, \\ \sum\limits_{r=1}^{s} \mu_r^d \widehat{y}_{rd} + u^d = 1, \\ \mu_r^d \geq 0, \quad \nu_i^d \geq 0, \quad \text{for all} r, i, \quad u^d \in \mathbb{R}, \end{cases} \tag{4}$$

According to Banker and Thrall [1],

$$\rho^{\min,d} = \frac{1}{1 - u^{\min,d}} \geq 0.$$

Note $\gamma_j = \rho^{\min,j}, \quad j = 1, \ldots, n$. We note that the objective is to determine the smallest theoretically observable scale performance.

4 Mathematical model

We now describe the formulation of the model, starting with the constraints and then the objective functions.

4.1 Constraints on resource availability

We consider the following:
–that for each input factor i, a b_i quantity is available to be distributed between the different units j.
–that each DMU j expresses a request a_{ij} units for input i (we assume $a_{ij} \leq b_i, \quad$ for all j.) by introducing binary variables $t_{i,j}$ by $t_{i,j} = 1$, if the request for input i of the DMU j is retained and $t_{i,j} = 0$, otherwise. The constraints of the MOILP-DEA model are then written

$$\sum_{j=1}^{n} a_{ij} t_{i,j} \leq b_i, \quad i = 1, \ldots, m. \tag{5}$$

4.2 Criteria

In the multicriteria model, we introduce s objective functions $z_k, k = 1, \ldots, s$ corresponding to the s output factors. z_k will translate the additional return (expected) of the factor k due to the increase of inputs i granted to the different DMUs j. In order to calculate this increase it is necessary to take into account both the coefficients α_i^k determined in 3.1 and the coefficients γ_j determined in 3.2.
Also the z_k function is written

$$\begin{aligned} &z_k = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}^k t_{i,j}, \\ &\text{where} \quad c_{ij}^k = \alpha_i^k \gamma_j a_{ij}. \end{aligned} \tag{6}$$

4.3 MOILP-DEA

The multicriteria problem that we propose to allocate additional resources to DMUs is thus finally formulated by

$$\begin{aligned} &\text{“max”} \; z_k = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}^k t_{i,j}, \quad k = 1, \ldots, s \\ &\text{s.c.} \quad \begin{cases} \sum_{j=1}^{n} a_{ij} t_{i,j} \leq b_i, \quad i = 1, \ldots, m, \\ t_{i,j} \in \{0, 1\} \quad \text{for all } i, \quad \text{for all } j. \end{cases} \end{aligned} \tag{7}$$

It should be noted that, since no interaction is involved in the constraints between the variables relating to two different inputs, the problem can be broken down into separate problems, considering each input separately.
For a fixed input i, the problem can be written

$$\begin{aligned} &\text{“max”} \; z_{k,i} = \sum_{j=1}^{n} c_{ij}^k t_{i,j}, \quad k = 1, \ldots, s \\ &\text{s.c.} \quad \begin{cases} \sum_{j=1}^{n} a_{ij} t_{i,j} \leq b_i, \\ t_{i,j} \in \{0, 1\} \quad \text{for all } j. \end{cases} \end{aligned} \tag{8}$$

These m problems are each presented as a multicriteria "Knapsack problem" for which many methods of resolution exist. We have of course

$$z_k = \sum_{i=1}^{m} z_{k,i}.$$

5 Illustration

We consider the illustrative example with 3 input and 3 output factors, the data of which are presented in Table 1.

Table 1: Data and Results of the Example

DMU	Input 1	Input 2	Input 3	Output 1	Output 2	Output 3	score CCR	$\rho^{\min,d}$
1	8	4	6	5	4	8	1.0000	0.6748
2	7	8	4	3	7	4	0.9960	0.6772
3	11	6	4	11	11	5	1.0000	0.3423
4	10	10	2	9	14	1	1.0000	0.0732
5	2	7	3	4	1	4	1.0000	0.9582
6	6	10	10	8	10	13	1.0000	0.0000
7	11	7	11	11	2	16	1.0000	0.0000
8	5	14	8	6	13	9	0.8885	0.2899
9	10	11	9	9	5	12	0.9101	0.8969
10	1	8	2	9	13	2	1.0000	0.0388
11	6	5	3	12	5	5	1.0000	0.1574
12	5	9	1	10	8	1	1.0000	0.0845
13	9	12	8	14	17	10	1.0000	0.0000
14	13	13	7	15	12	9	0.8099	0.0000

Using the CCR model, we get all the technically efficient DMUs

$$\mathcal{E} = \{1, \ldots, 14\} \setminus \{2, 8, 9, 14\}.$$

Using the approach proposed by Banker and Thrall in [1] (see problem (4) above), we obtain the minimum values of the variable returns to scale $\rho^{\min,d}$ of the 14 DMUs considered (see the last column of Table 1). To analyze each $d \in \mathcal{E}$ and each output factor k, we use the problem (2) and we obtain the following results (see Table 2). The results in Table 3 allow the calculation of the coefficients α_i^k.

Let us consider that the requests a_{ij} expressed in units i for the different DMUs j are those shown in Table 4.

Table 2: Results

DMU d	$\tilde{\Delta}_1^d$	$\tilde{\Delta}_2^d$	$\tilde{\Delta}_3^d$	$\tilde{\mu}_1^d$	$\tilde{\mu}_2^d$	$\tilde{\mu}_3^d$	$\tilde{\nu}_1^{1,d}$	$\tilde{\nu}_1^{2,d}$	$\tilde{\nu}_1^{3,d}$	$\tilde{\nu}_2^{1,d}$	$\tilde{\nu}_2^{2,d}$	$\tilde{\nu}_2^{3,d}$	$\tilde{\nu}_3^{1,d}$	$\tilde{\nu}_3^{2,d}$	$\tilde{\nu}_3^{3,d}$
1	0.000	0.000	0.000	0.001	0.001	0.001	0.060	0.061	0.062	0.129	0.126	0.124	0.001	0.001	0.001
3	0.185	0.000	0.185	0.001	0.030	0.001	0.001	0.001	0.001	0.137	0.141	0.135	0.043	0.036	0.045
4	0.218	0.000	0.218	0.001	0.182	0.001	0.001	0.001	0.001	0.071	0.087	0.071	0.142	0.062	0.142
5	0.000	0.000	0.000	0.001	0.001	0.001	0.056	0.056	0.053	0.107	0.109	0.105	0.047	0.042	0.054
6	0.385	0.345	0.000	0.005	0.031	0.050	0.046	0.012	0.065	0.072	0.092	0.060	0.001	0.001	0.001
7	0.312	0.365	0.000	0.019	0.001	0.148	0.001	0.038	0.089	0.140	0.081	0.001	0.001	0.001	0.001
10	0.000	0.000	0.000	0.001	0.090	0.001	0.019	0.010	0.112	0.083	0.001	0.111	0.158	0.491	0.001
11	0.000	0.000	0.000	0.001	0.001	0.001	0.001	0.001	0.001	0.090	0.091	0.090	0.181	0.180	0.181
12	0.000	0.000	0.000	0.001	0.001	0.001	0.001	0.017	0.001	0.001	0.084	0.001	0.986	0.158	0.986
13	0.000	0.000	0.157	0.238	0.220	0.082	0.048	0.109	0.021	0.047	0.001	0.001	0.001	0.001	0.100

Table 3: Results of the coefficients α_i^k

k	α_1^k	α_2^k	α_3^k
1	0.0233	0.0876	0.1560
2	0.0306	0.0812	0.0973
3	0.0406	0.0698	0.1513

Table 4: Quantities requested a_{ij}

DMU	1	2	3	4	5	6	7	8	9	10	11	12	13	14	b_i
Inputs i															
1	1	2	0	3	0	1	1	0	2	1	1	0	2	0	10
2	1	1	1	0	1	1	0	1	1	1	0	2	0	1	5
3	2	0	0	1	2	1	1	1	0	0	2	1	1	2	8

With this example, the MOILP-DEA model is equivalent to solving the 3 following multicriteria "knapsack problems".

$$\text{“max”} \begin{cases} \mathbf{z_{1,1}} = \alpha_1^1 \sum_{j=1}^{14} a_{1j}\gamma_j t_{1,j} \\ \mathbf{z_{2,1}} = \alpha_1^2 \sum_{j=1}^{14} a_{1j}\gamma_j t_{1,j} \\ \mathbf{z_{3,1}} = \alpha_1^3 \sum_{j=1}^{14} a_{1j}\gamma_j t_{1,j} \end{cases}$$
$$\text{s.c.} \begin{cases} \sum_{j=1}^{14} a_{1,j} t_{1j} \leq 10, \\ t_{1,j} \in \{0,1\} \quad j = 1, \ldots, 14. \end{cases}$$

$$\text{“max”} \begin{cases} \mathbf{z_{1,2}} = \alpha_2^1 \sum_{j=1}^{14} a_{2j}\gamma_j t_{2,j} \\ \mathbf{z_{2,2}} = \alpha_2^2 \sum_{j=1}^{14} a_{2j}\gamma_j t_{2,j} \\ \mathbf{z_{3,2}} = \alpha_2^3 \sum_{j=1}^{14} a_{2j}\gamma_j t_{2,j} \end{cases}$$
$$\text{s.c.} \begin{cases} \sum_{j=1}^{14} a_{2j} t_{2,j} \leq 5, \\ t_{2,j} \in \{0,1\} \quad j = 1, \ldots, 14. \end{cases}$$

$$\text{“max”} \begin{cases} \mathbf{z_{1,3}} = \alpha_3^1 \sum_{j=1}^{14} a_{3j}\gamma_j t_{3,j} \\ \mathbf{z_{2,3}} = \alpha_3^2 \sum_{j=1}^{14} a_{3j}\gamma_j t_{3,j} \\ \mathbf{z_{3,3}} = \alpha_3^3 \sum_{j=1}^{14} a_{3j}\gamma_j t_{3,j} \end{cases}$$

$$\text{s.c.} \begin{cases} \sum_{j=1}^{14} a_{3j} t_{3,j} \leq 8, \\ t_{3,j} \in \{0,1\} \quad j = 1, \ldots, 14. \end{cases}$$

Individual maximizations of the 3 criteria give:
– $k = 1, t_{1,1} = t_{1,2} = t_{1,4} = t_{1,9} = t_{1,10} = t_{1,11} = 1$, and $t_{1,j} = 0$ otherwise
– $k = 2, t_{2,1} = t_{2,2} = t_{2,3} = t_{2,5} = t_{2,9} = 1$, and $t_{2,j} = 0$ otherwise
– $k = 1, t_{3,1} = t_{3,5} = t_{3,8} = t_{3,11} = 1$, and $t_{3,j} = 0$ otherwise.

6 Conclusion

In this article, using the relationship between DEA efficiency and MOILP, we have proposed a mathematical model, a multicriteria linear problem for additional resources allocation, and we have solved an illustrative sample. The formulation described here can provide an efficient framework for optimizing investment taking into account analyses of the performance efficiency of a production system. On the one hand, the determination of the units of reference can facilitate the planning of the development programs of a given unit (efficient or inefficient); on the other hand, the minimum values of variable returns to scale make it possible to analyze the minimum need for additional input factors to improve returns. This different information provided by the use of DEA makes it possible to efficiently allocate additional resources so, to search and performance efficiency, and stability of efficiency of the units of the system. We then hope, for validation purposes, to apply these models to specific cases relating to public sectors such as education. In the case of aid and financing projects for farmers, it is important to note that this is the case in the case of health and a private sector such as agriculture. This would then allow validation of this initial model proposal and could be re-designed if needed.

Acknowledgements

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

References

1. Banker R.D. and Thrall R.M. *Estimation of returns to scale using data envelopment analysis*, Eur. J. Oper. Res. 62 (1992), 74–84.

2. Belton V. , Vickers S. *Demystifying DEA- a visual interactive approach based on multiple criteria analysis*, J. Oper. Res. Soc. 44 (1993), 883–896.

3. Bouyssou D. *Using DEA as a tool for MCDM: some remarks*, J. Oper. Res. Soc. 50 (1999), 974–978.

4. Charnes A., Cooper W.W. and Rhodes E. *Measuring the efficiency of decision making units*, Eur. J. Oper. Res. 2 (1978), 429–444.

5. Emrouznejad A. and Yang G., *A survey and analysis of the first 40 years of scholarly literature in DEA: 19782016*, Socio-Econ. Plan. Sci. 61 (2018), 4–8.

6. Fang L., *A generalized DEA model for centralized resource allocation*, Eur. J. Oper. Res. 228(2) (2013), 405–412.

7. Jahanshahloo G. R., Lotfi F. H., Shoja N. and Tohidi G. *A method for finding efficient DMUs in DEA using 0-1 linear programming*, Appl. Math. Comput. 159 (2004), 37–45.

8. Joro T., Korhonen P. and Wallenius J. *Structural comparison of data envelopment analysis and multiple objective linear programming*, Management Science. 44 (1998), no. 7, 962–970.

9. Korhonen, K. and Syrjnen, M. *Resource allocation based on efficiency analysis*, Management Science. 50 (2004) (8),1134–1144.

10. rkc H., nsal M.G. and Bal H. *A modification of a mixed integer linear programming (MILP) model to avoid the computational complexity*, Ann. Oper. Res. 235(1) (2015), 599–623.

11. Stewart T.J. *Relationships between data envelopment analysis and multiple criteria analysis*, J. Oper. Res. Soc. 47 (1996), no. 5, 654–665.

12. Sueyoshi T. *Mixed integer programming approach of extended DEA-discriminant analysis*, Eur. J. Oper. Res. 152 (1) (2004), 45–55.

13. Thanassoulis E. and Dyson R.G. *Estimating preferred target input-output levels using data envelopment analysis*, Eur. J. Oper. Res. 56 (1992), 80–97.

14. Wu J., Zhu Q., An Q., Chu J., and Ji X. *Resource allocation based on context-dependent data envelopment analysis and a multi-objective linear programming approach*, Comput. Ind. Eng.101 (2016), 81–90.

15. Yougbar J. W. and Teghem J. *Relationships between Pareto optimality in multi-objective 0-1 linear programming and DEA efficiency*, Eur. J. Oper. Res. 183 (2007), 608–617.

16. Yun Y.B., Nakayama H., Tanino T. and Arakawa M. *Generation of efficient frontiers in multi-objective optimization problems by generalized data envelopment analysis*, Eur. J. Oper. Res. 129 (2001), 586–595.

Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 207–230
DOI:10.22067/ijnao.v9i2.75517

Research Article

An updated two-step method for solving interval linear programming: A case study of the air quality management problem

M. Allahdadi and A. Batamiz*

Abstract

Many real-world problems occur under uncertainty. In this paper, we consider interval linear programming (ILP) which can be used to tackle uncertainties. Several methods have been proposed by researchers, such as the best and worst cases, Two-step method (TSM), improved TSM, ILP, improved ILP, three-step method, and robust two-step method. First, we define feasibility and optimality conditions in ILP models and review some solving methods shortly, and then show that some solutions of the TSM method are not feasible. Therefore, we propose an updated TSM method (namely, UTSM) by considering the feasibility and optimality conditions. In this paper, the UTSM method was applied to identify the reduction of aerosols by using two controllers with a minimized cost to demonstrate its application under uncertainty. Compared with other methods, the solutions obtained through ILP were presented as interval, which can provide intervals for the decision variables, objective function, and decision-makers. Therefore, the decision-makers can make the best decision based on the obtained solutions through ILP, and then identify desired plans for aerosol-emission control under uncertainty.

AMS(2010): 65K10; 90C05; 65G40.

Keywords: Interval linear programming; TSM; Uncertainty; Aerosol.

*Corresponding author
Received 22 September 2018; revised 15 July 2019; accepted 6 August 2019
Mehdi Allahdadi
Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran. e-mail: m_allahdadi@math.usb.ac.ir

Aida Batamiz
Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran. e-mail: aida.batamiz@yahoo.com

1 Introduction

Interval linear programming (ILP) is described as the lower and upper bounds, which is used as a powerful tool to deal with optimization problems under uncertainty. Some researchers have been working on solving methods of ILP. In the best and worst cases (BWC) method proposed by Tong [30], the ILP model was converted into two submodels: the best and worst submodels, which they, respectively, have the largest and the smallest feasible regions. A given point is feasible for the ILP model if it satisfies the constraints of the best problem, and it is optimal for the ILP model if it is optimal for at least one characteristic model. The BWC method was extended by Chinneck and Ramadan for ILP models with equality constraints; see [7]. A novel ILP method was proposed by Huang and Moore [18], and Huang and Cao [17] also proposed their analyzed principals of two-step method (TSM). Some solutions obtained through the BWC, ILP, and TSM may be infeasible. New solving methods named three-step method (ThSM) and robust two-step method (RTSM) have been developed for solving ILP models; see [10, 17]. To guarantee that the given solutions of ILP method are completely feasible, Zhou et al. [33] proposed the modified ILP method (MILP) in which an extra constraint is added to the second submodel. Since the resulting solutions from the MILP may be nonoptimal, two improved ILP and MILP (IILP and IMILP) methods for solving ILP problems have been proposed; see [4]. The solutions of these methods are completely feasible and optimal; see [3]. Also, there are other methods for solving ILP models including IThSM, a new algorithm by Ashayerinasab et al. and a method proposed by Garajov et al.; see [2, 5, 12].

In this paper, we update the TSM (namely, UTSM) by considering the feasibility and optimality conditions. Following that, a numerical example is solved to demonstrate the effectiveness of UTSM. The air pollution caused by polluted resources, which cause pollution emissions in many sources, receptors, and artificial factors, including emissions of dangerous gases from factories and coal-fired power plants which pose serious threats to humans. These can have a greater impact under stable atmospheric conditions and increased pollution emissions. Pollution control and the reduction in the levels of atmospheric pollution are very important. These would approach the levels of atmospheric stability and prevent the destructive effects of pollutants. The proposed models can identify pollution sources and atmospheric weather conditions. These methods suggest pollution control using efficient methods for the study area. In this case, the levels of emissions of pollutants and environmental loading capacity need to be considered. Using optimization models, the costs should be reduced by controlling methods.

Another aim of the study is to check air pollution management under uncertainty as a range in the ILP model. The suggested methods will be analyzed to show the control and the reduction of dust by mulching and green belt controllers. However, the suggested methods are common methods

for dust and aerosol control. Now, to assess the effectiveness of the two controllers (mulching and green belts), the ILP model will be used and solved by the BWC, ITSM, and UTSM methods. The aim of this model is to minimize the costs in order to reduce aerosols by using two controllers.

2 Interval linear programming

In this section, we define the ILP model and some solving methods. An interval number $[X^-, X^+]$ is shown as $X^\pm$ where $X^- \leq X^+$. If $X^- = X^+$, then $X^\pm$ is degenerate. If A^- and A^+ are two matrices in $\mathbb{R}^{m\times n}$ such that $A^- \leq A^+$, then the set of matrices $A^\pm = [A^-, A^+] = \{A \,|A^- \leq A \leq A^+\}$ is called an interval matrix, and the matrices A^- and A^+ are called bounds of A. Center and radius matrices are defined as $\Delta_{A^\pm} = \frac{1}{2}(\mathrm{A}^+ - \mathrm{A}^-)$ and $A^c = \frac{1}{2}(A^- + A^+)$. A special case of an interval matrix is an interval vector $\mathbf{x}^\pm = \{\mathrm{x}|\mathrm{x}^- \leq \mathrm{x} \leq \mathrm{x}^+\}$, where $\mathrm{x}^-, \mathrm{x}^+ \in \mathbb{R}^{\mathrm{n}}$; see [1]. If $\mathbf{A}^\pm$ is a square interval matrix and each $A \in \mathbf{A}^\pm$ is nonsingular, then $\mathbf{A}^\pm$ is called regular. Consider the following ILP model:

$$\begin{aligned} \max \quad & f^\pm = \sum_{j=1}^{n} c_j^\pm x_j^\pm \\ s.t. \quad & \sum_{j=1}^{n} a_{ij}^\pm x_j^\pm \leq b_i^\pm, \quad i = 1, 2, \ldots, m, \\ & 0 \leq x_j^- \leq x_j^+, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{1}$$

The characteristic model of ILP model (1) is

$$\begin{aligned} \max f &= \sum_{j=1}^{n} c_j x_j \\ s.t. \sum_{j=1}^{n} a_{ij} x_j &\leq b_i, \quad i = 1, 2, \ldots, m, \\ x_j &\geq 0, \quad j = 1, 2, \ldots, n, \end{aligned}$$

where $a_{ij} \in a_{ij}^\pm, c_j \in c_j^\pm$, and $b_i \in b_i^\pm$. The feasible solution set of the ILP is defined as

$$\left\{ x \in R^n : \sum_{j=1}^{n} a_{ij}^- x_j \leq b_i^+, \, x_j \geq 0, \, i = 1, \ldots, \, m, \, j = 1, \ldots, \, n \right\}.$$

Also, the optimal solution set of the ILP is defined as the set of all optimal solutions over all the characteristic models. Some methods for solving ILP models are BWC, TSM, ITSM methods (see [16,30,31]), which are presented

in Table 1. Also f_{opt}^{-} and f_{opt}^{+} are the best and worst optimal values of the objective function and the optimal solution is $x_{j\,opt}^{\pm} = \left[x_{j\,opt}^{-},\, x_{j\,opt}^{+}\right]$.

We also have

$$\left|a_{ij}^{\pm}\right|^{-} = \begin{cases} a_{ij}^{-}, & a_{ij}^{\pm} \geq 0, \\ -a_{ij}^{+}, & a_{ij}^{\pm} < 0, \end{cases} \qquad \left|a_{ij}^{\pm}\right|^{+} = \begin{cases} a_{ij}^{+}, & a_{ij}^{\pm} \geq 0, \\ -a_{ij}^{-}, & a_{ij}^{\pm} < 0, \end{cases}$$

$$\mathrm{Sign}(\mathrm{a}_{\mathrm{ij}}^{\pm}) = \begin{cases} 1, & a_{ij}^{\pm} \geq 0, \\ -1, & a_{ij}^{\pm} < 0. \end{cases}$$

Note that in all methods, for the objective functions, the notations "+" and "-" are always used for the first submodel and the second submodel, respectively. But for the variables, in the TSM and ITSM methods, the notation "+" for x_j, $j = 1, \ldots, k$, in the first submodel and $j = k+1, \ldots, n$ in the second submodel is used. Also, the notation "-" for x_j, $j = k+1, \ldots, n$ in the first submodel and $j = 1, \ldots, k$ in the second submodel is used. In the BWC method, these notations are not used.

Remark: If a decision-maker wants to minimize the costs, then the BWC method can provide the lower and upper bounds of the cost. Now, an example is solved and then the results are compared. Consider the ILP model as follows:

$$\begin{aligned} \max \quad & f^{\pm} = [3.5,\, 4]\, x_1^{\pm} - [1.5,\, 1.7]\, x_2^{\pm} \\ s.t. \quad & \\ & [1.1,\, 1.2]\, x_1^{\pm} + [1.7,\, 1.9]\, x_2^{\pm} \leq [11.7,\, 12.1]\,, \\ & [4,\, 5]\, x_1^{\pm} - [3,\, 4]\, x_2^{\pm} \leq [5,\, 7]\,, \\ & 0 \leq x_1^{-} \leq x_1^{+}, \\ & 0 \leq x_2^{-} \leq x_2^{+}. \end{aligned} \tag{2}$$

The results obtained through the BWC, TSM, and ITSM are given in Tables 2 and 3 and Figure 1.

Some points of the solution space obtained by the BWC and TSM are infeasible. For example, the point (5.3839, 4.0076) in the BWC and the point (5.1417, 4.3388) in the TSM are infeasible because they do not satisfy the inequality $1.1x_1^{+} + 1.7x_2^{-} \leq 12.1$, which is the first constraint of the best problem. The solution space obtained by the ITSM is feasible.

In the next subsection, we will define optimality condition and show that some points of the BWC, TSM, and ITSM are nonoptimal, and then we propose a new method, which we call an updated TSM method.

Table 1: submodels of BWC, TSM, and ITSM

Methods	The first submodel	The second submodel
BWC	$\max \quad f^+ = \sum_{j=1}^{n} c_j^+ x_j$ $s.t. \quad \sum_{j=1}^{n} a_{ij}^- x_j \le b_i^+, \quad i = 1, 2, \ldots, m,$ $x_j \ge 0, \quad j = 1, 2, \ldots, n.$	$\max \quad f^- = \sum_{j=1}^{n} c_j^- x_j$ $s.t. \quad \sum_{j=1}^{n} a_{ij}^+ x_j \le b_i^-, \quad i = 1, 2, \ldots, m,$ $x_j \ge 0, \quad j = 1, 2, \ldots, n.$
TSM	$\max \ f^+ = \sum_{j=1}^{k} c_j^+ x_j^+ + \sum_{j=k+1}^{n} c_j^+ x_j^-$ $s.t.$ $\sum_{j=1}^{k} \lvert a_{ij}\rvert^- \mathrm{Sign}(a_{ij}^{\pm}) x_j^+ + \sum_{j=k+1}^{n} \lvert a_{ij}\rvert^+ \mathrm{Sign}(a_{ij}^{\pm}) x_j^- \le b_i^+$ $x_j^+ \ge 0, \ j = 1, \ldots, k$ $x_j^- \ge 0, \ j = k+1, \ldots, n$	$\max \ f^- = \sum_{j=1}^{k} c_j^- x_j^- + \sum_{j=k+1}^{n} c_j^- x_j^+$ $s.t.$ $\sum_{j=1}^{k} \lvert a_{ij}\rvert^+ \mathrm{Sign}(a_{ij}^{\pm}) x_j^- + \sum_{j=k+1}^{n} \lvert a_{ij}\rvert^- \mathrm{Sign}(a_{ij}^{\pm}) x_j^+ \le b_i^-$ $x_j^- \le x_{j\,opt}^+, \ j = 1, \ldots, k$ $x_j^+ \ge x_{j\,opt}^-, \ j = k+1, \ldots, n$ $x_j^{\pm} \ge 0, \ j = 1, \ldots, n$
ITSM	$\max \ f^+ = \sum_{j=1}^{k} c_j^+ x_j^+ + \sum_{j=k+1}^{n} c_j^+ x_j^-$ $s.t.$ $\sum_{j=1}^{k} \lvert a_{ij}\rvert^- \mathrm{Sign}(a_{ij}^{\pm}) x_j^+ + \sum_{j=k+1}^{n} \lvert a_{ij}\rvert^+ \mathrm{Sign}(a_{ij}^{\pm}) x_j^- \le b_i^+$ $x_j^+ \ge 0, \ j = 1, \ldots, k$ $x_j^- \ge 0, \ j = k+1, \ldots, n$	$\max \ f^- = \sum_{j=1}^{k} c_j^- x_j^- + \sum_{j=k+1}^{n} c_j^- x_j^+$ $s.t.$ $\sum_{j=1}^{k} \lvert a_{ij}\rvert^+ \mathrm{Sign}(a_{ij}^{\pm}) x_j^- + \sum_{j=k+1}^{n} \lvert a_{ij}\rvert^- \mathrm{Sign}(a_{ij}^{\pm}) x_j^+ \le b_i^-$ $\sum_{j=1}^{n} a_{ij}^- x'_j \le b_i^+,$ $x'_j = \begin{cases} x_{j\,opt}^+ & if \ \mathrm{Sign}(a_{ij}^{\pm}) = \mathrm{Sign}(c_j^{\pm}), \ j = 1, \ldots, k \\ x_j^- & if \ \mathrm{Sign}(a_{ij}^{\pm}) \ne \mathrm{Sign}(c_j^{\pm}), \ j = 1, \ldots, k \\ x_{j\,opt}^- & if \ \mathrm{Sign}(a_{ij}^{\pm}) = \mathrm{Sign}(c_j^{\pm}), \ j = k+1, \ldots, n \\ x_j^+ & if \ \mathrm{Sign}(a_{ij}^{\pm}) \ne \mathrm{Sign}(c_j^{\pm}), \ j = k+1, \ldots, n \end{cases}$ $0 \le x_j^- \le x_{j\,opt}^+, \ j = 1, \ldots, k$ $x_j^+ \ge x_{j\,opt}^-, \ j = k+1, \ldots, n$

Table 2: submodels of BWC, TSM, and ITSM for ILP model (2).

Methods	The first submodel	The second submodel
BWC	$\max \quad f^+ = 4x_1 - 1.5x_2$ $s.t.$ $1.1x_1 + 1.7x_2 \leq 12.1$ $4x_1 - 4x_2 \leq 7$ $x_1,\, x_2 \geq 0$	$\max \quad f^- = 3.5x_1 - 1.7x_2$ $s.t.$ $1.2x_1 + 1.9x_2 \leq 11.7$ $5x_1 - 3x_2 \leq 5$ $x_1,\, x_2 \geq 0$
TSM	$\max \quad f^+ = 4x_1^+ - 1.5x_2^-$ $s.t.$ $1.1x_1^+ + 1.9x_2^- \leq 12.1$ $4x_1^+ - 4x_2^- \leq 7$ $x_1^+,\, x_2^- \geq 0$	$\max \quad f^- = 3.5x_1^- - 1.7x_2^+$ $s.t.$ $1.2x_1^- + 1.7x_2^+ \leq 11.7$ $5x_1^- - 3x_2^+ \leq 5$ $x_1^- \leq x_{1\,opt}^+ \rightarrow x_1^- \leq 5.1417$ $x_2^+ \geq x_{2\,opt}^- \rightarrow x_2^+ \geq 3.3917$ $x_1^-,\, x_2^+ \geq 0$
ITSM	$\max \quad f^+ = 4x_1^+ - 1.5x_2^-$ $s.t.$ $1.1x_1^+ + 1.9x_2^- \leq 12.1$ $4x_1^+ - 4x_2^- \leq 7$ $x_1^+,\, x_2^- \geq 0$	$\max \quad f^- = 3.5x_1^- - 1.7x_2^+$ $s.t.$ $1.2x_1^- + 1.7x_2^+ \leq 11.7$ $5x_1^- - 3x_2^+ \leq 5$ $1.1x_{1\,opt}^+ + 1.7x_2^+ \leq 12.1$ $4x_{1\,opt}^+ - 4x_2^+ \leq 7$ $x_1^- \leq x_{1\,opt}^+$ $x_2^+ \geq x_{2\,opt}^-$ $x_1^-,\, x_2^+ \geq 0$

Table 3: Solutions of BWC, TSM, ITSM for ILP model (2).

Methods	$x_1^{\pm}$	$x_2^{\pm}$	$z^{\pm}$
BWC	[3.4046,5.3839]	[3.6339,4.0076]	[5.1031,16.0848]
TSM	[3.6033,5.1417]	[3.3917,4.3388]	[5.2355,15.4792]
ITSM	[3.2744,5.1417]	[3.3917,3.7906]	[5.0162,15.4792]

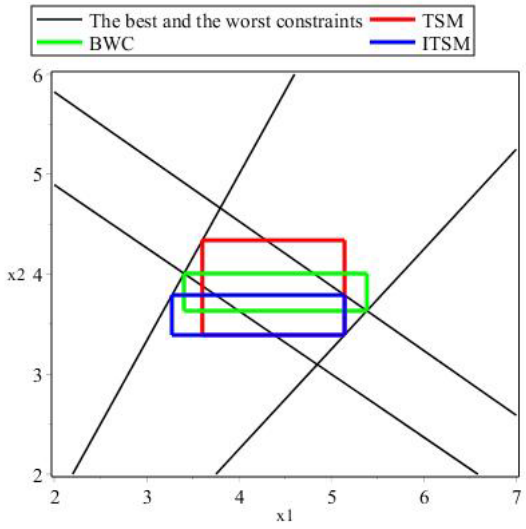


Figure 1: Solutions of ILP model (2)

2.1 A new method: An updated TSM

In this subsection, we suggest an updated TSM by considering both feasibility and optimality conditions. At first we review basis stability, and then obtain two feasibility and optimality conditions.

Definition 1 (see [15]). *The problem* $\max\{c^T x : Ax = b,\ x \geq 0\}$, *where* $c \in C^{\pm} \subseteq \mathbb{R}^n$, $A \in A^{\pm} \subseteq \mathbb{R}^{m\times n}$ *and* $b \in b^{\pm} \subseteq \mathbb{R}^m$ *is called* B*-stable with basis* B, *if* B *is an optimal basis for each characteristic model. The ILP model is called (unique) nondegenerate* B*-stable if each characteristic model has a (unique) nondegenerate optimal basic solution with the basis* B. *Let* $B \subseteq \{1, 2, \ldots, n\}$ *be an index set such that (the restriction of* A *to the columns indexed by* B*) is nonsingular. Similarly,* $N = \{1, 2, \ldots, n\} \setminus B$ *denotes the index set of nonbasic variables and* A_N *is the restriction of* A *to the columns indexed by* N. *Also,* B *can be computed by solving an arbitrary characteristic model. We now review the conditions for* B*-stability.*
Regularity: A_B *is regular.*
Feasibility: *The solutions set of the interval system* $A_B x_B = b$ *are nonnegative.*
Optimality: A_B *is optimal, that is,* $c_B^T A_B^{-1} A_N - c_N^T \geq 0^T$.

Theorem 1. *[27] If* $\rho\left(\left|(A^c)_B^{-1}\right| \Delta_{A_B}\right) < 1$, *then* A_B *is regular, where* $\rho(.)$ *denotes the spectral radius.*

Theorem 2. *[27] If* $\max_{1\leq i\leq n}\left(\left|(A^c)_B^{-1}\right| \Delta_{A_B}\right)_{ii} \geq 1$, *then* A_B *is not regular.*

Some conditions for the regularity of interval matrices have been proposed in [29].

Theorem 3. *[28] The interval vector r is an enclosure to the solution set of a system $A_B x_B = b$, where*

$$
\begin{aligned}
r_i^- &= \min\left\{-x_i^* + (x_i^c + |x_i^c|)M_{ii}) + \tfrac{1}{2M_{ii}-1}(-x_i^* + (x_i^c + |x_i^c|)M_{ii}\right\},\\
r_i^+ &= \max\left\{x_i^* + (x_i^c - |x_i^c|)M_{ii}) + \tfrac{1}{2M_{ii}-1}(x_i^* + (x_i^c - |x_i^c|)M_{ii}\right\},\\
M &= (I - \left|(A^c)_B^{-1}\right| \Delta_{A_B})^{-1},\ x^c = (A_B^c)^{-1} b^c,\\
x^* &= M(|x^c| + \left|(A^c)_B^{-1}\right| \Delta_b),
\end{aligned}
$$

with A_B^c is nonsingular and $\rho\left(\left|(A^c)_B^{-1}\right| \Delta_{A_B}\right) < 1$.

Theorem 4. *[15] Let y be an enclosure to the solution set of $A_B y = c_B$. If $\left(\left(A_N^T\right) y\right)^- \geq c_N^+$, then the optimality condition holds.*

Theorem 5. *[15] Let $diag(q)$ denote the diagonal matrix with entries $q_1, \ldots, q_m$. For each $q \in \{\pm 1\}^m$, if the solution set of the system*

$$
\begin{cases}
\left((A_B^c)^T - (\Delta_{A_B})^T\, diag(q)\right) y \leq c_B^+,\\
-\left((A_B^c)^T + (\Delta_{A_B})^T\, diag(q)\right) y \leq -c_B^-,\\
diag(q) y \geq 0,
\end{cases}
$$

lies in the solution set of the system

$$
\begin{cases}
\left((A_N^c)^T - (\Delta_{A_N})^T\, diag(q)\right) y \leq c_N^+,\\
diag(q) y \geq 0,
\end{cases}
$$

then the optimality condition holds.

Theorem 6. *[15] Let the ILP model be unique, nondegenerate B-stable, where $B = (1, \ldots, m)$. Then the optimal solution set of the ILP model is equal to the solution set of the interval system*

$$A_B x_B = b,\ x_B \geq 0,\ x_N = 0,\ or\ A_B^+ x_B \geq b^-,\ A_B^- x_B \leq b^+,\ x_B \geq 0,\ x_N = 0.$$

Now, we introduce two submodels of UTSM. Firstly, we solve the submodel corresponding to f^- ,which is the second submodel of the TSM. Secondly, we add two extra constraints in order to ensure that the resulting solution space is completely feasible and optimal in the submodel corresponding to f^+. Submodel 1 is as follows.

Submodel 1

$$\begin{aligned}
&\max\ f^- = \sum_{j=1}^{k} c_j^- x_j^- + \sum_{j=k+1}^{n} c_j^- x_j^+ \\
&s.t. \\
&\quad \sum_{j=1}^{k} |a_{ij}|^+ \mathrm{Sign}(\mathrm{a_{ij}^{\pm}})\mathrm{x_j^-} + \sum_{\mathrm{j=k+1}}^{\mathrm{n}} |\mathrm{a_{ij}}|^- \mathrm{Sign}(\mathrm{a_{ij}^{\pm}})\mathrm{x_j^+} \le \mathrm{b_i^-},\ \ \mathrm{i} = 1, \ldots \mathrm{m}, \\
&\quad x_j^- \ge 0,\ \ j = 1, \ldots, k, \\
&\quad x_j^+ \ge 0,\ \ j = k+1, \ldots, n,
\end{aligned} \tag{3}$$

For Submodel 2, $x_{j\,opt}^-$ for $j = 1, \ldots, k$ and $x_{j\,opt}^+$ for $j = k+1, \ldots, n$ are the optimal solutions of the first submodel.

$$\begin{aligned}
&\max\ f^+ = \sum_{j=1}^{k} c_j^+ x_j^+ + \sum_{j=k+1}^{n} c_j^+ x_j^-, \\
&s.t. \\
&\quad \sum_{j=1}^{k} |a_{ij}|^- \mathrm{Sign}(\mathrm{a_{ij}^{\pm}})\mathrm{x_j^+} + \sum_{\mathrm{j=k+1}}^{\mathrm{n}} |\mathrm{a_{ij}}|^+ \mathrm{Sign}(\mathrm{a_{ij}^{\pm}})\mathrm{x_j^-} \le \mathrm{b_i^+},\ \ \mathrm{i} = 1, \ldots \mathrm{m}, \\
&\quad 0 \le x_{j\,opt}^- \le x_j^+,\ \ j = 1, \ldots, k, \\
&\quad 0 \le x_j^- \le x_{j\,opt}^+,\ \ j = k+1, \ldots, n,
\end{aligned}$$

Now, we obtain an extra constraint to ensure that the solutions are optimal.

Theorem 6 implies that $\sum_{j=1}^{n} a_{ij}^+ x_j \ge b_i^-,\ i = 1, \ldots, m,$ or

$$\sum_{j=1}^{k-p} a_{ij}^+ x_j + \sum_{j=k-p+1}^{k} a_{ij}^+ x_j + \sum_{j=k+1}^{n-q} a_{ij}^+ x_j + \sum_{j=n-q+1}^{n} a_{ij}^+ x_j \ge b_i^-,\ i = 1, \ldots, m.$$

Since $a_{ij}^{\pm} \ge 0\ for\ j = 1, \ldots, k-p,\ a_{ij}^{\pm} \le 0\ for\ j = k-p+1, \ldots, k+1, \ldots, n-q$ and $a_{ij}^{\pm} \ge 0\ for\ j = n-q+1, \ldots, n$, therefore

$$\begin{aligned}
&\sum_{j=1}^{k-p} a_{ij}^+ x_j + \sum_{j=k-p+1}^{k} a_{ij}^+ x_j + \sum_{j=k+1}^{n-q} a_{ij}^+ x_j + \sum_{j=n-q+1}^{n} a_{ij}^+ x_j \\
&\qquad \ge \sum_{j=1}^{k-p} a_{ij}^+ x_{j\,opt}^- + \sum_{j=k-p+1}^{k} a_{ij}^+ x_j^+ + \sum_{j=k+1}^{n-q} a_{ij}^+ x_{j\,opt}^+ + \sum_{j=n-q+1}^{n} a_{ij}^+ x_j^-.
\end{aligned}$$

Therefore, for optimality, it is sufficient that

$$\sum_{j=1}^{k-p} a_{ij}^+ x_{j\,opt}^- + \sum_{j=k-p+1}^{k} a_{ij}^+ x_j^+ + \sum_{j=k+1}^{n-q} a_{ij}^+ x_{j\,opt}^+ + \sum_{j=n-q+1}^{n} a_{ij}^+ x_j^- \ge b_i^-,$$

or

$$\sum_{j=1}^{n} a_{ij}^+ x_j'' \ge b_i^-,\ x_j'' = \begin{cases}
x_{j\,opt}^- & if\ \mathrm{Sign}(\mathrm{a_{ij}^{\pm}}) = \mathrm{Sign}(\mathrm{c_j^{\pm}}),\ \ \mathrm{j} = 1, \ldots, \mathrm{k}, \\
x_j^+ & if\ \mathrm{Sign}(\mathrm{a_{ij}^{\pm}}) \ne \mathrm{Sign}(\mathrm{c_j^{\pm}}),\ \ \mathrm{j} = 1, \ldots, \mathrm{k}, \\
x_{j\,opt}^+ & if\ \mathrm{Sign}(\mathrm{a_{ij}^{\pm}}) = \mathrm{Sign}(\mathrm{c_j^{\pm}}),\ \ \mathrm{j} = \mathrm{k}+1, \ldots, \mathrm{n}, \\
x_j^- & if\ \mathrm{Sign}(\mathrm{a_{ij}^{\pm}}) \ne \mathrm{Sign}(\mathrm{c_j^{\pm}}),\ \ \mathrm{j} = \mathrm{k}+1, \ldots, \mathrm{n},
\end{cases}$$

To ensure that the solutions are feasible, let us obtain an extra constraint:

Theorem 6 implies that $\sum_{j=1}^{n} a_{ij}^{-} x_j \le b_i^{+},\ i=1,\ldots,m,$ or

$$\sum_{j=1}^{k-p} a_{ij}^{-} x_j + \sum_{j=k-p+1}^{k} a_{ij}^{-} x_j + \sum_{j=k+1}^{n-q} a_{ij}^{-} x_j + \sum_{j=n-q+1}^{n} a_{ij}^{-} x_j \le b_i^{+},\ i=1,\ldots,m.$$

Since $a_{ij}^{\pm} \ge 0$ for $j=1,\ldots,k-p,\ a_{ij}^{\pm} \le 0$ for $j=k-p+1,\ldots,k+1,\ldots,n-q$ and

$a_{ij}^{\pm} \ge 0$ for $j=n-q+1,\ldots,n$, therefore

$$\sum_{j=1}^{k-p} a_{ij}^{-} x_j + \sum_{j=k-p+1}^{k} a_{ij}^{-} x_j + \sum_{j=k+1}^{n-q} a_{ij}^{-} x_j + \sum_{j=n-q+1}^{n} a_{ij}^{-} x_j \le \sum_{j=1}^{k-p} a_{ij}^{-} x_j^{+} + \sum_{j=k-p+1}^{k} a_{ij}^{-} x_{j\,opt}^{-} + \sum_{j=k+1}^{n-q} a_{ij}^{-} x_j^{-} + \sum_{j=n-q+1}^{n} a_{ij}^{-} x_{j\,opt}^{+}$$

Therefore, for feasibility, it is sufficient that

$$\sum_{j=1}^{k-p} a_{ij}^{-} x_j^{+} + \sum_{j=k-p+1}^{k} a_{ij}^{-} x_{j\,opt}^{-} + \sum_{j=k+1}^{n-q} a_{ij}^{-} x_j^{-} + \sum_{j=n-q+1}^{n} a_{ij}^{-} x_{j\,opt}^{+} \le b_i^{+} \text{ or}$$

$$\sum_{j=1}^{n} a_{ij}^{-} x'_j \le b_i^{+},\ x'_j = \begin{cases} x_j^{+} & if\ \mathrm{Sign}(a_{ij}^{\pm}) = \mathrm{Sign}(c_j^{\pm}),\ j=1,\ldots,l, \\ x_{j\,opt}^{-} & if\ \mathrm{Sign}(a_{ij}^{\pm}) \ne \mathrm{Sign}(c_j^{\pm}),\ j=1,\ldots,l, \\ x_j^{-} & if\ \mathrm{Sign}(a_{ij}^{\pm}) = \mathrm{Sign}(c_j^{\pm}),\ j=l+1,\ldots,n, \\ x_{j\,opt}^{+} & if\ \mathrm{Sign}(a_{ij}^{\pm}) \ne \mathrm{Sign}(c_j^{\pm}),\ j=l+1,\ldots,n, \end{cases}$$

Finally, we have $f_{opt}^{\pm} = \left[f_{opt}^{-}, f_{opt}^{+}\right],\ x_{j\,opt}^{\pm} = \left[x_{j\,opt}^{-}, x_{j\,opt}^{+}\right]$. Therefore, the second submodel is as follows.

Submodel 2

$$\max\ f^{+} = \sum_{j=1}^{k} c_j^{+} x_j^{+} + \sum_{j=k}^{n} c_j^{+} x_j^{-}$$

$$s.t.\quad \sum_{j=1}^{k} |a_{ij}|^{-}\mathrm{Sign}(a_{ij}^{\pm}) x_j^{+} + \sum_{j=k+1}^{n} |a_{ij}|^{+}\mathrm{Sign}(a_{ij}^{\pm}) x_j^{-} \le b_i^{+},\ i=1,\ldots m,$$

$$0 \le x_{j\,opt}^{-} \le x_j^{+},\ j=1,\ldots,k,$$

$$0 \le x_j^{-} \le x_{j\,opt}^{+},\ j=k+1,\ldots,n,$$

$$\sum_{j=1}^{n} a_{ij}^{+} x'' \ge b_i^{-},\ x''_j = \begin{cases} x_{j\,opt}^{-} & if\ \mathrm{Sign}(a_{ij}^{\pm}) = \mathrm{Sign}(c_j^{\pm}),\ j=1,\ldots,k, \\ x_j^{+} & if\ \mathrm{Sign}(a_{ij}^{\pm}) \ne \mathrm{Sign}(c_j^{\pm}),\ j=1,\ldots,k, \\ x_{j\,opt}^{+} & if\ \mathrm{Sign}(a_{ij}^{\pm}) = \mathrm{Sign}(c_j^{\pm}),\ j=k+1,\ldots,n, \\ x_j^{-} & if\ \mathrm{Sign}(a_{ij}^{\pm}) \ne \mathrm{Sign}(c_j^{\pm}),\ j=k+1,\ldots,n, \end{cases}$$

$$\sum_{j=1}^{n} a_{ij}^{-} x' \le b_i^{+},\ x'_j = \begin{cases} x_{j\,opt}^{+} & if\ \mathrm{Sign}(a_{ij}^{\pm}) = \mathrm{Sign}(c_j^{\pm}),\ j=1,\ldots,l, \\ x_j^{-} & if\ \mathrm{Sign}(a_{ij}^{\pm}) \ne \mathrm{Sign}(c_j^{\pm}),\ j=1,\ldots,l, \\ x_{j\,opt}^{-} & if\ \mathrm{Sign}(a_{ij}^{\pm}) = \mathrm{Sign}(c_j^{\pm}),\ j=l+1,\ldots,n, \\ x_j^{+} & if\ \mathrm{Sign}(a_{ij}^{\pm}) \ne \mathrm{Sign}(c_j^{\pm}),\ j=l+1,n. \end{cases}$$

Therefore, the constraints $\sum_{j=1}^{n} a_{ij}^{-} x_j{}' \le b_i^{+}$ and $\sum_{j=1}^{n} a_{ij}^{+} x_j'' \ge b_i^{-}$ are feasibility and optimality conditions, respectively.

Note that, since the solution regions obtained through the UTSM method is completely optimal, the union of these regions will be closer to the exact optimal solution region of the ILP model.

Now, consider ILP model (2), again. Given Figure 1, the points (3.4046, 3.6330), (3.6033, 3.3917), and (3.2744, 3.3917) obtained by the BWC, TSM, and ITSM respectively, are nonoptimal, because they do not satisfy inequality $1.2x_1+1.9x_2 \geq 11.7$, which is the first constraint of the worst problem with the reverse sign. Model (2) is equivalent to the following model:

$$\begin{aligned}
&\max \quad z^{\pm} = [3.5,\ 4]\, x_1^{\pm} - [1.5,\ 1.7]\, x_2^{\pm} \\
&s.t. \\
&\qquad [1.1,\ 1.2]\, x_1^{\pm} + [1.7,\ 1.9]\, x_2^{\pm} + x_3^{\pm} = [11.7,\ 12.1]\,, \\
&\qquad [4,\ 5]\, x_1^{\pm} - [3,\ 4]\, x_2^{\pm} + x_4^{\pm} = [5,\ 7]\,, \\
&\qquad 0 \leq x_1^{-} \leq x_1^{+}, \\
&\qquad 0 \leq x_2^{-} \leq x_2^{+}, \\
&\qquad 0 \leq x_3^{-} \leq x_3^{+}, \\
&\qquad 0 \leq x_4^{-} \leq x_4^{+}.
\end{aligned}$$

According to Theorems 1, 3, and 4, ILP model 3 is B-stable, where $B = (1, 2)$. Now, we solve the model by the UTSM.

submodel 1

$$\begin{aligned}
&\max \quad z^{-} = 3.5x_1^{-} - 1.7x_2^{+} \\
&s.t. \\
&\qquad 1.2x_1^{-} + 1.7x_2^{+} \leq 11.7, \\
&\qquad 5x_1^{-} - 3x_2^{+} \leq 5, \\
&\qquad x_1^{-},\ x_2^{+} \geq 0,
\end{aligned}$$

submodel 2

$$\begin{aligned}
&\max \quad z^{+} = 4x_1^{+} - 1.5x_2^{-} \\
&s.t. \\
&\qquad 1.1x_1^{+} + 1.9x_2^{-} \leq 12.1, \\
&\qquad 4x_1^{+} - 4x_2^{-} \leq 7, \\
&\qquad x_1^{+} \geq x_{1\,opt}^{-} = 3.6033, \\
&\qquad x_2^{-} \leq x_{2\,opt}^{+} = 4.3388, \\
&\qquad \begin{cases} 1.1x_1^{+} + 1.7x_{2\,opt}^{+} \leq 12.1, \\ 4x_1^{+} - 4x_2^{-} \leq 7, \end{cases} \\
&\qquad \begin{cases} 1.9x_2^{-} \geq 1.7x_{2\,opt}^{+}, \\ 5x_1^{+} - 3x_2^{-} \geq 5x_{1\,opt}^{-} - 3x_{2\,opt}^{+}, \end{cases} \\
&\qquad x_1^{+}, x_2^{-} \geq 0.
\end{aligned}$$

Now, the results obtained by the UTSM have been given in Table 4 and Figure 2.

Table 4: Solutions of UTSM for ILP model (2).

Methods	$x_1^{\pm}$	$x_2^{\pm}$	$z^{\pm}$
UTSM	[3.6033,4.2945]	[3.8820, 4.3388]	[5.2355,11.3550]

As shown in Figure 2, the UTSM introduces a feasible and optimal space. All

points satisfy the constraints $1.1x_1 + 1.7x_2 \leq 12.1$ and $4x_1 - 4x_2 \leq 7$ (feasibility conditions) and the constraints $1.2x_1 + 1.9x_2 \geq 11.7$ and $5x_1 - 3x_2 \geq 5$ (optimality conditions).

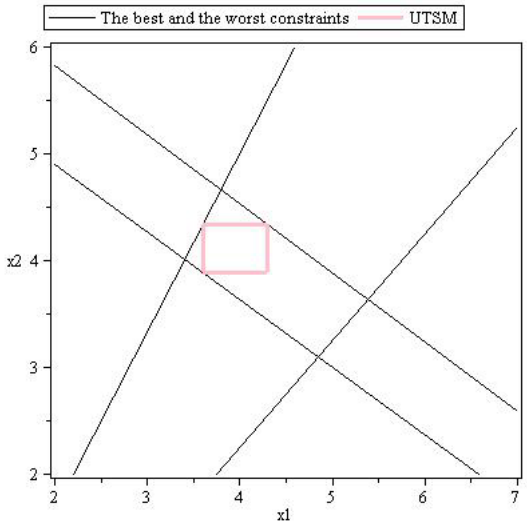


Figure 2: The solution region obtained by the UTSM for ILP model (2)

3 Application to air quality management

Atmospheric conditions can be affected by many factors, including SO2, CO, aerosols, dust, and so on. Aerosols have a very high diversity. As one of the greatest pollutants, this phenomenon occurs in arid, semiarid regions, and deserts, such that their sizes are different according to the concentrations in air. These can have destructive effects on health, nature, and the environment and create numerous economic problems for local people. It can be said that sandstorms and dust-storms are natural events with significant concentrations of particulate matter, which depend on the wind speeds. When the wind speed is almost more than 8 meters per second (and sometimes based on the region, which is stable or not, the wind speed is more than 6 meters/seconds), it can get into the air flow and the earth's atmosphere based on the roughness, particulate size, soil texture, soil moisture, and vegetation; see [9, 32]. The suggested methods are mulching and green belt that should be analyzed to control and reduce dust. These suggested methods are conventional methods for dust control, and many studies have been conducted in this area.

Mulch is a non-life coverage used for sand dune fixation in the desert lands. A variety of organic and inorganic mulches are available that can be used to accelerate production, reduce erosion, and even help control weeds. In Iran, oil mulch is usually used to cover. Most of the causes of oil mulching are the immediate effect, using them is provided in a short time, material source is available in the country, and there is no need for foreign resources.

Vegetation can have a major impact on air pollution and climate quality. It can also directly or indirectly decrease the temperature, increase moisture and dust absorption, and affect urban air pollution control. Studies on this subject have concluded that trees can greatly control air pollution, prevent harmful pollutants like PM10, SO2, NO2 and CO, and particulate matter in the atmosphere. Studies have shown that a lot of PM10 and PM2.5 is destroyed by trees, such that the efficiency and effectiveness of this factor plays a very important role in social and human health, and air quality control. In many countries nowadays, trees are being planted to reduce costs and harmful factors in the atmosphere; see [19, 21, 22].

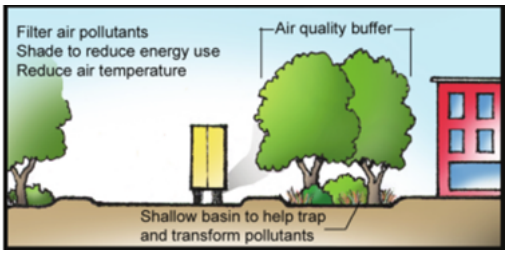


Figure 3: Trees as air quality buffer.

Figure 4: Trees as air quality buffer.

It has been well-documented that the plants destroy air pollutants, including hydrogen fluoride, sulfur dioxide, and some combinations of photochemical reactions in the air. It can be said that the green belt efficiency in the reduction of pollution defined by the following formula:

$$E = \frac{R}{R + A}$$

As E is the relative reduction effect by trees (percent), R is amount removed by trees (kg), and A is the amount of pollution in the atmosphere (kg).

3.1 Methodology and overview of the study system

The Sistan and Baluchestan region in Iran is located in the southeast, with arid and semi-arid weather. The Sistan plain covers a wide range of the western part of the province, and almost a wide catchment area. In addition to Iran, it has expanded into two countries Afghanistan and Pakistan. The border areas between Iran, Pakistan, and Afghanistan are the main dust source region in southwest of Asia; generating some 81 dust storms over Sistan; see [13, 24–26]. The Sistan region is one of the richest regions of Iran in terms of wind, and this sometimes creates difficulties for its people. Dust storms generate great waves, especially in Sistan. Figure 6 shows the geographical location of the Sistan and Baluchestan region.

Figure 5: Geographical location of the Sistan and Baluchestan region

The Sistan's 120-day winds are the most famous local winds of Iran, blowing from mid-May to mid-September, over a large area in the northern part of Sistan-Baluchestan. The strong "Levar" winds in summer favor the uplift of large quantities of dust from the Hamoun basin, which is located in the northern part of Sistan; see [25]. The 120-day winds of Sistan–in terms of time and local distribution and wind speed–have a high variety. In Sistan, the strong winds above eight nodes start in May and continue till November. In Sistan, winds are severe in summer, but can be slowed down in other seasons. Therefore, in this section, we focus on the optimization of an air

quality management model and PM10 emissions control in the Sistan region in a source through controllers, as mentioned in the previous section.

The aim of an air quality management model is to control effects on health by air pollution. These effects are directly related to pollutant concentration. Now, we establish the relationship between air pollution emissions and loading concentrations. The ground concentration at each downwind location (x, y) can be estimated as follows [14]:

$$C(x,y) = \frac{Q}{\pi u \sigma_y \sigma_z} \exp\left[\left(\frac{-y^2}{2{\sigma_z}^2}\right) - \left(\frac{H^2}{2{\sigma_z}^2}\right)\right],$$

where $C(x, y)$ is the ground-level concentration of pollutants (mg/m3) at point (x, y); x is the downwind distance from the source; Q denotes pollutant emission rate (mg/sec); u is average wind speed (m/ sec) along x direction; and σ_y and σ_z represent horizontal and vertical dispersion coefficients of plume versus(against) downwind distance x from the pollution source. Generally, σ_y and σ_z can be approximated by the following equations [8, 23]:

$$\sigma_y = \gamma_1 x^{\beta_1} \quad , \quad \sigma_z = \gamma_2 x^{\beta_2}.$$

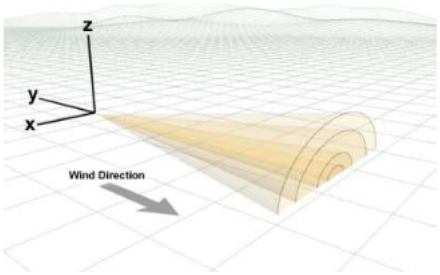


Figure 6: Dust-plume geometry

where γ_1 and β_1 are constants along the y direction and γ_2 and β_2 are the coefficients along the z direction, respectively. The values of γ_1, β_1, γ_2, and β_2 are related to atmospheric stability classes defined for different meteorological situations through wind speed, solar radiation (during the day), and cloud cover during the night [14, 23].

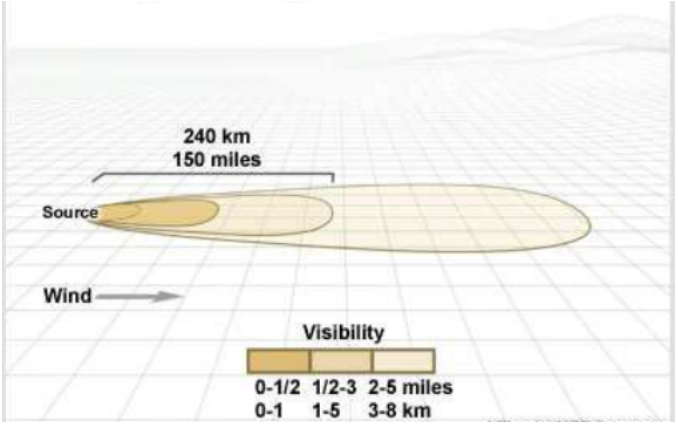


Figure 7: Typical visibility within a dust-storm

A transfer factor t_p can then be determined as follows [14]:

$$t_p = \frac{1}{\pi u \sigma_y \sigma_z} \exp\left[\left(\frac{-y_p^2}{2\sigma_y^2}\right) - \left(\frac{H^2}{2\sigma_z^2}\right)\right],$$

where t_p represents the contributions of pollutant emission rate at the Sistan area to the ground concentration of receptor zone p; the index p indicates different receptors influenced by air pollution emissions; H indicates the effective height of the air pollution plume from the Sistan region [6].

3.2 ILP model for air quality management modeling

The optimization model of air quality management has been presented as fuzzy linear programming [11,20]. Now, we consider the following ILP model:

$$
\begin{aligned}
Min \quad & \sum_{j=1}^{2}\sum_{k=1}^{5} L_k TC_{jk}^{\pm} X_{jk}^{\pm} \\
s.t. \quad & \\
& X_{jk}^{\pm} \le FC_{jk}^{\pm}\ , \text{ for all } j,k, \\
& \sum_{j=1}^{2} X_{jk}^{\pm} \ge S_k^{\pm}, \text{ for all } k, \\
& \sum_{j=1}^{2} \left(1-\eta_j^{\pm}\right) X_{jk}^{\pm} \le e_k^{\pm}, \text{ for all } k, \\
& \sum_{j=1}^{2} t_p \left(1-\eta_j^{\pm}\right) X_{jk}^{\pm} \le \theta_{kp}^{\pm}, \text{ for all } k,p, \\
& 0 \le X_{jk}^{-} \le X_{jk}^{+}, \text{ for all } j,k,
\end{aligned}
\tag{4}
$$

where i is the PM10 emission source, j is the type of PM10 control method ($j = 1$, 2, respectively, denotes mulching and green belt), and k is the planning period. The planning horizon is five years. Decision variables are the amount of PM10 allocated to control measure j for the reduction from source i in the period k (t/day). It has been shown by the symbol X_{ijk}. The objective is to minimize the total cost for the abatement of PM10. Now, the model describes the study area according to particulate reduction methods in this article.

Also L_k is length of period k (day), TC_{jk} is operating cost of control measure j during period k (Rials/t), FC_{jk} is maximum mitigation capacity of measure j in period k, ? is PM10 generation amount of emission source in period k (tonne/day); η_j is efficiency of control measure j, e_k is PM10-emission allowance for Sistan during period k (tonne/day), t_p is transfer factor from emission area to receptor zone p (day/m3), and θ_{kp} is environmental loading capacity of receptor zone during period (mg/m3). Note $i = 1$ represents the emission source of PM10 in the Sistan region. All the parameters in model (4) are presented as interval numbers. In this case, the parameters of TC_{jk}, FC_{jk}, S_k, η_j, e_k, and θ_{kp} are assumed to be estimated as interval numbers. Particulate matter in the Sistan region is discussed over four-month periods (the maximum amount of dust and particulate matter) during five years. We estimate-according to two methods, mulching and green belts-the tables are related to the costs of each method, efficiencies, and permitted amounts of dust in the region, and show their reduced values by using the two methods in the respective tables. Here, S_k is the aerosol generation amount of emission sources in period k in Sistan, which has been shown in Table 5.

The parameters related to the model in the Sistan region are now described. Tc_{jk} represents mulching costs per hectare with respect to the inflation rate (million). If we require 10 tonnes mulchs for every hectare and is equal to 20,000 Rials per liter of mulch, then the amount calculated using the inflation rate will be equal to Table 6.

Table 5: S_k-emission rate for ILP model (4)

Months	2012	2013	2014	2015	2016
May	[55,1345]	[76.6,4985]	[62.5,4983.6]	[15.3,460.1]	[45.6,4985]
June	[130.7,1384.8]	[60.1,2570.4]	[140.2,4985]	[17.2,452.4]	[49.1,630.8]
July	[315.6,9985]	[64.6,3044.6]	[4979.4,4985]	[10.5,4985]	[56.2,2489.4]
August	[100,1455.7]	[280.2,3553.5]	[130.2, 4985]	[880.9,4840]	[92.2,4983.6]

Table 6: Costs of different pollution controls for ILP model (4)

Tc_{jk}	2012	2013	2014	2015	2016
Mulching	[8.9,18]	[12.1,24.1]	[15,30]	[16.8,33.8]	[16.6,37.3]
Green belt	[2.5,6]	[3,6]	[3,7]	[4,8]	[5,10]

Figure 8: Dust precipitation

where e_{ik} denotes the aerosol (PM10) emission allowance for the source (Sistan) during period k (t/day) . If FC_{jk} is the maximum mitigation capacity of measure j at area in period k, then the amount of the reduction by the mulching and green belts methods can be seen in Tables 8 and 9.

Table 7: e_k for ILP model (4)

PM10-emission allowance	PM10
Good	0-54
Average (allowance)	55-154
Unhealthy for sensitive groups	155-254
Unhealthy	255-354
Very Unhealthy (crisis)	355-424
Dangerous (emergency)	425-604

Table 8: The value of PM10 emissions controlled by mulching

Months	2012	2013	2014	2015	2016
May	[44,1076]	[61.3,3988]	[50,3986.9]	[12.24,368.08]	[36.48,3988]
June	[104. 6,1108]	[48.08,2056.3]	[112.2,3988]	[13.76,361.9]	[39.28,504.6]
July	[252.5,7988]	[51.68,2435.9]	[4979.4,3989.1]	[8.4,3988]	[44.96,1991.5]
August	[80,1164.6]	[224.16,2842.8]	[104.16, 3988]	[704.72,3872]	[73.76,3986.9]
FC_{jk}	[120,2834.1]	[96.3,2830.7]	[1311.4,3988]	[184.8,2147.5]	[48.62,2617.8]

Table 9: The value of PM10 emissions controlled by green belt

Months	2012	2013	2014	2015	2016
May	[34.4,840.6]	[47.875,3115.6]	[39.06,3114.7]	[9.56,287.6]	[28.5,3115.6]
June	[81.7,865.5]	[37.6,1606.5]	[87.625,3115.6]	[10.7,282.7]	[30.687,394.2]
July	[197.3,6240.6]	[40.375,1903.1]	[3111.9,3115.6]	[6.56,3115.6]	[35.125,1555.9]
August	[62.5,909.8]	[175.1,2220.9]	[81.375,3115.6]	[550.56,3025]	[57.625,3114.7]
FC_{jk}	[193.9,2214.1]	[75.2,2211.5]	[829.98,2336.5]	[144.345,1677.7]	[37.98,2045.1]

Particulate reduction-with an efficiency of 75%-and relative reduction effect impact 0.2 and efficiency 40%, and the relative reduction effect impact 0.375 have been shown by a green belt. $\theta_{kp} = [45, 55]$ denotes environmental loading capacities at area p in period k; transfer factor t_p from source to area p is 1.1587×10^{-26}. The minimum and maximum wind speeds are $[6\ m/s,\ 15\ m/s]$ during the 120 days; $e_k = [55, 154]$ shows the PM10 emission allowance during the period k (t/day) in Sistan.

Table 10: PM10 efficiencies of different control measures

Methods efficiency	Interval efficiency
Mulching method η_1	[0.4, 0.6]
Green belt method η_2	[0.40, 0.75]

We now solve the air pollution control model using the methods mentioned in Section 2. The results are given in Table 11.

Table 11: Results obtained from solving ILP model (4)

	BWC	ITSM	UTSM
$Z \times 10^7$	[0.1490,6.0239]	[2.19050165304,5.16560105314]	[2.50191551199,7.7566276346]
x_{1opt}	[0.0,3348.7]	[3634.1,3834.1]	[3834.1 , 6795.9]
x_{2opt}	[0.0,3463.2]	[3730.7, 4830.7]	[4830.7 , 8152.3]
x_{3opt}	[136.0,4963.8]	[4988,5348]	[5071.4 , 8480]
x_{4opt}	[0.0, 2540.1]	[3307.5,4147.5]	[4147.5 , 7222.4]
x_{5opt}	[0.0,3234.3]	[3317.8,4117.8]	[4117.8 , 7181.9]
x_{6opt}	[150.3,194.0]	[193.95,663.7]	[240.6 , 240.6]
x_{7opt}	[75.2,120.4]	[75.2,521.3]	[46.8 , 46.8]
x_{8opt}	[20.9,1192.0]	[16.2,447.4]	[1.1044e-11 , 2.7188e-07]
x_{9opt}	[144.3,231.0]	[144.4,618.9]	[179.6 , 179.6]
x_{10opt}	[379.0,60.8]	[3798,623.2]	[185.4 , 185.4]

Results and analysis

Tables related to the reduction of particulates are given in the previous section, using two controllers as mulching and green belt. It can be said that the Sistan region aerosols were considered over four months in five-year periods, such that the lower bound is the lowest amount of the particulate matter PM10 and upper bound represent the most amount of the particulate matter PM10 in Sistan. Now, using the two above mentioned controller, the reduction of aerosols is calculated according to efficiencies in every method and these estimations are presented in Tables 8–11. Notably, the costs associated with each method can be considered based on the funding in the environmental section and different covering lands. Inflation in the country has been considered over the years according to variable data of costs and inflations as well as other parameters studied in the Sistan region. The greatest reductions in the particulate matter were calculated by the ILP model and the presented formulas.

For greater reliability and accuracy of calculations, the ILP would be solved by three methods: BWC, ITSM, and UTSM. The variations in the reduction of aerosols are studied. The results show that if the aim is to minimize the cost for decision-makers and officials, then the BWC method could be the best method because it can calculate the minimum and maximum of the costs under the two optimistic and pessimistic models (the best and worst) for finding the best answer. For example, the cost per hectare–using mulching–the minimum cost is estimated $\sim 0.1490 \times 10^7$ and the maximum cost is estimated $\sim 6.0239 \times 10^7$; Therefore, these are provided as an interval under the upper and lower bounds.

In the BWC method, as mentioned in Section 2, the resulting solution does not exactly hold into the constraints. For example, in the first constraint, the amount of reduction in particulates in every period may be more than the greatest reduction in particulates in each period. Therefore, UTSM and ITSM methods are used. In this case, maybe because of the reduction area which is created by the solutions, the model cannot be considered good enough in order to estimate the reduction of costs. It can make changes in the costs in some parts to find the best answer and increasing them because the constraints are not violated. Hence, we can say to find the best answer for the reduction of particulate matter, ITSM and UTSM methods are more suitable. In Table 11, 10 optimal solutions are given which are shown with the symbol x_{opt}. The first five numbers represent x_{opt} according to the mulching and the second set of five numbers is x_{opt} according to the green belt over five years. As seen in the ITSM method, the changes are: [3634.1, 3834.1] in 2012; in that year, the minimum value is 55 and the maximum value is 9985, which show the most amount of particulate matter in July. The above models considered all solutions in a feasible area for one another and the maximum value is 9985, which estimated the changes in the reduction of particulates [3634.14, 3834.14].

ITSM method enabled shows a reduction in the intervals. Using the UTSM method, which can be said to be one of the best methods, it seeks to find the best solution in the feasible area. Optimally, the reduction of particulates is [3834.14, 6795.9128], which has obtained good results, and further reductions are anticipated. Also, in 2013, the minimum value is 62.5 for the entire year and the maximum value is 4985, such that the reduction of the particulate matter has been estimated at [3730.7, 4830.7]. This indicates that in 2013, according to the amount of particulates, [3730.7, 4830.7] can undergo a decrease. It can be said that the obtained values from the model show a strong reaction over the years, according to the highest amount of the particulate matter and using the above analysis, we have the amount of particulates [4830.7 , 8152.3416] in the UTSM method similarly. It seeks to solve and attempts to reduce it, but according to the obtained numbers, the reduction is less in some years, it is because of the amount of particulate matter which is very large in all months of the year. However, there are the powerful and useful methods to control particulate matter, those may not be able to greatly reduce particulates, and many environmental factors can cause fewer reductions. The other numbers are explained similarly.

4 Conclusion

In this paper, some methods for solving the ILP models have been reviewed, such as the BWC, TSM, and improved TSM (ITSM). Besides, an updated two-step method (UTSM) has been proposed. This method improves the TSM by considering two feasibility and optimality conditions. The solutions comparison for the BWC, TSM, ITSM, and UTSM methods based on the feasibility and optimality conditions shows that some points of the BWC and TSM may be infeasible. Also, some solutions of the ITSM method may be non-optimal. The advantage of the UTSM is that, two constraints have been added in the second submodel which guarantees the feasibility and optimality conditions, and so the obtained region by the UTSM will be feasible and optimal. Also, a case study related to air quality management has been done.

For solving many uncertain problems in the real-world, different models are used. The ILP model which is one of the models under uncertainty can be used for modeling these problems. In this paper, we consider an ILP model related to minimizing the costs in order to control aerosols and solve it by the BWC, ITSM, and UTSM methods. The results show that the lower and upper bounds of the costs obtain by the BWC method. The reduction of particulates can be clearly seen in two ITSM and UTSM methods.

Acknowledgements

We would like to thank the editor- in-chief and the anonymous reviewers for the useful remarks and valuable suggestions which helped to improve this paper.

References

1. Alefeld, G. and Herzberger, J. *Introduction to interval computations*. Academic Press, New York, 1983.

2. Allahdadi, M. and Deng, C.*An improved three-step method for solving the interval linear programming problem*, Yugoslav J. Oper. Res. 28 (4) (2018), 435–451.

3. Allahdadi, M. and MishmastNehi H. *The optimal solution set of the interval linear programming problems*, Optim. Lett. 7(2013), 893–1911.

4. Allahdadi, M., MishmastNehi, H. Ashayerinasab, H.A. and Javanmard, M. *Improving the modified interval linear programming method by new techniques*. Inf. Sci. 339 (2016), 224–236.

5. Ashayerinasab, H. A., Mishmast Nehi, H. and Allahdadi, M. *Solving the interval linear programming problem: A new algorithm for a general case*, Expert Sys. Appl. 93 (2018), 1, 39–49.

6. Atmospheric dust (2009), Comet Program, http://kejian1.cmatc.cn/vod /comet/mesoprim/at_dust/print.htm.

7. Chinneck, J. W. and Ramadan, K. *Linear programming with interval coefficients* J. Oper. Res. Soc. 51 (2000), 209–220.

8. Davidson, G.A. *A modified power law representation of the pasquuill gifford dispersion coefficients* J. Air Waste Manage. Assoc. 40 (1990), 1146–1147.

9. Engelstaedter. S, Tegen. I, and Washington. R. *North African dust emissions and transport*, Earth-Sci. Rev. 79 (2006), 73–100.

10. Fan, Y.R., and Huang, G.H. *A robust two-step method for solving interval linear programming problems within an environmental management context*, J. Environ. Inform. 19 (1) (2012), 1–9.

11. Fan, Y., Huang, G. and Veawab, A. *A generalized fuzzy linear programming approach for environmental management problem under uncertainty*, J. Air Waste Manage. Assoc. 62(1) (2011), 72–86.

12. Garajova, E., Hladk, M. and Rada, M. *The effects of transformations on the optimal set in interval linear programming*, In Proceedings of the 14th International Symposium on Operational Research, SOR17, Bled, Slovenian Society Informatika, Ljubljana, Slovenia, (2017), 487–492.

13. Goudie A.S. and Middleton N.J. *desert dust in the global system*, Springer Berlin Heidelberg , Berlin (2006).

14. Haith, A.D. *Environmental systems optimization.* John Wiley & Sons, New York, 1982.

15. Hladk, M. *How to determine basis stability in interval linear programming*, Optim.Lett. 8 (2014), 375–389.

16. Huang, G.H., Baetz, B.W. and Patry, G.G. *A gray integer programming: an application to waste management planning under uncertainty*, Eur. J. Oper. Res. 83 (1995), 594–620.

17. Huang, G.H. and Cao, M.F. *Analysis of solution methods for interval linear programming*, J. Environ. Inform. 17 (2) (2011), 54–64.

18. Huang, G.H. and Moore, R.D. *Grey linear programming, its solving approach, and its application*, Int. J. Sys. Sci. 24 (1993), 159–172.

19. Islam, M.N., Rahman, K.S., Bahar, M.M., Habib, M.A., Ando, K. and Hattori, N. *Pollution attenuation by roadside green belt in and around urban areas*, Urban For. Urban Gree. (2012), 460–464.

20. Li, Y.P., Huang, G.H., Guo, P., Yang, Z.F. and Nie S. *A dual-interval vertex analysis method and its application to environmental decision making under uncertainty.* Eur. J. Oper. Res. 200 (2010), 536–550.

21. Nowak, D.J., Crane, D.E., and Stevens, J.C. *Air pollution removal by urban trees and shrubs in the United States*, Urban For. Urban Gree. 4 (2006), 115–123.

22. Nowak, J., Hirabayashi. S., Bodine. A. and Hoehn. R. *Modeled PM2.5 removal by trees in ten U.S. cities and associated health effects*, Environ. Pollut. 178 (2013), 395–402.

23. Pasguill, F.S. *Atmospheric diffusion.* third edition. Ellis Horwood Limited, Publishers Chichester, 1983.

24. Rashki, A. *Seasonality and mineral, chemical and optical properties of dust storms in the Sistan region of Iran and their influence on human health.* Faculty of natural and agricultural sciences university of Pretoria, PhD thesis, 2012.

25. Rashki, A., Kaskaoutis, D.G., Francois, P., Kosmopoulos, P.G. and Legrand, M. *Dust storms and their horizontal dust loading in the Sistan region, Iran.* Aeolian Res. 5 (2012), 51–62

26. Rashki, A., Kaskaoutis, D.G., Rautenbach C.J. de W., Eriksson, P.G., M. Qiang, M. and Gupta, P. *Dust storms and their horizontal dust loading in the Sistan region, Iran: seasonality, transport characteristics and affected areas*. Aeolian Res. 16 (2015), 35–48.

27. Rex, J. and Rohn, J. *Sufficient conditions for regularity and singularity of interval matrices*, SIAM J. Matrix Anal. Appl. 20 (2) (1998), 437–445.

28. Rohn, J. *Cheap and tight bound: The recent result by E. Hansen can be made more efficient*, Interval Comput. 4(1993),13–21.

29. Rohn, J. *Forty necessary and sufficient conditions for regularity of interval matrices: A survey*, Electron. J. Linear Algebra 18 (2009), 500–512.

30. Tong, S.C. *Interval number, fuzzy number linear programming*. Fuzzy Sets Syst., 66 (1994), 301–306.

31. Wang, X. and Huang, G. *Violation analysis on two-step method for interval linear programming* Inf. Sci. 281 (2014), 85–96.

32. Xuan, J., Sokolik, I.N., Hao, J., Guo., F., Mao, H. and Yang, G. *Identification and characterization of sources of atmospheric mineral dust in east Asia*, Atmos. Environ. 38(36) (2004), 6239–6252.

33. Zhou, F., Huang, G.H., Chen, G. and Guo, H. *Enhanced interval linear programming*, Eur. J. Oper. Res. 199 (2009), 323–333.

Aims and scope

Iranian Journal of Numerical Analysis and Optimization (IJNAO) is published twice a year by the Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad. Papers dealing with different aspects of numerical analysis and optimization, theories and their applications in engineering and industry are considered for publication.

Journal Policy

After receiving an article, the editorial committee will assign referees. Refereeing process can be followed via the web site of the Journal.
The manuscripts are accepted for review with the understanding that the work has not been published and also it is not under consideration for publication by any other journal. All submissions should be accompanied by a written declaration signed by the author(s) that the paper has not been published before and has not been submitted for consideration elsewhere.

Instruction for Authors

The Journal publishes all papers in the fields of numerical analysis and optimization. Articles must be written in English.
All submitted papers will be refereed and the authors may be asked to revise their manuscripts according to the referee's reports. The Editorial Board of the Journal keeps the right to accept or reject the papers for publication.
The papers with more than one authors, should determine the corresponding author. The e-mail address of the corresponding author must appear at the end of the manuscript or as a footnote of the first page.
It is strongly recommended to set up the manuscript by Latex or Tex, using the template provided in the web site of the Journal. Manuscripts should be typed double-spaced with wide margins to provide enough room for editorial remarks.
References should be arranged in alphabetical order by the surname of the first author as examples below:
[1] Stoer, J. and Bulirsch, R. *Introduction to Numerical Analysis*, Springer-verlag, New York, 2002.
[2] Brunner, H. *A survey of recent advances in the numerical treatment of Volterra integral and integro-differential equations*, J. Comput. Appl. Math. 8 (1982), 213-229.

Submission of Manuscripts

Authors may submit their manuscripts by either of the following ways:

a) Online submission (pdf or dvi files) via the website of the Journal at:

http://ijnao.um.ac.ir

b) Via journal's email mjms@um.ac.ir

Copyright Agreement

Upon the acceptance of an article by the Journal, the corresponding author will be asked to sign a "Copyright Transfer Agreement" (see the web site) and send it to the Journal address. This will permit the publisher to publish and distribute the work.

Iranian Journal of Numerical Analysis and Optimization

Former : MRJMS

CONTENTS Vol. 9, No.2, pp 1-230, 2019

web site : http://ijnao.um.ac.ir

Email : ijnao@um.ac.ir

ISSN-Print : 2423-6977

ISSN-Online : 2423-6969