



# *Iranian Journal of Numerical Analysis and Optimization*

**Volume 13, Number 1**

**March 2023**

Serial Number: 24

*Ferdowsi University of Mashhad, Iran*

In the Name of God

**Iranian Journal of Numerical Analysis and Optimization (IJNAO)**

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

**Volume 13, Number 1, March 2023**

**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](mailto:ijnao@um.ac.ir)

**Website:** <http://ijnao.um.ac.ir>

**This journal is indexed by:**

- [SCOPUS](#)
- [ZbMATH Open](#)
- [ISC](#)
- [DOAJ](#)
- [SID](#)
- [Civilica](#)
- [Magiran](#)
- [Mendeley](#)
- [Academia.edu](#)
- [Linkedin](#)

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

# Iranian Journal of Numerical Analysis and Optimization

Volume 13, Number 1, March 2023

Ferdowsi University of Mashhad - Iran

©2023 All rights reserved. Iranian Journal of Numerical Analysis and Optimization

# Iranian Journal of Numerical Analysis and Optimization

## Director

M. H. Farahi

## Editor-in-Chief

Ali R. Soheili

## Managing Editor

M. Gachpazan

## EDITORIAL BOARD

### Abbasbandi, Saeid\*

(Numerical Analysis)

Imam Khomeini International University,  
Iran.

e-mail: abbasbandy@ikiu.ac.ir

### Abdi, Ali\*

(Numerical Analysis)

University of Tabriz, Iran.

e-mail: a\_abdi@tabrizu.ac.ir

### Area, Iván\*

(Numerical Analysis)

Universidade de Vigo, Spain.

e-mail: area@uvigo.es

### Babaie Kafaki, Saman\*

(Optimization)

Semnan University, Iran.

e-mail: sbk@semnan.ac.ir

### Babolian, Esmail\*

(Numerical Analysis)

Kharazmi University, Iran.

e-mail: babolian@khu.ac.ir

### Cardone, Angelamaria\*

(Numerical Analysis)

Università degli Studi di Salerno, Italy.

e-mail: ancardone@unisa.it

### Dehghan, Mehdi\*

(Numerical Analysis)

Amirkabir University of Technology, Iran.

e-mail: mdehghan@aut.ac.ir

### Effati, Sohrab\*

(Optimal Control & Optimization)

Ferdowsi University of Mashhad, Iran.

e-mail: s-effati@um.ac.ir

### Emrouznejad, Ali\*

(Operations Research)

Aston University, UK.

e-mail: a.emrouznejad@aston.ac.uk

### Farahi, Mohammad Hadi\*

(Optimal Control & Optimization)

Ferdowsi University of Mashhad, Iran.

e-mail: farahi@um.ac.ir



**Gachpazan, Mortaza\*\***

(Numerical Analysis)

Ferdowsi University of Mashhad, Iran.

e-mail: gachpazan@um.ac.ir

**Ghanbari, Reza\*\***

(Operations Research)

Ferdowsi University of Mashhad, Iran.

e-mail: rghanbari@um.ac.ir

**Hadizadeh Yazdi, Mahmoud\*\***

(Numerical Analysis)

Khaje-Nassir-Toosi University of

Technology, Iran.

e-mail: hadizadeh@kntu.ac.ir

**Hojjati, Gholamreza\***

(Numerical Analysis)

University of Tabriz, Iran.

e-mail: ghobjati@tabrizu.ac.ir

**Hong, Jialin\***

(Scientific Computing )

Chinese Academy of Sciences (CAS),  
China.

e-mail: hjl@lsec.cc.ac.cn

**Karimi, Hamid Reza\***

(Control)

Politecnico di Milano, Italy.

e-mail: hamidreza.karimi@polimi.it

**Khojasteh Salkuyeh, Davod\***

(Numerical Analysis)

University of Guilan, Iran.

e-mail: khojasteh@guilan.ac.ir

**Lohmander, Peter\***

(Optimization)

Swedish University of Agricultural Sci-  
ences, Sweden.

e-mail: Peter@Lohmander.com

**Lopez-Ruiz, Ricardo\*\***

(Complexity, nonlinear models)

University of Zaragoza, Spain.

e-mail: rilopez@unizar.es

**Mahdavi-Amiri, Nezam\***

(Optimization)

Sharif University of Technology, Iran.

e-mail: nezamm@sina.sharif.edu

**Mirzaei, Davoud\***

(Numerical Analysis)

University of Uppsala, Sweden.

e-mail: davoud.mirzaei@it.uu.se

**Omrani, Khaled\***

(Numerical Analysis)

University of Tunis El Manar, Tunisia.

khaled.omrani@issatso.rnu.tn

**Salehi Fathabadi, Hasan\***

(Operations Research )

University of Tehran, Iran.

e-mail: hsalehi@ut.ac.ir

**Soheili, Ali Reza\***

(Numerical Analysis)

Ferdowsi University of Mashhad, Iran.

e-mail: soheili@um.ac.ir

**Soleimani Damaneh, Majid\***

(Operations Research and Optimization,  
Finance, and Machine Learning)

University of Tehran, Iran.

e-mail: m.soleimani.d@ut.ac.ir

**Toutounian, Faezeh\***

(Numerical Analysis)

Ferdowsi University of Mashhad, Iran.

e-mail: toutouni@um.ac.ir

**Türkyılmazoğlu, Mustafa\***

(Applied Mathematics )

Hacettepe University, Turkey.

e-mail: turkyilm@hacettepe.edu.tr

**Vahidian Kamyad, Ali\***

(Optimal Control & Optimization)

Ferdowsi University of Mashhad, Iran.

e-mail: vahidian@um.ac.ir

**Xu, Zeshui\***

(Decision Making)

Sichuan University, China.

e-mail: xuzeshui@263.net

**Vasagh, Zohreh**

(English Text Editor)

Ferdowsi University of Mashhad, Iran.

---

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

\* Full Professor

\*\* Associate Professor

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

## Letter from the Editor-in-Chief

I would like to welcome you to the Iranian Journal of Numerical Analysis and Optimization (IJNAO). This journal has been published two issues per year and supported by the Faculty of Mathematical Sciences at the Ferdowsi University of Mashhad. The faculty of Mathematical Sciences with the centers of excellence and the 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 the 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.

We would like to inform all readers that the Iranian Journal of Numerical Analysis and Optimization (IJNAO), has changed its publishing frequency from "Semiannual" to a "Quarterly" journal since January 2023. The four journal issues per year will be published in the months of March, June, September, and December. One of our goals is to continue to improve the speed of both the review and publication processes, while try continuing to publish the best available international research in numerical analysis and optimization, with the high scientific and publication standards that the journal is known for.

I am also proud to announce that the following professors have accepted the journal's invitation to join the Editorial Board of IJNAO in 2023

- 1- Professor Hamid Reza Karimi, Politecnico di Milano, Italy
- 2- Professor Davoud Mirzaei, University of Uppsala, Sweden
- 3- Professor Saman Babaie Kafaki, Semnan University, Iran
- 4- Professor Khaled Omrani, University of Tunis El Manar, Tunisia
- 5- Professor Angelamaria Cardone, Università degli Studi di Salerno, Italy
- 6- Professor Ali Abdi, University of Tabriz, Iran.

Ali R. Soheili

Editor-in-Chief

## Contents

<b>Applying the meshless Fragile Points method to solve the two-dimensional linear Schrödinger equation on arbitrary domains . . . . .</b>	<b>1</b>
D. Haghighi, S. Abbasbandy and E. Shivanian	
<b>Finding an efficient machine learning predictor for lesser liquid credit default swaps in equity markets . . . . .</b>	<b>19</b>
F. Soleymani	
<b>A modified Liu-Storey scheme for nonlinear systems with an application to image recovery . . . . .</b>	<b>38</b>
A.I. Kiri, M.Y. Waziri and K. Ahmed	
<b>An improvised technique of quintic hermite splines to discretize generalized Burgers–Huxley type equations . . . . .</b>	<b>59</b>
I. Kaur, S. Arora and I. Bala	
<b>Generalization of equitable efficiency in multiobjective optimization problems by the direct sum of matrices . . . . .</b>	<b>80</b>
F. Ahmadi, A. R. Salajegheh and D. Foroutannia	
<b>A family of eight-order interval methods for computing rigorous bounds to the solution to nonlinear equations . . . . .</b>	<b>102</b>
M. Dehghani-Madiseh	
<b>Numerical method for solving fractional Sturm–Liouville eigenvalue problems of order two using Genocchi polynomials</b>	<b>121</b>
A. Aghazadeh, Y. Mahmoudi and F. Dastmalchi Saei	
<b>Impact of inclination angle on thermo-bioconvection of nanofluid containing gyrotactic microorganisms saturated in porous square cavity . . . . .</b>	<b>141</b>
J. Bodduna, C.S. Balla and M.P. Mallesh	





# Applying the meshless Fragile Points method to solve the two-dimensional linear Schrödinger equation on arbitrary domains

D. Haghighi, S. Abbasbandy\* and E. Shivanian

## Abstract

The meshless Fragile Points method (FPM) is applied to find the numerical solutions of the Schrödinger equation on arbitrary domains. This method is based on Galerkin's weak-form formulation, and the generalized finite difference method has been used to obtain the test and trial functions. For partitioning the problem domain into subdomains, Voronoi diagram has been applied. These functions are simple, local, and discontinuous polynomials. Because of the discontinuity of test and trial functions, FPM may be inconsistent. To deal with these inconsistencies, we use numerical flux corrections. Finally, numerical results are presented for some examples of domains with different geometric shapes to demonstrate accuracy, reliability, and efficiency.

**AMS subject classifications (2020):** 35J10, 65M99, 65M20, 65N99.

**Keywords:** Fragile Points Method; Numerical Fluxes; Schrödinger equation; Voronoi Diagram.

---

\*Corresponding author

Received 5 October 2021; revised 20 February 2022; accepted 6 March 2022

Donya Haghighi

Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, 34149-16818, Iran. e-mail: [haghighi.donya@edu.ikiu.ac.ir](mailto:haghighi.donya@edu.ikiu.ac.ir)

Saeid Abbasbandy

Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, 34149-16818, Iran. e-mail: [abbasbandy@sci.ikiu.ac.ir](mailto:abbasbandy@sci.ikiu.ac.ir), [abbasbandy@yahoo.com](mailto:abbasbandy@yahoo.com).

Elyas Shivanian

Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, 34149-16818, Iran. e-mail: [e\\_shivanian@yahoo.com](mailto:e_shivanian@yahoo.com)

## 1 Introduction

Numerical methods are mainly used to solve partial differential equations and have been studied, for example, the finite element method [1], finite volume method [4], and boundary element method [13] to discretize the spatial dimension can be mentioned. In these methods, the accuracy of the method may be affected by deforming the elements or meshes. Therefore, meshless methods such as element free Galerkin [3] and meshless local Petrov–Galerkin [2] were considered. In these methods, the trial and test functions must be continuous, and usually, the trial functions in these methods are complicated. Dong et al. [6] introduced a new meshless method in which test and trial functions are considered as simple, local, and discontinuous polynomials. Very recently, this method has been used to solve the two-dimensional hyperbolic telegraph equation [8]. This new method is called the Fragile points method (FPM), which we will study in this article. This method is also used for complex and irregular domains, which are discussed in this study.

Solving the Schrödinger equation is very important in quantum dynamic calculations, and it has received a lot of attention as a model that describes several important chemical and physical phenomena [11]. This equation is derived from the vector wave equation for the electric field, which governs the propagation of electromagnetic waves in an inhomogeneous medium [10]. Schrödinger equations are also applicable in underwater acoustics, optics, and the design of optoelectronic devices [5].

We consider the two-dimensional time-dependent Schrödinger equation with the form

$$-i\frac{\partial u}{\partial t}(\mathbf{x}, t) = \nabla^2 u(\mathbf{x}, t) + w(\mathbf{x})u(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad (1)$$

with initial conditions

$$u(\mathbf{x}, 0) = g(\mathbf{x}), \quad (2)$$

and the boundary conditions

$$u(\mathbf{x}, t) = h_1(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_D, \quad \nabla u \cdot \mathbf{n}(\mathbf{x}, t) = h_2(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_N. \quad (3)$$

In the above equation,  $w(\mathbf{x})$  is an arbitrary potential function.

In the rest of this paper, in Section 2, the process of obtaining test and trial functions is described. In Section 3, the implementation of numerical flux corrections is given. Some numerical results and examples are provided in Section 4. Finally, the conclusions reached from using the FPM for the two-dimensional linear Schrödinger equation are expressed in Section 5.

## 2 Polynomial discontinuous trial and test functions

In this section, we describe the process of obtaining local, simple, discontinuous, polynomial, point-based trial and test functions. In order to divide the domain into subdomains, we consider several scattered points in the domain  $\Omega$  and its boundary  $\partial\Omega$ . This subdivision should be such that each subdomain contains only one point. Domain partitioning can be done in several ways, for example, the Voronoi diagram partition, quadrilateral and triangular partition (in 2D), tetrahedron and hexahedron partition (in 3D), and so on. In this study, the Voronoi diagram method has been selected. In the present FPM, only nonuniform or uniform points inside and on the domain boundary are applied, and it is a meshless method.

The trial function  $u_h$  in the subdomain  $E_0$  that includes the point  $P_0$ , can be written as

$$u_h(\mathbf{x}, t) = u_0(\mathbf{x}, t) + (\mathbf{x} - \mathbf{x}_0) \nabla u(\mathbf{x}, t)|_{P_0}, \quad \mathbf{x} \in E_0. \quad (4)$$

In the above equation,  $u_0$  is the value of  $u_h$  at  $P_0$  and  $\mathbf{x}_0$  denotes the coordinate of the point  $P_0$ .

The gradient of  $\nabla u$  at  $P_0$  is yet unknown. We employ the generalized finite difference method to calculate  $\nabla u$  at  $P_0$  in terms of the values of  $u_h$  at several neighboring points of  $P_0$ . We name these neighboring points as  $q_1, q_2, \dots, q_m$ . In the following, to calculate the amount of the gradient of  $\nabla u$  at  $P_0$ , we minimize a weighted discrete  $L^2$  norm  $\mathbf{J}$  so that

$$\mathbf{J} = \sum_{i=0}^m \left( \nabla u|_{P_0} \cdot (\mathbf{x}_i - \mathbf{x}_0)^T - (u_i - u_0) \right)^2 w_i, \quad (5)$$

where  $w_i$  denotes the value of weight function at  $q_i$ ,  $\mathbf{x}_i$  is the coordinate vector of  $q_i$ , and  $u_i$  is the value of  $u_h$  at  $q_i$  ( $i = 1, 2, \dots, m$ ). For convenience, we assume that  $w$  is constant. Due to the stationarity of  $\mathbf{J}$ , we have

$$\nabla u = (A^T A)^{-1} A^T (\mathbf{u}_m - u_0 \mathbf{I}_m), \quad (6)$$

where

$$A = \begin{bmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \\ \vdots & \vdots \\ x_m - x_0 & y_m - y_0 \end{bmatrix}, \quad \mathbf{u}_m = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad \mathbf{I}_m = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{m \times 1}.$$

Also equation (6) can be expressed at point  $P_0$  as follows:

$$\nabla u = \mathbf{B} \mathbf{u}_E, \quad (7)$$



where

$$\mathbf{B} = (A^T A)^{-1} A^T \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -1 & 0 & \dots & 0 & 1 \end{bmatrix}_{m \times (m+1)}, \quad \mathbf{u}_E = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_m \end{bmatrix}.$$

Also by substituting (7) into (4), the relation between  $u_h$  and  $\mathbf{u}_E$  will be obtained as

$$u_h = \mathbf{N} \mathbf{u}_E, \quad \text{for all } \mathbf{x} \in E_0, \quad \mathbf{N} = [\mathbf{x} - \mathbf{x}_0] \mathbf{B} + [1, 0, \dots, 0]_{1 \times (m+1)}. \quad (8)$$

### 3 Numerical flux corrections

We can rewrite Schrödinger equation (1)–(3) using mixed form as follows:

$$\begin{cases} \sigma(\mathbf{x}, t) = \nabla u(\mathbf{x}, t), & \text{in } \Omega, \\ -\nabla \cdot \sigma(\mathbf{x}, t) = i \frac{\partial u}{\partial t}(\mathbf{x}, t) + w(\mathbf{x}) u(\mathbf{x}, t), & \text{in } \Omega, \\ u(\mathbf{x}, t) = h_1(\mathbf{x}, t), & \text{in } \Gamma_D, \\ \sigma \cdot \mathbf{n}(\mathbf{x}, t) = h_2(\mathbf{x}, t), & \text{in } \Gamma_N. \end{cases} \quad (9)$$

By multiplying the first and second equations in (9) by the test functions  $\tau$  and  $\nu$ , respectively, and integrating it on the subdomain  $E$ ,

$$\int_E \sigma_h \cdot \tau d\Omega = \int_E \nabla u_h(\mathbf{x}, t) \cdot \tau d\Omega, \quad (10)$$

$$\int_E -\nabla \cdot \sigma \nu d\Omega = i \int_E \frac{\partial u}{\partial t}(\mathbf{x}, t) \nu d\Omega + \int_E w(\mathbf{x}) u(\mathbf{x}, t) \nu d\Omega, \quad (11)$$

using the Green formula and by summing these equations over all subdomains, we have

$$\int_{\Omega} \sigma_h \cdot \tau d\Omega = - \int_{\Omega} u_h \nabla \cdot \tau d\Omega + \sum_{E \in \Omega} \int_{\partial E} \hat{u}_h \mathbf{n} \cdot \tau d\Gamma, \quad (12)$$

$$\int_{\Omega} \sigma_h \cdot \nabla \nu d\Omega = \sum_{E \in \Omega} \int_{\partial E} \hat{\sigma}_h \cdot \mathbf{n} \nu d\Gamma + i \int_{\Omega} \frac{\partial u}{\partial t}(\mathbf{x}, t) \nu d\Omega + \int_{\Omega} w(\mathbf{x}) u(\mathbf{x}, t) \nu d\Omega. \quad (13)$$

In the above equations, values  $\hat{\sigma}_h$  and  $\hat{u}_h$  represent approximations  $\sigma_h$  and  $u_h$  on  $\partial E$ . These values are named numerical fluxes. To simplify (12) and (13), we define the operators *average* and *jump*, where by these operators,

we can manage the numerical fluxes. As regards,  $\Gamma = \Gamma_h + \Gamma_D + \Gamma_N$  ( $\Gamma_h$  is the set of all internal boundaries of subdomains), using [6, Table 3.1 ], and substituting the interior penalty numerical fluxes, we have

$$\begin{aligned} & \sum_{E \in \Omega} \int_E \nabla u_h \cdot \nabla \nu d\Omega - \sum_{e \in \Gamma_h \cup \Gamma_D} \int_e (\{\nabla u_h\} [\nu] + \{\nabla \nu\} [u_h]) d\Gamma \\ & + \sum_{e \in \Gamma_h \cup \Gamma_D} \frac{\eta}{h_e} \int_e [\nu] [u_h] d\Gamma = \sum_{e \in \Gamma_D} \int_e \left( \frac{\eta}{h_e} \nu - \nabla \nu \cdot \mathbf{n} \right) h_1(\mathbf{x}, t) d\Gamma \\ & + \int_{\Omega} w(\mathbf{x}) u(\mathbf{x}, t) \nu d\Omega + \sum_{e \in \Gamma_N} \int_e \nu h_2(\mathbf{x}, t) d\Gamma \\ & + i \int_{\Omega} \frac{\partial u}{\partial t}(\mathbf{x}, t) \nu d\Omega. \end{aligned}$$

The above equation is the formula of FPM, which is called FPM-primal method.

The method (matrix form) can be expressed as follows:

$$(\mathbf{K} - \mathbf{W})\mathbf{u} - i\mathbf{C}\dot{\mathbf{u}} = \mathbf{F}. \quad (14)$$

Using  $\theta$ -weighted scheme [12], the above equation can be written as follows:

$$(\mathbf{K} - \mathbf{W})(\theta \mathbf{u}^{n+1} + (1 - \theta) \mathbf{u}^n) - i\mathbf{C} \left( \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \right) = \mathbf{F}^n. \quad (15)$$

In (15),  $\mathbf{u}^k(\mathbf{x}) = \mathbf{u}(\mathbf{x}, k\Delta t)$ , where  $\Delta t$  is the time step and  $0 \leq \theta \leq 1$ . By substituting values  $\mathbf{B}$  instead of  $\nabla \nu$  and  $\nabla u$ ,  $\mathbf{N}$  instead of  $u_h$  and  $\nu$  in the formula of FPM-primal, the point stiffness matrices  $\mathbf{K}$ ,  $\mathbf{C}$ ,  $\mathbf{W}$  and also the right vector  $\mathbf{F}$  will be achieved as follows:

$$\mathbf{W} = \int_E \mathbf{N}^T \mathbf{N} w(\mathbf{x}) d\Omega, \quad \mathbf{C} = \int_E \mathbf{N}^T \mathbf{N} d\Omega, \quad \mathbf{K}_E = \int_E \mathbf{B}^T \mathbf{B} d\Omega, \quad E \in \Omega, \quad (16)$$

$$\begin{aligned} \mathbf{K}_h = & \frac{-1}{2} \int_e (\mathbf{B}_1^T \mathbf{n}_1^T \mathbf{N}_1 + \mathbf{N}_1^T \mathbf{n}_1 \mathbf{B}_1) d\Gamma + \frac{\eta}{h_e} \int_e \mathbf{N}_1^T \mathbf{N}_1 d\Gamma \\ & + \frac{-1}{2} \int_e (\mathbf{B}_2^T \mathbf{n}_2^T \mathbf{N}_2 + \mathbf{N}_2^T \mathbf{n}_2 \mathbf{B}_2) d\Gamma + \frac{\eta}{h_e} \int_e \mathbf{N}_2^T \mathbf{N}_2 d\Gamma \\ & + \frac{-1}{2} \int_e (\mathbf{B}_2^T \mathbf{n}_1^T \mathbf{N}_1 + \mathbf{N}_2^T \mathbf{n}_2 \mathbf{B}_1) d\Gamma + \frac{\eta}{h_e} \int_e \mathbf{N}_1^T \mathbf{N}_2 d\Gamma \\ & + \frac{-1}{2} \int_e (\mathbf{B}_1^T \mathbf{n}_2^T \mathbf{N}_2 + \mathbf{N}_1^T \mathbf{n}_1 \mathbf{B}_2) d\Gamma + \frac{\eta}{h_e} \int_e \mathbf{N}_2^T \mathbf{N}_1 d\Gamma, \quad e \in \partial E_1 \cap \partial E_2, \end{aligned}$$

$$\mathbf{K}_D = - \int_e (\mathbf{B}^T \mathbf{n}^T \mathbf{N} + \mathbf{N}^T \mathbf{n} \mathbf{B}) d\Gamma + \frac{\eta}{h_e} \int_e \mathbf{N}^T \mathbf{N} d\Gamma, \quad e \in \Gamma_D, \quad (17)$$

and it can also be written

$$\begin{aligned} \mathbf{F}_N &= \int_e \mathbf{N}^T h_2(\mathbf{x}, t) d\Gamma, & e \in \Gamma_N, \\ \mathbf{F}_D &= \int_e \left( \frac{\eta}{h_e} \mathbf{N}^T - \mathbf{B}^T \mathbf{n} \right) h_1(\mathbf{x}, t) d\Gamma, & e \in \Gamma_D. \end{aligned} \quad (18)$$

## 4 Numerical results

In this section, we will study some numerical examples, and using the results obtained from the application of FPM on these examples, the accuracy and stability of the method are investigated. All examples were done in MATLAB software on a Core i5, 2.67 GHz CPU machine with 4 GB of memory. The relative errors used in this section are defined as follows:

$$r_0 = \frac{\|u_h - u\|_{L^2}}{\|u\|_{L^2}}, \quad r_1 = \frac{\|\nabla u_h - \nabla u\|_{L^2}}{\|\nabla u\|_{L^2}}.$$

Also we calculate the convergence orders in space and time via

$$C - order(space) = \frac{\log 10 \left( \frac{e_1}{e_2} \right)}{\log 10 \left( \frac{h_1}{h_2} \right)}, \quad C - order(time) = \frac{\log 10 \left( \frac{e_1}{e_2} \right)}{\log 10 \left( \frac{\Delta t_1}{\Delta t_2} \right)},$$

such that  $h_1$  and  $\Delta t_1$  correspond to  $e_1$  and also  $h_2$  and  $\Delta t_2$  correspond to error  $e_2$ . In numerical examples, we consider errors  $e_1$  and  $e_2$  as relative error  $r_0$ .

**Example 1.** (a) We first consider (1) with potential function  $w(x, y) = 1 - \frac{2}{x^2} - \frac{2}{y^2}$  and exact solution  $u(x, y, t) = e^{it} x^2 y^2$  in the region  $\Omega = [0, 1] \times [0, 1]$ . Boundary and initial conditions are obtained using the exact solution [7, 11]. In Table 1 relative errors are shown for the number of different points of the domain that are uniformly distributed and all boundary conditions are considered Dirichlet. This table shows the good accuracy and stability of the method, and as the number of points increases, the accuracy of the method improves. In addition, as shown in Figure 1, the relative errors decrease as the number of points increases.

(b) Next, we consider the boundary conditions as follows:

$$u(0, y, t) = u(x, 0, t) = 0, \quad \nabla u \cdot \mathbf{n}(1, y, t) = 2e^{it} y^2, \quad u(x, 1, t) = e^{it} x^2.$$

Table 1: Relative errors of the method for example 1(a) at  $T = 1$  and  $\Delta t = 0.05$  with  $\theta = 0.6$  and uniform points.

h	Number of point	parameters	Errors	CPU time (s)	C-order
0.25	$N = 25$	$h_e = 0.1$	$3.942832 \times 10^{-2}$	0.31	-
		$\eta = 2.5$	$2.523913 \times 10^{-1}$		
0.1	$N = 121$	$h_e = 0.1$	$5.73321 \times 10^{-3}$	0.63	2.1044
		$\eta = 4$	$3.52258 \times 10^{-2}$		
0.04	$N = 676$	$h_e = 0.1$	$9.91802 \times 10^{-4}$	9.34	1.9148
		$\eta = 9$	$9.26574 \times 10^{-3}$		
0.02	$N = 2601$	$h_e = 0.1$	$4.30894 \times 10^{-4}$	163s	1.2027
		$\eta = 9$	$4.27138 \times 10^{-3}$		

In Table 2, relative errors have been reported for the number of different points that are uniformly and nonuniformly distributed over the domain. Comparing the results, we find that in this example, how the points are distributed does not have much effect on the accuracy of the method. Also Figure 2 shows error plots for  $N = 676$  uniform and nonuniform points. Figures 3 and 4 demonstrate the appropriate accuracy of the FPM for different times for  $x = 0.6$ ,  $\theta = 0.51$ ,  $\Delta t = 0.01$ ,  $N = 121$ ,  $h_e = 0.1$ , and  $\eta = 4$ . Therefore, according to these results, the method has appropriate accuracy for different boundary conditions and is also convergent. In addition, as the final time increases, accuracy is maintained and FPM is stable.

Compared to [15] and [16], the proposed method achieves almost the same accuracy in much less time.

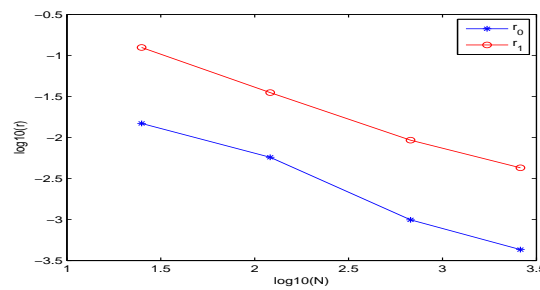
Figure 1: Relative errors for Example 1 at  $T = 1$  and  $\Delta t = 0.05$ .

Table 2: Relative errors of the method for Example 1(b) at  $T = 1$ ,  $\theta = 0.51$  and  $\Delta t = 0.01$  for points with uniform and nonuniform distribution.

uniform points				
h	Number of points	Errors	CPU time (s)	C-order
0.25	$N = 25$	$6.476626 \times 10^{-2}$ $2.636926 \times 10^{-1}$	0.40	-
0.1	$N = 121$	$5.763461 \times 10^{-3}$ $3.419285 \times 10^{-2}$	1.08	2.1864
0.04	$N = 676$	$2.552031 \times 10^{-3}$ $1.216389 \times 10^{-2}$	27.9	0.8891
nonuniform points				
	Number of points	Errors	CPU time (s)	
	$N = 25$	$4.273177 \times 10^{-2}$ $1.484748 \times 10^{-1}$	0.33	
	$N = 121$	$9.238729 \times 10^{-3}$ $9.161839 \times 10^{-2}$	0.94	
	$N = 676$	$2.423225 \times 10^{-3}$ $1.216389 \times 10^{-2}$	27.7	

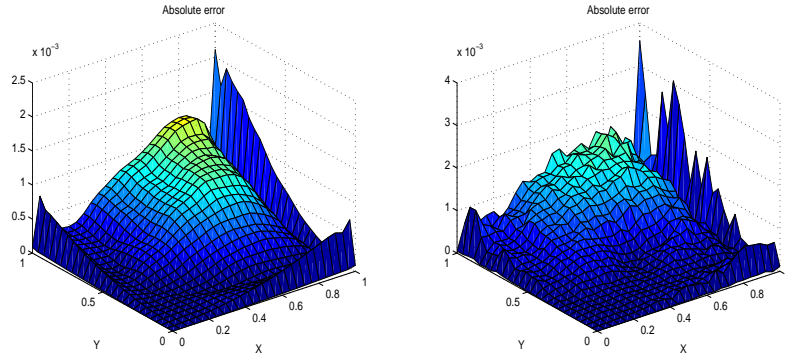


Figure 2: Comparison the absolute errors related to Example 1(b) for  $T = 1$ ,  $\Delta t = 0.01$ ,  $\theta = 0.51$ , and  $N = 676$  for uniform and nonuniform points.

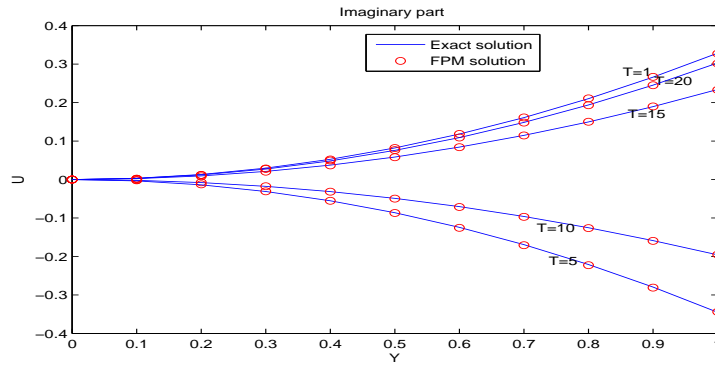


Figure 3: Comparison the imaginary parts of exact and numerical solutions related to Example 1 for  $x = 0.6$ ,  $\Delta t = 0.01$ , and  $N = 121$ .

**Example 2.** In this example, we consider (1) with  $N = 676$  uniform points in the domain  $\Omega = [0, 1] \times [0, 1]$  such that

$$w(x, y) = -\frac{4x^2 + 4y^2 - 4x - 4y + \beta^2 - 4\beta + 2}{\beta^2},$$

$$u(x, y, t) = \exp\left(-\frac{(x - 0.5)^2}{\beta} - \frac{(y - 0.5)^2}{\beta} - it\right).$$

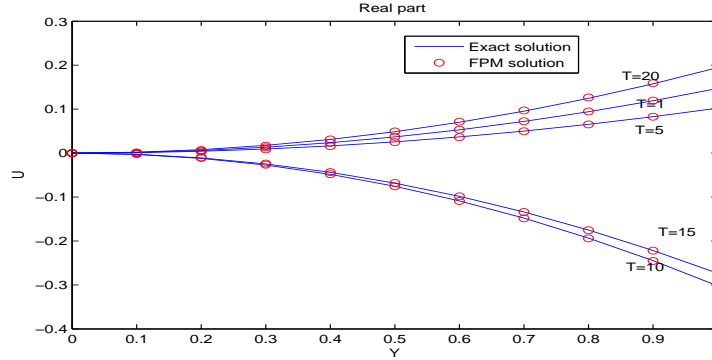


Figure 4: Comparison the real parts of exact and numerical solutions related to Example 1 for  $x = 0.6$ ,  $\Delta t = 0.01$  and  $N = 121$ .

Table 3: Relative errors of the method for Example 2 with  $h_e = 0.1$ ,  $\theta = 0.52$ ,  $\eta = 11$ , and  $\Delta t = 0.01$ .

Final time	$r_0$	$r_1$	CPU time (s)
$T = 1$	$1.85272 \times 10^{-4}$	$1.18535 \times 10^{-2}$	28.3
$T = 5$	$3.84086 \times 10^{-4}$	$2.00094 \times 10^{-2}$	99.5
$T = 10$	$5.48035 \times 10^{-4}$	$2.10124 \times 10^{-2}$	201.3
$T = 15$	$5.00376 \times 10^{-4}$	$2.07114 \times 10^{-2}$	281.5

With Dirichlet boundary conditions for  $\Delta t = 0.01$ ,  $\theta = 0.52$ , and  $\beta = 2$ , relative errors related to different final times are shown in Table 3. This table shows the stability of the method over time. In the following, Figures 5 and 6 show the plots of imaginary and real parts of numerical and exact solutions for  $N = 2601$  uniform points with  $h_e = 0.1$  and  $\eta = 11$ . These figures indicate that the method is also accurate for a large number of points. Also, the plot of errors for  $N = 676$  uniform and nonuniform points is provided in Figure 7. As this figure shows, under similar conditions, the error of the proposed method is less for points with a uniform distribution.

**Example 3.** a) Now we solve the previous example in an L-shaped domain that has the following boundaries:

$$\Omega_1 = \{0\} \times [0, 1], \quad \Omega_2 = \{0.36\} \times [0.52, 1],$$

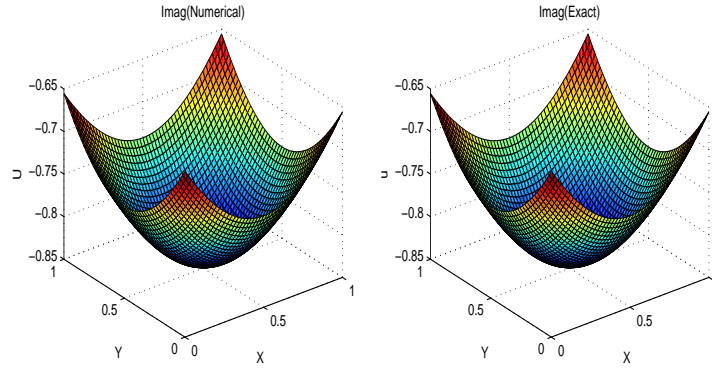


Figure 5: Comparison of imaginary parts of the numerical and exact solutions for Example 2 with  $\eta = 11$ ,  $h_e = 0.1$ ,  $\theta = 0.52$ ,  $\Delta t = 0.01$ ,  $N = 2601$ , and  $T = 1$ .

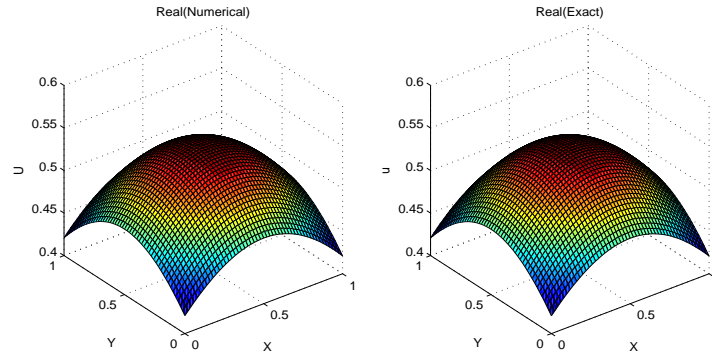


Figure 6: Comparison of real parts of the numerical and exact solutions for Example 2 with  $\eta = 11$ ,  $h_e = 0.1$ ,  $\theta = 0.52$ ,  $\Delta t = 0.01$ ,  $N = 2601$ , and  $T = 1$ .

$$\Omega_3 = [0, 1] \times \{0\}, \quad \Omega_4 = [0.36, 1] \times \{0.52\}.$$

Figure 8 shows the uniform distribution of  $N = 484$  points in this domain. If we consider the boundary conditions completely Dirichlet, then we have Table 4 for the number of different points of the domain that are uniformly selected. As this table shows, according to the number of points, the numerical results have good accuracy that is obtained in a short time. Figure 9 shows the relation between the distance between points with uniform distributions with relative errors.

b) Next, we consider a circular domain as follows:

$$\Omega = \left\{ (x, y) \in \mathbb{R}^2 : \sqrt{x^2 + y^2} \leq 1 \right\},$$



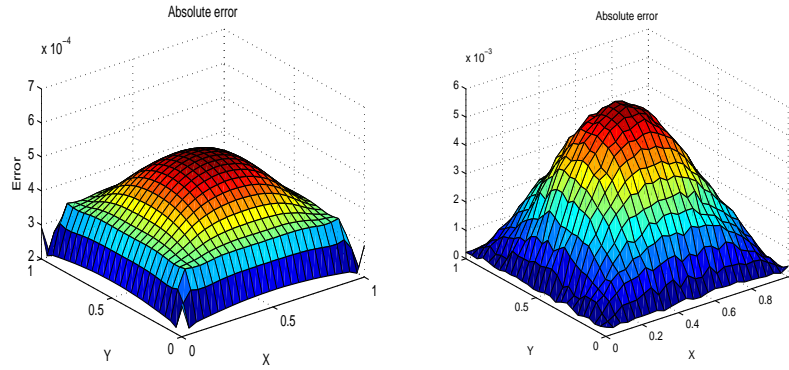


Figure 7: Plot of error for Example 2 based on uniform points for  $\theta = 0.52$   $N = 676$ ,  $T = 1$ , and  $\Delta t = 0.01$ .

and the equation is solved by FPM. Table 5 shows the relative error values and convergence orders over time. According to this table, there is no need to reduce the time step too much, because making it smaller does not have much effect on accuracy, and we should improve the accuracy by changing other parameters or the number of points. Also, due to the circular amplitude and the number of points used, the relative errors obtained are acceptable and are obtained in a short time.

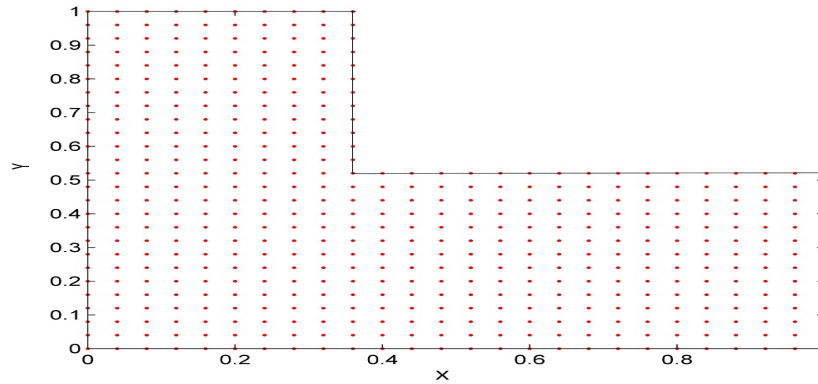
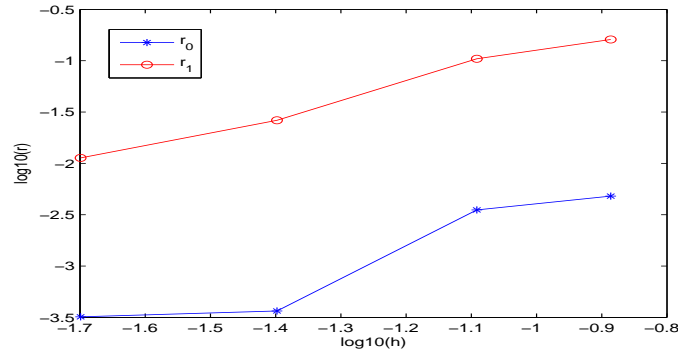


Figure 8: L-shaped domain related to Example 3(a) with  $N = 484$  uniform points.

**Example 4.** In this example, we consider the time-dependent Schrödinger (1)–(3) in  $(x, y) \in \Omega = [0, \pi] \times [0, \pi]$  with initial condition  $u(x, y, 0) =$

Table 4: Relative errors of the method for Example 3(a) at  $T = 1$  and  $\Delta t = 0.01$ ,  $\theta = 0.65$ .

Number of points	parameters	Errors	CPU time(s)
$N = 49$	$h_e = 1$	$r_0 = 4.796603 \times 10^{-3}$	0.69
	$\eta = 75$	$r_1 = 1.608921 \times 10^{-1}$	
$N = 121$	$h_e = 0.001$	$r_0 = 3.516616 \times 10^{-3}$	1.20
	$\eta = 190$	$r_1 = 1.044712 \times 10^{-1}$	
$N = 484$	$h_e = 0.1$	$r_0 = 3.649402 \times 10^{-4}$	12.60
	$\eta = 75$	$r_1 = 2.631145 \times 10^{-2}$	
$N = 1849$	$h_e = 0.01$	$r_0 = 3.196693 \times 10^{-4}$	280.72
	$\eta = 27$	$r_1 = 1.131135 \times 10^{-2}$	

Figure 9: Error curves with respect to the distance of selected points from the domain to each other for example 3(a) with  $\Delta t = 0.01$  and  $T = 1$ s.Table 5: Relative error values and convergence orders over time for example 3(b) with  $N = 529$  uniform points,  $T = 1$ ,  $\theta = 0.65$ ,  $h_e = 0.01$ , and  $\eta = 16$ .

Time step	$r_0$	$r_1$	CPU time(s)	C-order
$\Delta t = 0.08$	$1.261254 \times 10^{-2}$	$6.019296 \times 10^{-2}$	6.88	-
$\Delta t = 0.04$	$7.115889 \times 10^{-3}$	$4.330438 \times 10^{-2}$	9.04	0.8257
$\Delta t = 0.02$	$3.882812 \times 10^{-3}$	$3.440278 \times 10^{-2}$	13.67	0.8739
$\Delta t = 0.01$	$3.042351 \times 10^{-3}$	$3.105425 \times 10^{-2}$	23.26	0.3519

$\sin(x)\sin(y)$  and Dirichlet boundary conditions that are zero on all sides, with the given potential function as  $w(x, y) = 3, (x, y) \in \Omega$ . The analytical solution is as  $u(x, y, t) = e^{it} \sin(x) \sin(y)$ .

As you can see in Table 6, for a number of different points, the method has the appropriate accuracy to solve this example. Also, Figures 10 and 11 demonstrate the plots for  $h_e = 0, 1$ ,  $\eta = 3.8$ , and  $N = 676$  nonuniform points for  $T = 1s$  and  $T = 3s$ , respectively. These figures show the accuracy of FPM for the case where the points are considered nonuniform. Compared to [9], the proposed method reports better computational times and accuracy.

Table 6: Relative errors of the method for Example 4 at  $T = 1$ ,  $\theta = 0.52$ , and  $\Delta t = 0.01$ .

h	Number of points	parameters	Errors	CPU time (s)	C-order
0.25	$N = 25$	$h_e = 0.1$	$1.797325 \times 10^{-2}$	0.41	-
		$\eta = 1$	$1.185351 \times 10^{-1}$		
0.1	$N = 121$	$h_e = 0.1$	$1.89251 \times 10^{-3}$	0.97	2.4566
		$\eta = 2.7$	$1.92942 \times 10^{-2}$		
0.04	$N = 676$	$h_e = 0.1$	$6.002710 \times 10^{-4}$	9.47	1.2532
		$\eta = 3.8$	$4.150509 \times 10^{-3}$		
0.02	$N = 2601$	$h_e = 0.1$	$4.561287 \times 10^{-4}$	107	0.3962
		$\eta = 2$	$6.729947 \times 10^{-3}$		

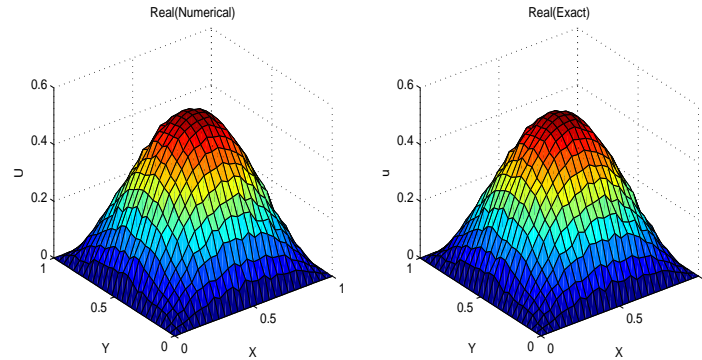


Figure 10: Plots for real parts of the numerical and exact solutions related to Example 4 for  $dt = 0.01$ ,  $N = 676$ ,  $T = 1$ ,  $\theta = 0.54$ ,  $\eta = 38$ , and  $h_e = 1$

**Example 5.** For the last example, we consider Schrödinger equation with the following exact solution and initial conditions:

$$u(x, y, t) = e^{(-it)} (\sin(x) + \cos(y)), \quad u(x, y, 0) = (\sin(x) + \cos(y)).$$

This equation is solved using the proposed method on the connected amoeba-like domain according to Figure 12 with  $N = 100$  points that are nonuniformly distributed in the domain. Table 7 shows relative errors and CPU

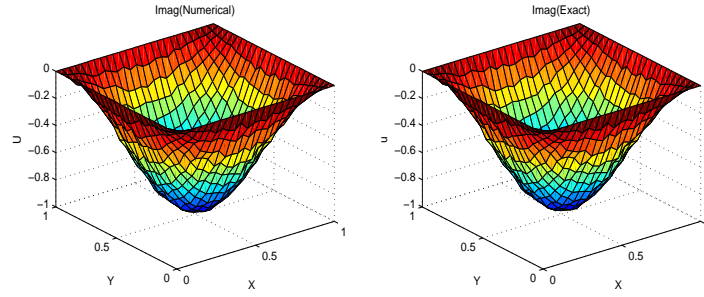


Figure 11: Plots for imaginary parts of the numerical and exact solutions related to Example 4 for  $\Delta t = 0.01$ ,  $N = 676$ ,  $T = 5s$ ,  $\theta = 0.54$ ,  $\eta = 38$ , and  $h_e = 1$ .

time related to different final times. Due to the nonuniform points and the domain of the problem in this example, the accuracy of the method is acceptable. Also, plots related to numerical and exact solutions for  $T = 2s$  are presented in Figure 13, which confirms the suitability of the method for irregular domains.

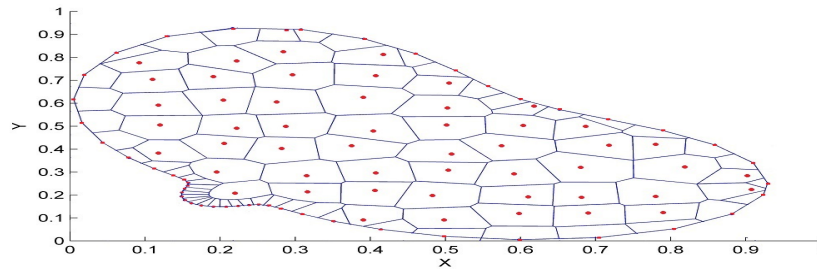


Figure 12: Domain of the problem in Example 5 with  $N = 100$  selected points that are nonuniformly distributed.

Table 7: Relative errors of the method for Example 5 for  $T = 1, 5, 10, 15, 20$ ,  $\Delta t = 0.0097$ , and  $\theta = 0.5$ .

T	Parameters	$r_0$	$r_1$	CPU time (s)
1	$h_e = 0.001$ , $\eta = 215$	$5.697623 \times 10^{-3}$	$9.610123 \times 10^{-2}$	1.06
5	$h_e = 0.001$ , $\eta = 600$	$6.276758 \times 10^{-3}$	$9.613064 \times 10^{-2}$	3.22
10	$h_e = 0.001$ , $\eta = 500$	$6.286391 \times 10^{-3}$	$9.620433 \times 10^{-2}$	6.05
15	$h_e = 0.001$ , $\eta = 550$	$6.282664 \times 10^{-3}$	$9.618269 \times 10^{-2}$	8.69
20	$h_e = 0.001$ , $\eta = 600$	$6.279557 \times 10^{-3}$	$9.616465 \times 10^{-2}$	11.33

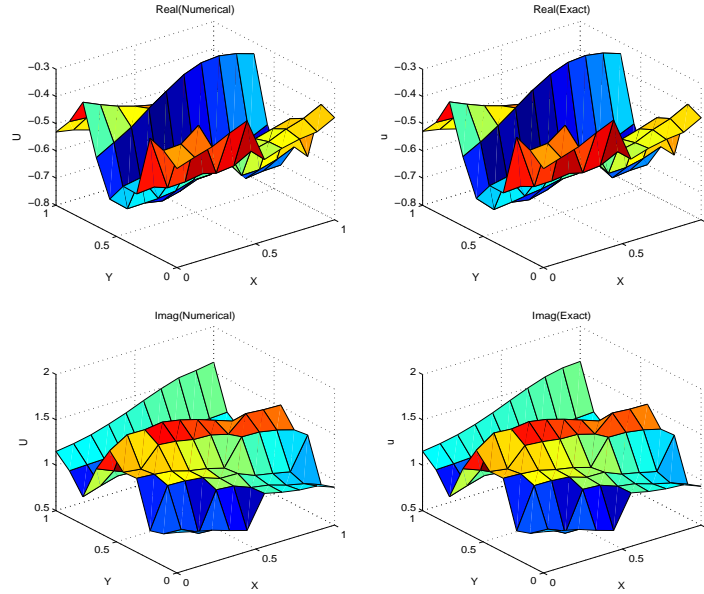


Figure 13: Plots related to numerical and exact solutions for Example 5 with  $\theta = 0.5$ ,  $h_e = 0.001$ ,  $\eta = 50$ ,  $T = 2s$  and  $\Delta t = 0.0097$ .

## 5 Conclusion

In this paper, the meshless Fragile Points Method (FPM) is used to obtain numerical solutions to the two-dimensional linear Schrödinger equation. This method is based on Galerkin's weak form, and the test and trial functions are considered simple, local, and discontinuous polynomials. Numerical flux corrections have been used to deal with inconsistencies due to the discontinuity of trial functions. Finally, the efficiency, stability, and accuracy of the method were evaluated with several numerical examples. In these numerical examples, the accuracy and stability of the method were evaluated both for a large number of points and for the case where the points were selected nonuniformly. We also got good solutions for larger final times and the problems with irregular domains.

According to the results of the tables and comparison of the curves obtained by FPM with the exact curves, it can be seen that the method is stable and has good accuracy. Also, the method does not have much computational cost and depending on the number of points used, it will achieve numerical solutions with good accuracy in a short time, which is an advantage over the finite element. Other advantages of this method over other numerical methods are described in detail in [14, Table 1].

## Acknowledgements

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

## References

- [1] Asadzadeh, M. *An introduction to the finite element method (FEM) for differential equations*, Chalmers: Lecture notes. 2010.
- [2] Atluri, S.N. and Zhu, T. *A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics*, Comput. Mech. 22(2) (1998), 117–127.
- [3] Belytschko, T., Lu, Y.Y. and Gu, L. *Element free Galerkin methods*, Int. J. Numer. Methods. Eng. 37(2) (1994), 229–256.
- [4] Chai, J.C., Lee, H.S. and Patankar, S.V. *Finite volume method for radiation heat transfer*, J. Thermophys. Heat. Trans. 8(3) (1994), 419–425.
- [5] Dehghan, M. and Emami-Naeini, F., *The Sinc-collocation and Sinc-Galerkin methods for solving the two-dimensional Schrödinger equation with nonhomogeneous boundary conditions*, Appl. Math. Mode. 37(22) 2013, 9379–9397.
- [6] Dong, L., Yang, T., Wang, K. and Atluri, S.N. *A new fragile points method (FPM) in computational mechanics, based on the concepts of Point Stiffnesses and Numerical Flux Corrections*, Eng. Anal. Bound. Elem. 107 (2019), 124–133.
- [7] Gao, Z., Xie, S. *Fourth-order alternating direction implicit compact finite difference schemes for two-dimensional Schrödinger equations*, Appl. Numer. Math. 61(4) (2011), 593–614.
- [8] Haghighi, D., Abbasbandy, S., Shivanian, E., Dong, L. and Atluri, S.N. *The fragile points method (FPM) to solve two-dimensional hyperbolic telegraph equation using point stiffness matrices*, Eng. Anal. Bound. Elem., 134 (2022), 11–21.
- [9] Karabaş, N. İ., Korkut, S. Ö., Tanoğlu, G. and Aziz, I. *An efficient approach for solving nonlinear multidimensional Schrödinger equations*, Eng. Anal. Bound. Elem. 132 (2021), 263–270.
- [10] Levy, M. *Parabolic equation methods for electromagnetic wave propagation*, IEE Electromagnetic Waves Series, 45. Institution of Electrical Engineers (IEE), London, 2000.

- [11] Subaşı, M. *On the finite differences schemes for the numerical solution of two dimensional Schrödinger equation*, Numer. Methods Partial Differential Equations 18(6) (2002), 752–758.
- [12] Tian, Z.F. and Yu, P.X. *High-order compact ADI (HOC-ADI) method for solving unsteady 2D Schrödinger equation*, Comput. Phys. Commun. 181(5) (2010), 861–868.
- [13] Wrobel, L.C. *The Boundary Element Method, Volume 1: Applications in Thermo-Fluids and Acoustics*, Vol. 1. John Wiley & Sons. 2002.
- [14] Yang, T., Dong, L. and Atluri, S.N. *A simple Galerkin meshless method, the fragile points method using point stiffness matrices, for 2D linear elastic problems in complex domains with crack and rupture propagation*, Int. J. Numer. Meth. Eng. 122(2) (2021), 348–385.
- [15] Zhang, L.W., Deng, Y.J., Liew, K.M. and Cheng, Y.M. *The improved complex variable element-free Galerkin method for two-dimensional Schrödinger equation*, Comput. Math. with Appl. 68(10) (2014), 1093–1106.
- [16] Zhang, S. and Chen, S. *A meshless symplectic method for two-dimensional Schrödinger equation with radial basis functions*, Comput. Math. with Appl. 72(9) (2016), 2143–2150.

#### How to cite this article

Haghighi, D., Abbasbandy, S. and Shivanian, E., Applying the meshless Fragile Points method to solve the two-dimensional linear Schrödinger equation on arbitrary domains. *Iran. j. numer. anal. optim.*, 2023; 13(1): [1-18](#).  
<https://doi.org/10.22067/ijnao.2022.72900.1063>.



# Finding an efficient machine learning predictor for lesser liquid credit default swaps in equity markets

F. Soleymani<sup></sup>

## Abstract

To solve challenges occurred in the existence of large sets of data, recent improvements of machine learning furnish promising results. Here to propose a tool for predicting lesser liquid credit default swap (CDS) rates in the presence of CDS spreads over a large period of time, we investigate different machine learning techniques and employ several measures such as the root mean square relative error to derive the best technique, which is useful for this type of prediction in finance. It is shown that the nearest neighbor is not only efficient in terms of accuracy but also desirable with respect to the elapsed time for running and deploying on unseen data.

**AMS subject classifications (2020):** Primary 91G80; Secondary 62J05.

**Keywords:** Credit default swap (CDS); machine learning; prediction; liquidity; spread.

## 1 Introduction

### 1.1 Credit default swap

Using credit derivatives, participants of market can transfer credit risk for a portfolio of credits. The most important kind of credit derivative is the credit default swap (CDS) which consists of credit default index swap tranches, credit default index swaps, basket swaps, and swaps with single names, [6,

---

Received 8 November 2021; revised 22 February 2022; accepted 9 March 2022

Fazlollah Soleymani

Department of Mathematics, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan 45137-66731, Iran. e-mails: fazlollah.soleymani@gmail.com, soleymani@iasbs.ac.ir



Chapter 1], [8]. To be more precise, CDS is a bilateral over-the-counter (OTC) derivative contract that enables two counterparties to buy and sell protection on the given reference entity. It inherits the traditional swap format and consists of two legs: a) premium leg and b) severity leg. The protection buyer pays regular fixed premium in return for receiving from the protection seller and the loss payment in case the reference entity defaults, see e.g. [22, Part 6], [26].

CDS is a basic building block for many other derivative contracts and methods. In this way, it is closely linked to debit valuation adjustment (DVA) and credit valuation adjustment (CVA). By assuming CDS contract as a continuous process/observation, it can be defined as follows [20]:

$$\text{CDS} = c \int_0^T \exp(t(-(h+r))) dt - h(1-R) \int_0^T \exp(-ht) \exp(-rt) dt, \quad (1)$$

where  $c$ ,  $r$ ,  $h$ ,  $T$  and  $R$  are the CDS coupon, zero coupon interest rate, hazard rate, maturity of the contract and the recovery rate, respectively. The key driver of the CDS value is the hazard rate  $h = -c/(R-1)$ , which shows that the hazard rate is simply the CDS rate divided by the loss function.

CDS enables counterparties to manage and control credit exposure to a reference credit entity. Under the contract, on one side, the protection buyer (seller) pays (receives) a premium in return for credit loss compensation that is received (paid) when the reference entity defaults. In essence, the CDS is an insurance contract against reference entity default. Over the years, CDS has evolved in many directions and structural variations were proposed to adapt the contract to particular market needs, [4]. For example, the reference obligations can be a single entity, index, and baskets of a few names or larger pools.

To discuss about the applicability, a company proposes a strategic risk indicator, CDS spreads, for its risk dashboard. If the CDS rate, which is the price for insuring against the default of a client, went outside a specified range, then mitigation steps can be performed to deal with the client's increased risk, [12]. Recalling that CDS is basically a form of insurance that the buyer of say, a bond, buys from a financial institution, say a bank, against the bond's going "bad" (not paying in full.)

The CDS spread is the rate of payments that the buyer of the CDS makes to the seller each year. To discuss further, say the value of the bond was \$1,000. A bank might charge an annual amount \$10 per thousand if it felt that the bond had a slightly less than 1% chance of going bad in the coming year (because the \$10 would include a commission.) If the buyer paid \$10 or 1% (the credit default swap rate), the buyer would be purchasing the right to sell back the bond to the bank for \$1,000, no matter what the bond was really worth. This would be the "swap." And its purpose would be to protect against credit default. Note that in practice, CDS contracts pay in regular intervals - typically quarterly.

In addition, quanto CDSs are designated in a given currency to furnish protection when default of a certain entity, [23]. There are some instances, such as for systemically important companies or for sovereign entities, when an investor considers purchasing protection on a currency against the one, at which the reference entity's assets are denominated. It is known that in different currency denominations quanto CDS spreads are differences in CDS premiums of the same reference entity, [23].

## 1.2 Motivation

In this paper, predictive analysis based on machine learning (ML) [25] is discussed for some active groups of liquid CDS rates given for various daily rates, which are then employed to predict lesser liquid CDS. This is the main motivation of the paper since this application can mostly be observed in the equity or credit markets where the factor of liquidity drives specific tools into certain categories, [21], [28, Chapter 9].

## 1.3 ML

ML uses statistical methods to train machines from a given data set. After the learning, the systems produce optimized models that explain the data in the best way and restrict the potential biases, further enabling better assessments and decision making. Thus, such models are also broadly employed for predictions. ML is based on acquiring a habit in terms of learning. In fact, learning is considered as a process of progressive adaptation and the ability to produce the right patterns in response to a given set of inputs, [15, Chapter 6].

Here, an application of ML approach in financial mathematics is discussed, see the book [13, Part 18.8] for some background. It is shown how we can predict lesser liquid CDS spreads over a list of actively traded and liquid CDS spreads over several years of daily spreads.

ML is useful in finding patterns in contrast to traditional linear models, [16, Chapter 1]. The methods and tools like the nearest neighborhood (NeN), neural networks (NN), or decision trees (DT) suggest better flexibility in finding complex relationships. In fact, prediction as a technique to approximate outcome from supporting features basically recommends practical solutions to economics and finance where the approximation could be very invaluable. Inflation prediction, marketing campaign model testing, growth rates forecast, or market data generation, are just few instances where ML becomes a necessary tool in making decisions, [5, 27]. Additionally, reasonable CPU

times and prediction ability too make ML a promising approach than the traditional regression-like methods.

## 1.4 Contribution

Here, we contribute by providing an ML-based model by comparing and finding several well-known ML methods when there are many features for the model which are CDS rates. As a matter of fact, we attempt in finding the best model when the number of CDS rates are 5, 8, 10 over a period of 5 to 10 years. ML is used because of the existence of large sets of data. It is discussed and illustrated that the nearest neighbor prediction method performs the best when the size of the original set of data is becoming larger and larger. In the presence of such a complex large set of financial data, it will be observed that the regression-type methods which are classical statistical tools cannot anymore be employed to tackle such problems. This paper also follows the recent works [21, 30]. In this paper, we show that ML furnishes an efficient method to solve this challenge in finance.

The advantages of this study comprise:

- Considering many CDS spreads as features for the ML techniques to do the prediction.
- By implementing and imposing several well-known ML predictions, we obtain a method for predicting CDS rates.
- Furnishing insights into the applicability and accuracy of ML-based economic models for predicting CDS rates.

## 1.5 Structure of the paper

After having an introductory discussion regarding the issues with CDS rates and ML in this section, the remaining sections of this article are structured as follows. In Section 2, predictive analytics (PA) is introduced, which comprises different statistical methods from ML, predictive modeling, and data mining which investigate historical and current facts to forecast the forthcoming unknown events. Section 3 furnishes how the proposed procedure can be applied on a large set of financial data. The sample size of the series is more than thousands of observations. It is shown by way of illustration that the proposed solution method for prediction is useful and provides promising results. Several numerical experiments are investigated with implementation details in Section 4 to confirm the applicability of the ML methods and to compare with several well-known and state-of-the-art methods in literature for prediction. Lastly, several concluding summaries are made in Section 5.

## 2 Predictive analytics

PA is a term mostly employed in analytical and statistical methods, [25], which forecasts the future by investigating the historical and current data. The forthcoming occurrences and behavior of variables can be predicted by the PA's models and a score is furnished. A lower score shows a lower likelihood of occurrence of the event and a higher score shows a higher likelihood of occurrence of an event. Transactional and historical data patterns are evaluated by such techniques for finding out the solution to many scientific problems. These models are useful in recognizing the opportunities and risks for each manager, employee, or customer, [11, 19].

Statistically speaking, the problem of prediction simplifies in finding conditional distribution of a variable  $y$  considering other variables  $x = (x_1, x_2, \dots, x_n)$ . Furthermore in the methodology of data science, variables  $x$  are named as features. For the calibrated conditional distribution, generally the prediction point  $y$  is the highest value (mean), [14].

It is well known that the most common tool is (linear) regression analysis. ML recommends a better set of tools that could summarize usefully different types of nonlinear relations in the economic data. A promising predictor includes deriving a function that minimizes an error function. Then, the target of prediction methods is to obtain promising out-of-sample approximations for unseen data. This process is not trivial and generally regressions are known to be weak around out-of-sample predictions. This leads to overfitting issues (especially for regression-like methods of higher orders), [7, Chapter 3].

Basically, the targets of a ML predictive modeling task are twofold: to return a high-performing predictive model for operational use and an approximate of its performance, [9]. The process basically consists of the following stages: (a) Tuning, at which various combinations of methods and their hyper-parameter values are calibrated, (b) Attaining an ultimate model trained on all existing data by the best configuration, and finally (c) Performance estimation.

## 3 Problem set-up

CDS indices are tradable products that permit investors to take short or long credit risk positions in certain equity markets or segments thereof. Here it is considered that the data are generally identically distributed and independent.

### 3.1 Data

Let us consider  $n$  number of CDSs which are served as features, see e.g. [31]. These CDS rates are fed from market but in this work we employ simulated values based on a correlation matrix as follows:

$$A = (a_{i,j})_{n+1 \times n+1}, \quad a_{i,j} = a_{j,i}, \quad a_{i,i} = 1, \quad (2)$$

where its entries are obtained via uniform distributions subject to the *volatility* vector  $V = (v_1, v_2, \dots, v_{n+1})$ ,  $0 < v_i < 1$ . Then the simulated data are extracted from the covariance-variance matrix

$$CM = (v_i v_j a_{i,j})_{n+1 \times n+1}, \quad (3)$$

which is a symmetric positive definite (SPD) matrix. The financial data sets can be constructed in the software system Wolfram Mathematica [1, 2] as comes next:

```
SeedRandom[12];
volatility = Reverse@Sort@RandomReal[{0, 0.1}, n + 1];
crl = RandomReal[{0, 0.5}, {n + 1, n + 1}];
crl = (1/2) (crl + Transpose[crl]);
crl = crl.Transpose[crl];
Table[crl[[i, i]] = 1, {i, n + 1}];
cm = Table[volatility[[i]]*volatility[[j]]*crl[[i, j]],
  {i, 1, Length[volatility]}, {j, 1, Length[volatility]}];

initial = RandomReal[{0.5, 1.5}, n + 1];
max = Number of days in the time period;
mean1 = {0, 0, 0, 0};
mn = MultinormalDistribution[mean1, cm];
data2 = RandomVariate[mn, max];
data1 = Prepend[data2, initial];
data = Accumulate[data1];
```

We have chosen `SeedRandom[12]` on purpose to let readers reproduce the financial data set. The data follow a multivariate normal (Gaussian) distribution with mean vector 0 and variance matrix (3). We now divide the data into several different scenarios:

- $n = 5, 8, 10$  spreads.
- $T = 5, 10$  years which roughly indicates 1825 and 3650 days, respectively.

Financial set of data with higher dimensions and more features is a recent problem. Usually, many sets of data have features with similar information.

This can act as noise in the system and increase the complexity. Note that if the data are fat then it states more features relative to observations or tall, which states many observations relative to features. Figure 1 reveals a sample set of data for  $n = 5, 8, 10$  spreads when  $T = 5$ .

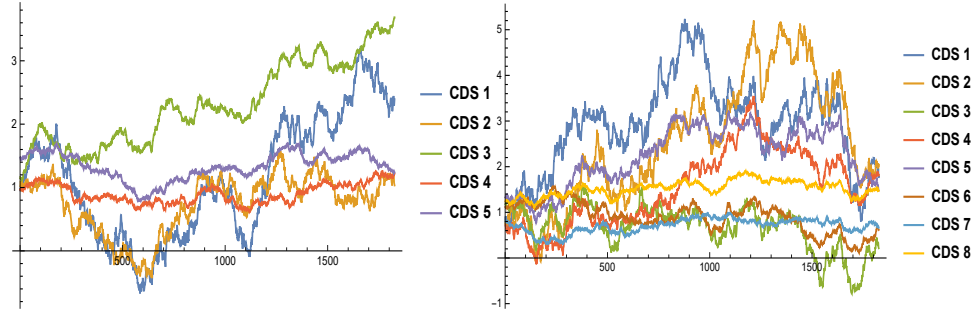


Figure 1: Five years CDS spreads for  $n = 5, 8, 10$ , in top, middle, and bottom, respectively.

### 3.2 The sub-sets

The aim of this subsection is to impose different prediction routines under the ML environment to obtain a model and then employ it for out-of-sample domain, [3, 21]. Here we first break the in-sample data (obtained from the CDS markets in practice) into three famous sets known as (i) training, (ii) validation and (iii) testing sets. The training, validation and testing subsets are roughly 60%, 20% and 20%, of the whole original set, respectively.

Here the data type is in numerical vector format and *only* the first three members of the training set is now given to illustrate how the prediction routine is going to be incorporated ( $n = 10$ ):

```
Training set = {
  {0.839768, 1.48679, 0.710729, 1.48696, 0.713219,
    0.923109, 1.46489, 1.43977, 1.23923, 1.29405} -> 1.2654,
  {0.961012, 1.48665, 0.78337, 1.54342, 0.696918, 0.958177,
    1.51974, 1.45154, 1.25072, 1.3013} -> 1.26877,
  {0.846854, 1.4823, 0.676574, 1.56933, 0.661563, 0.944591,
    1.51674, 1.4319, 1.23782, 1.28701} -> 1.26747
};
```

### 3.3 Predictors

The classification methods are needed for our purpose and we focus on prediction methods. The predictor routines used for comparisons in this work are given as follows:

- RF: Random forest is an ensemble learning method in regression and classification that incorporates deriving a multitude of decision trees (DT). To understand this further, the prediction of the forest is attained by the mean-value tree predictions or getting the most common class. Every DT is trained on a random subset of the set of training and employs only a random subset of the features.
- DT: A DT is a flow chart-like network, in which each internal node shows a test on a feature, each branch shows the test's outcome, and each leaf shows a probability density, value distribution or class distribution.
- GBT: The Gradient boosting tree is a technique of ML for classification and problems that provides a prediction model as an ensemble of trees. The training on the trees is done sequentially with the aim of compensating for the weaknesses of previous trees. Basically the Light Gradient Boosting Machine framework is used in the back end.
- LR: The linear regression forecasts the computational value  $y$  employing a linear combination of numerical features  $x = \{x_1, x_2, \dots, x_n\}$ . The conditional probability  $P(y|x)$  is modeled based on

$$P(y|x) \propto \exp\left(-\frac{(y - f(\theta, x))^2}{2\sigma^2}\right), \quad (4)$$

with  $f(\theta, x) = x\theta$ .


- NN: A neural network comprises stacked layers, each doing a simple calculation. Then, the information layer by layer is processed from the input layer to the output layer. A loss function is minimized to train the NN on the training set employing gradient descent.
- NeN: Nearest neighbors is a kind of instance-based learning and it chooses the averages of the values among the  $k$  nearest neighbors or the commonest class in its simplest form. To generate short term forecasts, similar patterns of behavior are located with respect to NeN by a distance measure that is normally the Euclidean distance. The time evolution of these NeNs is exploited to obtain the required forecast. Thus, the procedure employs only local information to predict and makes no effort to fit a model to the whole time series at once. The selection of the size ( $m$ ) normally called the embedding dimension and

of the number of neighbors ( $k$ ) is a fundamental point of this routine, [10]. Note that there is no training step during NeN.

- GP: The Gaussian method is via the assumption of a Gaussian process for the model. This process is expressed by its covariance function and it will estimate the parameters of this covariance function in the training phase. Then, it is conditioned on the training data and employed to infer the value of a new instance by a Bayesian inference.

Having a large set of data, we assign 20% of the data in each scenario for the validation set. Typically this is employed when the data in the training set and the data that we want to forecast arise from various resources. Using this, the hyper-parameter selections are done by testing performance on data.

In addition, the predictors are set and trained on the target device CPU, while we use 1234 as random seeding whenever required inside the predictors. Our prediction routine can be obtained as a predictor function. As an instance for the case of NeN when  $n = 10$ ,  $T = 5$ , it can be written in what follows:

PredictorFunction[  Input type: NumericalVector (length: 10)  
Method: NearestNeighbors  
Number of training examples: 1095 ]

To illustrate the results of comparisons, we furnish Table 1 *for one run* in the case  $n = 5$ ,  $T = 5$ , which shows that number of training examples, validation set examples, and test set instances, would be 1095, 365, and 366, respectively. Here since we employ the built-in functions for the predictor routines in Mathematica, so all the hyper-parameters have been assigned automatically based on the built-in optimization techniques to choose the best hyper-parameters for the model corresponding to the training and validity sets.

Predictors are compared with respect to their CPU times (seconds) in Table 2. The time reported is based on the running CPU times for constructing the models. A critical bottleneck in using the NN is its high computational time for constructing the model. This restricts its applicability for the purpose of our financial application in predicting lesser liquid CDS rates. In other words, the larger the financial data set (of this type), the larger the CPU time is. Therefore, it becomes requisite to improve and rely on alternate predictors.

Meanwhile, the testing stage requires the comparison of the test vector to all existing data points in the data set which might take a significant amount of time.



## 4 Benchmarking

The implementations in this paper were performed by Mathematica 12.0 [17, Chapter 7] installed in a computer having Core i7-9750H with SSD memory and 16 GB RAM. It is important to mention that the general results and conclusion are obtained by shuffling the data set for 50 times and getting means whenever required.

### 4.1 Implementation details

We can apply several predictive routines defined in Subsection 3.3 to attain the forthcoming value in the out-of-sample domain (for unseen data). In ML, the validity set is used to tune our model parameter settings and a test set to evaluate the model's performance on unseen events. The procedure of prediction here using our ML methods on the list of large data sets is incorporating a prediction function on the unseen data.

Table 1: Information of compared predictors for  $n = 5$ , and  $T = 5$ .

	Sub-method, parameters	Single evaluation time (ms/ example)	Batch evaluation speed (example/ ms)	Loss	Model memory (kB)	Running time (s)
RF	Feature fraction=1/3, Leaf size = 4, Tree number = 100	8.4	12.9	$-1.83 \pm 0.01$	605	1.3
DT	Feature fraction=1, Distribution smoothing = 1	1.14	487.	$-1.43 \pm 0.08$	121	0.6
GBT	Leaves number=60, Learning rate = 0.2, Leaf size = 7	4.04	44.0	$+2.63 \pm 0.26$	723	10.93
LR	L1 regularization=0, L2 regularization = 100.0, Optimization method = Normal equation	1.31	362.	$-2.61 \pm 0.03$	307	3.72
NN	Network depth=2, Max training rounds = 30	2.98	25.1	$-2.95 \pm 0.04$	471	38.1
NeN	Neighbors number=2, Distribution smoothing = 0.5, Nearest method = $k$ -D tree	1.11	194.	$+0.5 \pm 0.49$	174	0.6
GP	Estimation method =Maximum posterior, Search method = Simulated Annealing	3.83	9.7	$-0.98 \pm 0.26$	6.3	5.6

Table 2: Comparisons of running times for various routines to construct the model.

$n$	$T$	RF	DT	GBT	LR	NN	NeN	GP
5	5	1.3	0.6	10.93	3.72	38.1	0.6	5.6
8	5	0.8	0.6	10.8	3.7	34.6	0.6	5.2
10	5	1.2	0.69	21.5	3.79	86.	0.7	5.9
5	10	1.4	1.29	13.0	4.42	85.	1.32	31.1
8	10	1.81	1.30	13.0	4.3	158.	1.40	33.8
10	10	1.87	1.31	17.2	4.06	419	1.41	31.3

To compare various routines for prediction, two different measures are employed as described below. The absolute error is calculated via

$$\varepsilon = \|p_{\text{predict}} - p_{\text{actual}}\|_{\infty}, \quad (5)$$

where  $p_{\text{actual}}$  and  $p_{\text{predict}}$  are the exact and predicted values, respectively. Also, the root mean square relative error (RMSRE) of  $\mathcal{N}$  predicted values  $p_{\text{predict}}$  whose real values are  $p_{\text{actual}}$ , is defined by

$$\epsilon = \sqrt{\sum_{i=1}^{\mathcal{N}} \frac{1}{\mathcal{N}} \left| \frac{p_{\text{predict}}^i - p_{\text{actual}}^i}{p_{\text{actual}}^i} \right|^2}. \quad (6)$$

Here  $\mathcal{N}$  is the length of each sub-sets. For this analysis, we compared the  $\varepsilon$  and  $\epsilon$  for each training, validity and test sets in each scenario. The routines have not seen the test sets before and the accuracies that come from the incorporation of the test sets are important to find the most useful algorithm for prediction. The results based on a *mean of over 50 shuffles* on the original set containing the training, validity, and test (prediction) sets, each time, are gathered in Table 3. It is important to state that for some routines such as NeN, there is no training part and its model representation is the entire training dataset, [24, Chapter 7]. The training set in Table 3 for such methods is in fact the calibration set.

**Remark 1.** Here it is noted that RF is an ensemble DT model based on different feature selections and data set partitions that do not require cross-validation. However, it is used in a similar fashion just like the other models. This is mainly to check the robustness of the models when the input data change. The more general the model, the less susceptible it would be to data variation. And every model needs to be cross-validated and because of this as well as having fair comparisons, we compute mean over 50 shuffles on the original set.

Also note that KNN makes predictions using the training dataset directly.

Table 3: Comparison of mean results among different methods.

Scenario	Routine	$\varepsilon$			$\epsilon$		
		Training set	Validity set	Test set	Training set	Validity set	Test set
$n = 5, T = 5$	RF	$5.5 \times 10^{-2}$	$4.4 \times 10^{-2}$	$4.6 \times 10^{-2}$	$9.3 \times 10^{-3}$	$9.7 \times 10^{-3}$	$9.8 \times 10^{-3}$
	DT	$6.6 \times 10^{-2}$	$6.5 \times 10^{-2}$	$6.5 \times 10^{-2}$	$1.0 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-2}$
	GBT	$2.9 \times 10^{-2}$	$3.9 \times 10^{-2}$	$3.9 \times 10^{-2}$	$3.2 \times 10^{-3}$	$5.6 \times 10^{-3}$	$5.5 \times 10^{-3}$
	LR	$5.2 \times 10^{-2}$	$5.0 \times 10^{-2}$	$5.0 \times 10^{-2}$	$1.3 \times 10^{-2}$	$1.3 \times 10^{-2}$	$1.3 \times 10^{-2}$
	NN	$2.6 \times 10^{-2}$	$2.4 \times 10^{-2}$	$2.6 \times 10^{-2}$	$4.0 \times 10^{-3}$	$4.8 \times 10^{-3}$	$4.9 \times 10^{-3}$
	NeN	$3.1 \times 10^{-2}$	$3.3 \times 10^{-2}$	$3.4 \times 10^{-2}$	$2.6 \times 10^{-3}$	$4.2 \times 10^{-3}$	$4.3 \times 10^{-3}$
$n = 8, T = 5$	GP	$2.4 \times 10^{-2}$	$2.7 \times 10^{-2}$	$2.6 \times 10^{-2}$	$3.3 \times 10^{-3}$	$4.7 \times 10^{-3}$	$4.7 \times 10^{-3}$
	RF	$2.6 \times 10^{-2}$	$2.6 \times 10^{-2}$	$2.6 \times 10^{-2}$	$7.1 \times 10^{-3}$	$7.2 \times 10^{-3}$	$7.2 \times 10^{-3}$
	DT	$5.4 \times 10^{-2}$	$5.1 \times 10^{-2}$	$5.0 \times 10^{-2}$	$8.1 \times 10^{-3}$	$8.9 \times 10^{-3}$	$8.9 \times 10^{-3}$
	GBT	$1.9 \times 10^{-2}$	$2.6 \times 10^{-2}$	$2.6 \times 10^{-2}$	$2.1 \times 10^{-3}$	$4.1 \times 10^{-3}$	$4.1 \times 10^{-3}$
	LR	$4.4 \times 10^{-2}$	$4.3 \times 10^{-2}$	$4.3 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-2}$
	NN	$1.7 \times 10^{-2}$	$1.6 \times 10^{-2}$	$1.7 \times 10^{-2}$	$2.7 \times 10^{-3}$	$3.4 \times 10^{-3}$	$3.4 \times 10^{-3}$
$n = 10, T = 5$	NeN	$1.6 \times 10^{-2}$	$1.8 \times 10^{-2}$	$1.8 \times 10^{-2}$	$2.0 \times 10^{-3}$	$3.1 \times 10^{-3}$	$3.1 \times 10^{-3}$
	GP	$1.7 \times 10^{-2}$	$1.7 \times 10^{-2}$	$1.7 \times 10^{-2}$	$1.6 \times 10^{-3}$	$3.3 \times 10^{-3}$	$3.3 \times 10^{-3}$
	RF	$1.7 \times 10^{-2}$	$1.7 \times 10^{-2}$	$1.6 \times 10^{-2}$	$6.0 \times 10^{-3}$	$5.8 \times 10^{-3}$	$5.8 \times 10^{-3}$
	DT	$3.6 \times 10^{-2}$	$3.2 \times 10^{-2}$	$3.3 \times 10^{-2}$	$6.2 \times 10^{-3}$	$6.7 \times 10^{-3}$	$6.7 \times 10^{-3}$
	GBT	$1.2 \times 10^{-2}$	$1.4 \times 10^{-2}$	$1.3 \times 10^{-2}$	$1.1 \times 10^{-3}$	$2.1 \times 10^{-3}$	$2.2 \times 10^{-3}$
	LR	$1.8 \times 10^{-2}$	$1.7 \times 10^{-2}$	$1.7 \times 10^{-2}$	$4.6 \times 10^{-3}$	$4.7 \times 10^{-3}$	$4.7 \times 10^{-3}$
$n = 5, T = 10$	NN	$7.0 \times 10^{-3}$	$7.1 \times 10^{-3}$	$7.3 \times 10^{-3}$	$1.1 \times 10^{-3}$	$1.5 \times 10^{-3}$	$1.5 \times 10^{-3}$
	NeN	$1.0 \times 10^{-2}$	$1.0 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.4 \times 10^{-3}$	$2.1 \times 10^{-3}$	$2.1 \times 10^{-3}$
	GP	$1.7 \times 10^{-2}$	$1.3 \times 10^{-2}$	$1.5 \times 10^{-2}$	$2.0 \times 10^{-3}$	$2.3 \times 10^{-3}$	$2.4 \times 10^{-3}$
	RF	$5.0 \times 10^{-2}$	$4.8 \times 10^{-2}$	$5.0 \times 10^{-2}$	$9.5 \times 10^{-3}$	$9.7 \times 10^{-3}$	$9.7 \times 10^{-3}$
	DT	$7.0 \times 10^{-2}$	$6.6 \times 10^{-2}$	$6.5 \times 10^{-2}$	$9.9 \times 10^{-3}$	$1.0 \times 10^{-2}$	$1.0 \times 10^{-2}$
	GBT	$3.5 \times 10^{-2}$	$4.1 \times 10^{-2}$	$4.2 \times 10^{-2}$	$4.0 \times 10^{-3}$	$5.6 \times 10^{-3}$	$5.6 \times 10^{-3}$
$n = 8, T = 10$	LR	$6.8 \times 10^{-2}$	$6.6 \times 10^{-2}$	$6.7 \times 10^{-2}$	$1.5 \times 10^{-2}$	$1.5 \times 10^{-2}$	$1.5 \times 10^{-2}$
	NN	$2.7 \times 10^{-2}$	$2.7 \times 10^{-2}$	$2.8 \times 10^{-2}$	$4.0 \times 10^{-3}$	$4.7 \times 10^{-3}$	$4.7 \times 10^{-3}$
	NeN	$1.9 \times 10^{-2}$	$2.9 \times 10^{-2}$	$2.8 \times 10^{-2}$	$1.8 \times 10^{-3}$	$3.8 \times 10^{-3}$	$3.4 \times 10^{-3}$
	GP	$2.7 \times 10^{-2}$	$2.8 \times 10^{-2}$	$2.9 \times 10^{-2}$	$3.2 \times 10^{-3}$	$4.4 \times 10^{-3}$	$4.4 \times 10^{-3}$
	RF	$3.7 \times 10^{-2}$	$3.9 \times 10^{-2}$	$3.9 \times 10^{-2}$	$1.2 \times 10^{-2}$	$1.2 \times 10^{-2}$	$1.2 \times 10^{-2}$
	DT	$7.2 \times 10^{-2}$	$7.2 \times 10^{-2}$	$6.9 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-2}$
$n = 10, T = 10$	GBT	$2.5 \times 10^{-2}$	$3.0 \times 10^{-2}$	$3.0 \times 10^{-2}$	$3.4 \times 10^{-3}$	$4.9 \times 10^{-3}$	$4.9 \times 10^{-3}$
	LR	$6.5 \times 10^{-2}$	$6.3 \times 10^{-2}$	$6.3 \times 10^{-2}$	$1.6 \times 10^{-2}$	$1.6 \times 10^{-2}$	$1.6 \times 10^{-2}$
	NN	$1.9 \times 10^{-2}$	$2.1 \times 10^{-2}$	$2.0 \times 10^{-2}$	$3.3 \times 10^{-3}$	$4.0 \times 10^{-3}$	$3.9 \times 10^{-3}$
	NeN	$1.3 \times 10^{-2}$	$1.9 \times 10^{-2}$	$2.0 \times 10^{-2}$	$1.6 \times 10^{-3}$	$3.0 \times 10^{-3}$	$3.0 \times 10^{-3}$
	GP	$2.0 \times 10^{-2}$	$2.0 \times 10^{-2}$	$2.1 \times 10^{-2}$	$2.5 \times 10^{-3}$	$3.7 \times 10^{-3}$	$3.7 \times 10^{-3}$
	RF	$3.1 \times 10^{-2}$	$3.1 \times 10^{-2}$	$3.2 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-2}$
$n = 10, T = 10$	DT	$7.7 \times 10^{-2}$	$7.1 \times 10^{-2}$	$6.7 \times 10^{-2}$	$8.9 \times 10^{-3}$	$9.6 \times 10^{-3}$	$9.5 \times 10^{-3}$
	GBT	$2.0 \times 10^{-2}$	$2.1 \times 10^{-2}$	$2.3 \times 10^{-2}$	$1.7 \times 10^{-3}$	$2.7 \times 10^{-3}$	$2.7 \times 10^{-3}$
	LR	$2.9 \times 10^{-2}$	$2.8 \times 10^{-2}$	$2.8 \times 10^{-2}$	$8.0 \times 10^{-3}$	$8.1 \times 10^{-3}$	$8.0 \times 10^{-3}$
	NN	$1.3 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.2 \times 10^{-2}$	$1.7 \times 10^{-3}$	$2.0 \times 10^{-3}$	$2.0 \times 10^{-3}$
	NeN	$8.6 \times 10^{-3}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-2}$	$1.1 \times 10^{-3}$	$2.0 \times 10^{-3}$	$2.0 \times 10^{-3}$
	GP	$1.1 \times 10^{-2}$	$1.0 \times 10^{-2}$	$1.0 \times 10^{-2}$	$1.7 \times 10^{-3}$	$1.9 \times 10^{-3}$	$1.9 \times 10^{-3}$

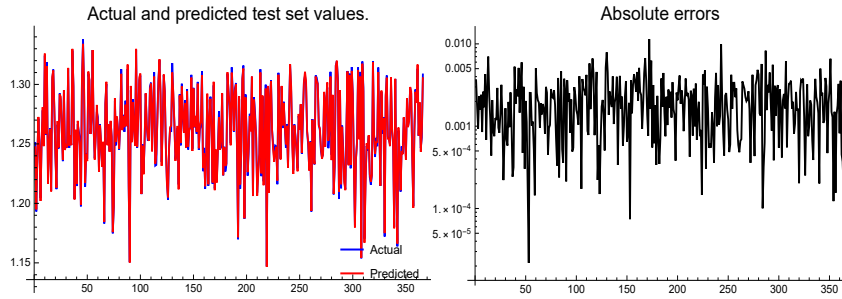


Figure 2: NeN results on the test sub-set for *one run* of the scenario  $n = 10$ , and  $T = 5$  in left and its associated log absolute errors in right. See the online colorful version for further clarification.

## 4.2 Quantitative results

The large data sets considered for checking and comparing the presented models have 1825 and 3650 members each comprising 5, 8 and 10 elements respectively.

The results of comparisons for the unseen CDS rates (the test sets) are provided in Table 3. We observe that all the predictors replicated the training sets quite well. But it is observable that the best one by considering both the numerical accuracy and CPU time is mostly NeN. In order to save space and avoid providing repeated similar figures of comparisons, Figure 2 is provided only for the scenario  $n = 10, T = 5$  to reveal that the NeN furnishes promising prediction on the test (prediction) sub-set.

We visualize the scatter plot of the test values as a function of the predicted values in Figure 3 for one run. This illustrates again the point that the size of the input financial data and their types have clear effect on the choice of the predictors in ML. We have illustrated the prediction by CDS data and revealed the application of non-regression tools as better techniques in PA.

It is well known that data correlation is the way at which one data set can correspond to another data set, [29]. And in ML, we can think of how the features correspond with the output. Data visualization and correlation may help decide, which ML method to use. Accordingly, Table 4 provides the means of correlations for 50 runs of the shuffled data among the actual and predicted values of different methods under several scenarios to also help in obtaining the best method for our CDS problem. Noticing that small values may not necessarily represent a bad correlation as long as the set of data has a large statistically significant correlation. When we have datasets with many features, the data correlation would be of clear importance.

Considering all the results given in Tables 1-4 reveal that NeN is the best ML routine that can be considered for prediction of lesser liquid CDS rates

under the format of the financial data described in Section 3 both in terms of accuracy and the elapsed CPU time. NeN is a non-parametric non-linear forecasting routine, which is mainly via pieces of time series, in the past, that may have a resemblance to pieces in the future. This is useful in finance, see also [18]. For NeN, no learning of is needed and all of the work happens at the time a prediction is requested. Recalling that the number of neighbors selected, the length of the series and the embedding dimension to perform predictions are indicated automatically by the built-in routines inside our programming language.

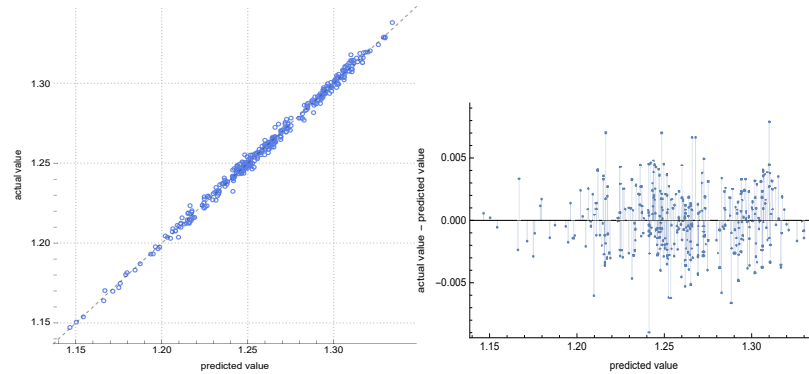


Figure 3: Comparison of perfect prediction line and predictions (left) and the residual plot (right) in NeN routine for the scenario  $n = 10$ , and  $T = 5$ .

Table 4: Correlation between the actual and predicted values.

$n$	$T$	RF	DT	GBT	LR	NN	NeN	GP
5	5	0.97	0.96	0.99	0.95	0.99	0.99	0.99
8	5	0.97	0.96	0.99	0.93	0.99	0.99	0.99
10	5	0.97	0.97	0.99	0.98	0.99	0.99	0.99
5	10	0.97	0.96	0.99	0.92	0.99	0.99	0.99
8	10	0.97	0.97	0.99	0.95	0.99	0.99	0.99
10	10	0.97	0.98	0.99	0.98	0.99	0.99	0.99

Computational pieces of evidence from Table 3 reveal that NN, NeN and GP have the best performance on the unseen test sets over 50 shuffles. However, accuracy is not the main factor in choosing the best routine when the computational CPU times are different. Clearly the lower the CPU time of training, the more useful method we have when the accuracies are almost the same. Due to this and based on Table 2, the best performance belongs to NeN in terms of computational time. Thus, we have found the NeN as our model that could be performed automatically without any theoretical assumptions through the process of progressive adaptation.

This subsection is ended by pointing out that some other models such as RNN and LSTM can also be used for comparisons that implicitly fit well with time series data. For such routines, we remind that, e.g., LSTMs take longer to train, require more memory to train, are easy to overfit, are sensitive to different random weight initializations and dropout is much harder to implement in LSTMs. Besides, our data are not stock prices and the NeN already performed best in terms of running time as well as the accuracy. Thus, comparisons to other solvers is no longer necessary.

### 4.3 Results on unseen data

Finally in this section, it is necessary to illustrate how the NeN as an efficient ML technique for predicting lesser liquid CDS rates can be imposed on totally new unseen data. Considering the NeN routine as the predictor, saved as `prediction` previously and obtained when  $n = 10$ , and  $T = 10$ , then the data for this verification are simulated as follows:

```
n2 = 10;
SeedRandom[12345];
volatility2 = Reverse@Sort@RandomReal[{0, 0.1}, n2];
cr12 = RandomReal[{0, 0.5}, {n2, n2}];
cr12 = (1/2) (cr12 + Transpose[cr12]);
cr12 = cr12.Transpose[cr12];
Table[cr12[[i, i]] = 1, {i, n}];
cm2 = Table[volatility2[[i]]*volatility2[[j]]*cr12[[i, j]],
  {i, 1, Length[volatility2]}, {j, 1, Length[volatility2]}];
initial2 = RandomReal[{0.1, 2.0}, n2];
max2 = 100;
mean2 = ConstantArray[0, n2];
mn2 = MultinormalDistribution[mean2, cm2];
data22 = RandomVariate[mn2, max2];
data12 = Prepend[data22, initial2];
dataTest = Accumulate[data12];
dataTest2 = RandomSample[dataTest];
```

Now we impose the prediction model from NeN on unseen data:

```
pdataTest = prediction[dataTest2];
```

Here for 101 unseen data, we obtain the results of predictions based on NeN and plot them in Figure 4 (left), while the distribution for this unseen data can be obtained as follows:

with the following probability density of the predicted values:  
and illustrated in Figure 4 (right).

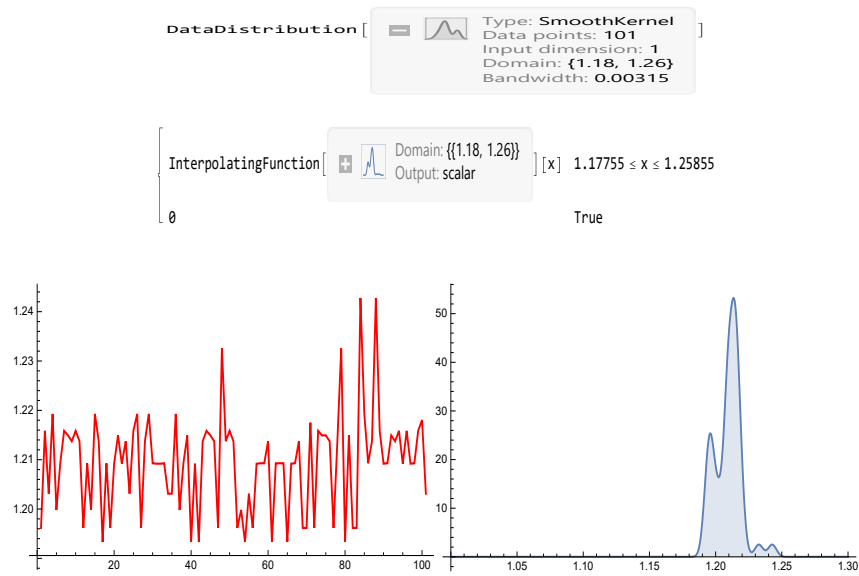


Figure 4: Predicted values (left) and their data distribution (right) in NeN routine for unseen data.

## 5 Conclusions and future work

Recently, there has been a proliferation of ML techniques and growing interest in their applications in finance, where they have been applied to sentiment analysis of news, trend analysis, and portfolio optimization. This paper has explored the potential of ML to enhance the investment process. A prediction model based on ML was discussed in equity markets when the numbers of predictors/features which are CDS rates are high over a larger period of time. This is especially relevant to finance where the ability of data groups to forecast the values of lesser liquid tools is of high interest. The results obtained in this work are useful as a model for predicting lesser liquids and can be employed for further investigation of the dynamic relationship between the VIX index and the CDS markets. When dealing with multidimensional set of data, it is requisite to filter out non-correlated features. Instead, it is better to use fewer highly correlated features to train a model. Taking into account such consideration may help improve the robustness of the NeN method, which is under study for further work in our team.

## Acknowledgements

The author is grateful to two anonymous referees for their constructive comments and corrections on an earlier version of this work.

## References

- [1] R. Adhikari, *Foundations of computational finance*, The Mathematica J., 22 (2020), 1-59.
- [2] R. Adhikari, *Selected financial applications*, The Mathematica J., 23 (2021), 1-33.
- [3] A. Antonov, *Variable importance determination by classifiers implementation in Mathematica*, Lecture Notes, Florida, 2015.
- [4] G.O. Aragon, L. Li, J. Qian, *The use of credit default swaps by bond mutual funds: Liquidity provision and counterparty risk*, J. Finan. Econ., 131 (2019), 168-185.
- [5] J. Bao, S. Franco, Y.-H. He, E. Hirst, G. Musiker, Y. Xiao, *Quiver mutations, Seiberg duality, and machine learning*, Phys. Rev. D., 102 (2020), Art. ID: 086013.
- [6] T.R. Bielecki, M.R. Rutkowski, *Credit Risk: Modeling, Valuation and Hedging*, Springer, New York, 2004.
- [7] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, 2006.
- [8] D. Brigo, N. Pede, A. Petrelli, *Multi currency credit default swaps*, Int. J. Theor. Appl. Finan., 22 (2019), 1950018.
- [9] S. Carbo-Valverde, P. Cuadros-Solas, F. Rodríguez-Fernández, *A machine learning approach to the digitalization of bank customers: Evidence from random and causal forests*, Plos One, 15 (2020), Art. ID: e0240362.
- [10] G.H. Chen, D. Shah, *Explaining the success of nearest neighbor methods in prediction*, Found. Trends Mach. Learn., 10 (2018), 337-588.
- [11] K. Cortez, M. Rodríguez-García, S. Mongrut, *Exchange market liquidity prediction with the K-nearest neighbor approach: Crypto vs. fiat currencies*, Mathematics, 9 (2021), Art. ID: 56.
- [12] P.P. da Silva, I. Vieira, C. Vieira, *M&A operations: Further evidence of informed trading in the CDS market*, J. Multi. Fin. Manag. 32-33 (2015), 116-130.
- [13] M.L. De Prado, *Advances in Financial Machine Learning*, Wiley, New Jersey, 2018.
- [14] W. E, *Machine learning and computational mathematics*, Commun. Comput. Phys., 28 (2020), 1639-1670.



- [15] F.J. Fabozzi, S.M. Focardi, P.N. Kolm, *Trends in Quantitative Finance*, The Research Foundation of CFA Institute, USA, 2006.
- [16] G. Gan, C. Ma, J. Wu, *Data Clustering: Theory, Algorithms, and Applications*, SIAM, Philadelphia, 2007.
- [17] N.L. Georgakopoulos, *Illustrating Finance Policy with Mathematica*, Springer International Publishing, Cham, Switzerland, 2018.
- [18] D. Guégan, N. Huck, *On the use of nearest neighbors in finance*, Finance, 26 (2005), 67-86.
- [19] B.M. Henrique, V.A. Sobreiro, H. Kimura, *Literature review: Machine learning techniques applied to financial market prediction*, Expert Sys. Appl., 124 (2019), 226-251.
- [20] I. Hlivka, *Credit default swap valuation*, Lecture Notes, London, Quant Solutions Group, (2014), 1-2.
- [21] I. Hlivka, *Predictive analytics in finance: Patterns detection for outcome prediction*, Lecture Notes, London, Quant Solutions Group, (2015), 1-14.
- [22] A. Itkin, A. Lipton, D. Muravey, *Generalized Integral Transforms in Mathematical Finance*, World Scientific Publishing, Toh Tuck, Singapore, 2021.
- [23] A. Itkin, V. Shcherbakov, A. Veygman, *New model for pricing quanto credit default swaps*, Int. J. Theor. Appl. Fin., 22 (2019), Art. ID: 1950003.
- [24] M. Kuhn, K. Johnson, *Applied Predictive Modeling*, 1st ed., Springer Science + Business Media, New York, 2013.
- [25] V. Kumar, M.L. Garg, *Predictive analytics: A review of trends and techniques*, Int. J. Comput. Appl., 182 (2018), 31-37.
- [26] R. Mohamadinejad, A. Neisy, J. Biazar, *ADI method of credit spread option pricing based on jump-diffusion model*, Iran. J. Numer. Anal. Optim., 11 (2021), 195-210.
- [27] A. Mosavi, Y. Faghan, P. Ghamisi, P. Duan, S.F. Ardabili, E. Salwana, S.S. Band, *Comprehensive review of deep reinforcement learning methods and applications in economics*, Mathematics, 8 (2020), Art. ID. 1640.
- [28] H. Ni, X. Dong, J. Zheng, G. Yu, *An Introduction to Machine Learning in Quantitative Finance*, World Scientific Publishing Europe Ltd., London, 2021.
- [29] L. Sandoval Junior, *Correlation of financial markets in times of crisis*, Phys. A: Stat. Mech. Appl., 391 (2012), 187-208.

- [30] J. Sirignano, A. Sadhwani, K. Giesecke, *Deep learning for mortgage risk*, J. Finan. Econometrics, 19 (2021), 313-368.
- [31] Y. Son, H. Byun, J. Lee, *Nonparametric machine learning models for predicting the credit default swaps: An empirical study*, Expert Sys. Appl., 58 (2016), 210-220.

**How to cite this article**

Soleymani, F., Finding an efficient machine learning predictor for lesser liquid credit default swaps in equity markets. *Iran. j. numer. anal. optim.*, 2023; 13(1): 19-37. <https://doi.org/10.22067/ijnao.2022.73453.1073>.



# A modified Liu-Storey scheme for nonlinear systems with an application to image recovery

A.I. Kiri, M.Y. Waziri\*, and K. Ahmed

## Abstract

Like the Polak-Ribière-Polyak (PRP) and Hestenes-Stiefel (HS) methods, the classical Liu-Storey (LS) conjugate gradient scheme is widely believed to perform well numerically. This is attributed to the in-built capability of the method to conduct a restart when a bad direction is encountered. However, the scheme's inability to generate descent search directions, which is vital for global convergence, represents its major shortfall. In this article, we present an LS-type scheme for solving system of monotone nonlinear equations with convex constraints. The scheme is based on the approach by Wang et al. (2020) and the projection scheme by Solodov and Svaiter (1998). The new scheme satisfies the important condition for global convergence and is suitable for non-smooth nonlinear problems. Furthermore, we demonstrate the method's application in restoring blurry images in compressed sensing. The scheme's global convergence is established under mild assumptions and preliminary numerical results show that the proposed method is promising and performs better than two recent methods in the literature.

**AMS subject classifications (2020):** Primary 90C30; Secondary 90C26, 94A12.

\*Corresponding author

Received 9 January 2022; revised 18 May 2022; accepted 19 May 2022

Aliyu Ibrahim Kiri

Department of Mathematical Sciences, Bayero University, Kano, Nigeria. e-mail: aikiri.mth@buk.edu.ng

Mohammed Yusuf Waziri

Department of Mathematical Sciences, Bayero University, Kano, Nigeria. e-mail: my-waziri.mth@buk.edu.ng

Kabiru Ahmed

Department of Mathematical Sciences, Bayero University, Kano, Nigeria. e-mail: kabiruhungu16@gmail.com

**Keywords:** Nonlinear Monotone Equations; Line search; Projection method; Signal processing; Convex constraint; Image de-blurring.

## 1 Introduction

In this paper, the following constrained system of nonlinear equations is considered:

$$F(x) = 0, \quad x \in \Phi, \quad (1)$$

where  $F : \mathbf{R}^n \rightarrow \mathbf{R}^n$  is a nonlinear mapping, which is continuous and monotone, namely it satisfies the inequality

$$(F(x) - F(y))^T(x - y) \geq 0, \quad \forall x, y \in \mathbf{R}^n. \quad (2)$$

Also,  $\Phi$  in (1) is nonempty, closed convex set, which is sometimes expressed as

$$\Phi = \{x \in \mathbf{R}^n : u \leq x \leq v\}, \quad (3)$$

where  $u$  and  $v$  stand for the lower and upper bounds on the vector  $x$ . Moreover, when  $\Phi$  is expressed as (3), the problem in (1) is referred to as box constrained nonlinear system.

Several applications in science, engineering and other areas of human endeavour involve the system of equations represented in (1). For example, in problem of radiative transfer and transport theory [15], the popular Chandrasekhar integral equations is discretized and expressed as (1). Also, the economic equilibrium problems studied in [5, 24], are reformulated as problem (1). In addition, some  $\ell_1$ -norm regularized optimization problems in signal and image processing [23, 46] are obtained by reformulating systems of monotone nonlinear equations.

In order to solve (1), various iterative schemes have been proposed over the years. Newton-type schemes [32] and their improved variants, the quasi-Newton schemes [4, 17] are the most widely used due to their rapid convergence properties. These methods, however, are not suitable for large-dimension problems due to their huge matrix storage requirement. The conjugate gradient (CG) method, by virtue of its low memory requirement is the proper choice for problems with large dimensions. The scheme was primarily developed to solve the unconstrained optimization problem

$$\min_{x \in \mathbf{R}^n} f(x), \quad (4)$$

where  $f : \mathbf{R}^n \rightarrow \mathbf{R}$  denotes a nonlinear mapping that is assumed to be at least twice continuously differentiable and bounded below. The following notations are used with respect to problem (4):

$$g_k = g(x_k) = \nabla f(x_k), \quad g_{k-1} = g(x_{k-1}), \quad y_{k-1} = g_k - g_{k-1}.$$

The scheme is implemented using the iterative formula

$$x_0 \in \mathbf{R}^n, \quad x_{k+1} = x_k + s_k, \quad s_k = \alpha_k d_k, \quad k = 0, 1, \dots, \quad (5)$$

where  $x_k$  denotes the  $k^{th}$  iterate,  $\alpha_k$  represents a step-size that is computed using an appropriate line search technique, and  $d_k$  is the CG search direction defined by

$$d_k = -g_k + \beta_k d_{k-1}, \quad d_0 = -g_0, \quad (6)$$

and  $\beta_k$  is a scalar known as the CG (update) parameter, which defines each CG scheme and influences its performance. As part of the conditions for the sequence of iterative points  $\{x_k\}$  generated via (5) and (6) to converge globally, the following sufficient descent condition is required:

$$d_k^T g_k \leq -\tau \|g_k\|^2, \quad \tau > 0. \quad (7)$$

Due to the fine attributes of CG methods with the known fact that the first order optimality condition for (4), namely,  $\nabla f(x) = 0$ , is equivalent to (1) with  $F = \nabla f$  denoting the gradient of some nonlinear functions, different adaptations of the scheme for solving (1) have been proposed over the years. Search directions of these adaptations are defined as

$$d_k = \begin{cases} -F_k, & \text{if } k = 0; \\ -F_k + \beta_k d_{k-1}, & \text{otherwise,} \end{cases} \quad (8)$$

where  $F_k = F(x_k)$ .

One of the CG adaptations that has gained attention of researchers in the last decade is the three-term methods for solving (1). Wang et al. [35], proposed a self-adaptive three-term nonlinear CG method for solving convex constrained system of nonlinear equations. Search direction of the scheme was obtained by employing an adaptive technique and under mild conditions, its global convergence was established. Based on the hyperplane projection scheme [32], Gao and He [9], proposed a three-term modified CG method for solving (1). Due to its derivative-free and low storage requirement, the method is also ideal for nonsmooth nonlinear problems. Motivated by the works in [1, 13, 27, 47], Koorapetse and Kaelo [25] also proposed three-term adaptation of CG projection methods for solving (1). The proposed methods were proven to satisfy the global convergence condition and under suitable assumptions, the authors showed that the schemes converge globally. For more details see ([44, 37, 38, 39, 40, 31, 41, 11, 42, 10, 43, 12, 33, 26]).

In this paper, our aim is to develop an efficient three-term adaptation of the Liu-Storey (LS) [22] CG method for solving (1). Our inspiration comes from the work of Wang et al. [36] and the projection method by Solodov and Svaiter [32]. Apart from developing a new scheme that satisfies the condition for global convergence, a notable contribution of this research is

its application in restoring blurry images, which is a trend in compressed sensing.

The paper is organized as follows: Some preliminaries leading to derivation of the proposed method are given in Section 2. The proposed method and its convergence analysis are presented in Section 3. Numerical results and their discussions are presented in Section 4, while application of the proposed scheme is discussed in Section 5. Concluding remarks are made in Section 6.

## 2 Preliminaries

In the remaining part of the article,  $\|x\| = \sqrt{x^T x}$ , stands for the  $\ell_2$  norm,  $F_{k-1} = F(x_{k-1})$ , and  $s_{k-1} = x_k - x_{k-1}$ . The following assumptions will be required later in the article:

- (i) The solution set  $\Phi$  is not empty, namely, there exists  $\tilde{x} \in \Phi$  such that  $F(\tilde{x}) = 0$ .
- (ii) The mapping  $F$  satisfies the Lipschitz continuity property; i.e, there exists a positive constant  $L$  such that for all  $x, y \in \mathbf{R}^n$ , the following is satisfied:

$$\|F(x) - F(y)\| \leq L\|x - y\|. \quad (9)$$

We now introduce the projection operator. Let  $\Phi \subset \mathbf{R}^n$  be a nonempty, closed and convex set. Then for each vector  $x \in \mathbf{R}^n$ , its projection onto  $\Phi$  is given by

$$P_\Phi(x) = \arg \min \|x - y\| : y \in \Phi.$$

$P_\Phi : \mathbf{R}^n \rightarrow \Phi$  is referred to as the projection operator, with the nonexpansive property given by,

$$\|P_\Phi(x) - P_\Phi(y)\| \leq \|x - y\|, \quad \forall x, y \in \mathbf{R}^n,$$

for which, we can write

$$\|P_\Phi(x) - y\| \leq \|x - y\|, \quad \forall y \in \Phi. \quad (10)$$

## 3 Motivation, algorithm and global convergence of the new method

This section deals with motivation of the scheme, its algorithm and convergence analysis. First, we introduce the spectral CG method, which is an extension of the classical scheme for solving (4). By employing the spectral gradient scheme by Barzilai and Bowein [2], Birgin and Martinez [3] developed a spectral CG (SCG) method with the following search direction:

$$d_k = -\theta_k g_k + \beta_k d_{k-1},$$

where the update parameter  $\beta_k$  is defined by

$$\beta_k = \frac{(\theta_k y_{k-1} - s_{k-1})^T g_k}{d_{k-1}^T y_{k-1}},$$

and  $\theta_k$  is the spectral parameter given as

$$\theta_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}.$$

The scheme is computationally efficient, but its search directions are generally not descent directions, i.e, the scheme does not satisfy the inequality defined in (7). Over the years, three-term SCG schemes that satisfy the condition (7) have been developed. By modifying the classical PRP CG method [29, 30], Li et al. [21] proposed a spectral three-term scheme with the following search direction:

$$d_k = \begin{cases} -\gamma_k g_k + \beta_k^{MPRP} d_{k-1} - v_k y_{k-1} & k \geq 1; \\ -g_k, & \text{if } k = 0, \end{cases}$$

with

$$\beta_k^{MPRP} = \frac{g_k^T y_{k-1}}{\varphi |g_k^T d_{k-1}| + \|g_{k-1}\|^2}, \quad v_k = \frac{g_k^T d_{k-1}}{\varphi |g_k^T d_{k-1}| + \|g_{k-1}\|^2}, \quad \varphi \geq 0.$$

Simple inspection reveals that the method satisfies the sufficient descent condition (7). Also, by employing standard assumptions, the authors proved global convergence of the scheme for uniformly convex functions. Only recently, Wang et al. [36] presented a spectral three-term modification of the classical Conjugate Descent scheme [8] with search direction defined as

$$d_k = \begin{cases} -\gamma_k g_k + \frac{\|g_k\|^2 d_{k-1} - (g_k^T d_{k-1}) g_k}{\max\{-d_{k-1}^T g_{k-1}, \eta_1 \|g_k\| \|d_{k-1}\|\}} & k \geq 1; \\ -g_k, & \text{if } k = 0, \end{cases}$$

where

$$\gamma_k = \frac{\phi_k}{\min\{\eta_2 s_{k-1}^T y_{k-1}, \eta_3 \phi_k\}}, \quad \phi_k = \|s_{k-1}\|^2, \quad \eta_1 > 0, \quad \eta_2, \eta_3 \in (0, 1).$$

Now, like the classical PRP [29, 30] and HS [14] methods, the classical LS [22] CG scheme is equipped with an in-built mechanism that addresses the jamming phenomenon. So, like the others in the group, the scheme is numerically effective. However, it does not satisfy the sufficient descent condition (7). Motivated by this shortcoming of the LS scheme, the nice attributes

of three-term methods discussed above and the spectral three-term scheme by Wang et al. [36], we propose a spectral three-term adaptation of the LS scheme [22] with the following search direction:

$$d_k = \begin{cases} -\gamma_k F_k + \frac{F_k^T \bar{y}_{k-1} d_{k-1} - F_k^T d_{k-1} \bar{y}_{k-1}}{\max\{-d_{k-1}^T F_{k-1}, \zeta_1 \|\bar{y}_{k-1}\| \|d_{k-1}\|\}} & k \geq 1; \\ -F_k, & \text{if } k = 0, \end{cases} \quad (11)$$

where

$$\gamma_k = \frac{\max\{\zeta_2 \|s_{k-1}\|^2, \zeta_3 \chi_k\}}{\chi_k}, \quad \chi_k = s_{k-1}^T \bar{y}_{k-1}, \quad \zeta_1 > 0, \quad \zeta_2, \zeta_3 \in (0, 1), \quad (12)$$

and

$$\bar{y}_{k-1} = y_{k-1} + r s_{k-1}, \quad y_{k-1} = F_k - F_{k-1}, \quad r > 0.$$

Note: It can easily be deduced from (11) and (12) that

$$\max\{-d_{k-1}^T F_{k-1}, \zeta_1 \|\bar{y}_{k-1}\| \|d_{k-1}\|\} \geq \zeta_1 \|\bar{y}_{k-1}\| \|d_{k-1}\|, \quad (13)$$

and

$$\frac{\max\{\zeta_2 \|s_{k-1}\|^2, \zeta_3 \chi_k\}}{\chi_k} \geq \frac{\zeta_3 \chi_k}{\chi_k} = \zeta_3. \quad (14)$$

Next, we obtain a bound for the spectral parameter  $\gamma_k$ . To achieve that, we analyze two cases:

First case: If  $\max\{\zeta_2 \|s_{k-1}\|^2, \zeta_3 \chi_k\} = \zeta_2 \|s_{k-1}\|^2$ , then, by monotonicity of  $F$ , and definition of  $\bar{y}_{k-1}$ , we have

$$s_{k-1}^T \bar{y}_{k-1} = s_{k-1}^T y_{k-1} + r \|s_{k-1}\|^2 \geq r \|s_{k-1}\|^2. \quad (15)$$

So, using (15) we have that

$$\gamma_k = \frac{\max\{\zeta_2 \|s_{k-1}\|^2, \zeta_3 \chi_k\}}{\chi_k} = \frac{\zeta_2 \|s_{k-1}\|^2}{s_{k-1}^T \bar{y}_{k-1}} \leq \frac{\zeta_2 \|s_{k-1}\|^2}{r \|s_{k-1}\|^2} = \frac{\zeta_2}{r}.$$

Second case: If  $\max\{\zeta_2 \|s_{k-1}\|^2, \zeta_3 \chi_k\} = \zeta_3 \chi_k$ , then

$$\gamma_k = \frac{\max\{\zeta_2 \|s_{k-1}\|^2, \zeta_3 \chi_k\}}{\chi_k} = \frac{\zeta_3 \chi_k}{\chi_k} = \zeta_3.$$

Therefore, setting  $\kappa = \max\{\frac{\zeta_2}{r}, \zeta_3\}$ , we see that  $0 < \gamma_k \leq \kappa$ .

#### Algorithm 1. Modified Liu-Storey Method (MLSTM)

**Step 0:** Choose a tolerance  $\epsilon > 0$ , initial guess  $x_0 \in \Phi$ ,  $\beta > 0$ ,  $\rho \in (0, 1)$ ,  $0 < \varsigma < 2$ ,  $\sigma > 0$ . Set  $k = 0$  and  $d_0 = -F_0$ .

**Step 1:** Compute  $F(x_k)$ . If  $\|F(x_k)\| \leq \epsilon$ , stop, if not, proceed to **Step 2**.

**Step 2:** Find  $z_k = x_k + t_k d_k$ , where



$$t_k = \max\{\beta\rho^m : m = 0, 1, 2, \dots\},$$

for which

$$-F(x_k + t_k d_k)^T d_k \geq \sigma t_k \|d_k\|^2, \quad (16)$$

**Step 3:** If  $z_k \in \Phi$  and  $\|F(z_k)\| \leq \epsilon$  stop, else determine

$$x_{k+1} = P_\Phi [x_k - \varsigma \mu_k F(z_k)], \quad (17)$$

where

$$\mu_k = \frac{F(z_k)^T (x_k - z_k)}{\|F(z_k)\|^2}. \quad (18)$$

**Step 4:** Obtain the direction  $d_{k+1}$  by (11) and (12).

**Step 5:** Set  $k = k + 1$ . Go to **Step 1**.

**Lemma 1.** Let the sequence  $\{d_k\}$  be generated by (11) and (12). Then

$$F_k^T d_k \leq -\tau \|F_k\|^2, \quad \tau > 0. \quad (19)$$

*Proof.* First, from (14) we have that  $\gamma_k \geq \zeta_3$ . This implies that  $-\gamma_k \leq -\zeta_3$ . We now consider two cases:

1. For  $k = 0$ , by (11), it is obvious that  $F_0^T d_0 = -\|F_0\|^2$ .
2. For  $k \geq 1$ , from (11) and (12), we have

$$\begin{aligned} F_k^T d_k &= -\gamma_k \|F_k\|^2 + \frac{(F_k^T \bar{y}_{k-1}) d_{k-1}^T F_k - (d_{k-1}^T F_k) F_k^T \bar{y}_{k-1}}{\max\{-d_{k-1}^T F_{k-1}, \zeta_1 \|\bar{y}_{k-1}\| \|d_{k-1}\|\}} \\ &= -\gamma_k \|F_k\|^2 \\ &\leq -\zeta_3 \|F_k\|^2. \end{aligned} \quad (20)$$

Setting  $\tau = \zeta_3$ , we obtain the result in both cases.  $\square$

In the following Lemma, we prove that when the solution of (1) is not attained, i.e.,  $F(x) \neq 0$ , then a stepsize  $t_k$  exists for which (16) is satisfied.

**Lemma 2.** Suppose condition (i) in section 2 holds. Then, for every  $k \geq 0$ , there exists a positive constant  $t_k$  such that (16) is satisfied.

*Proof.* Assuming that the statement is not true. It implies that a constant  $k_0$  exists for which (16) does not hold for each integer  $m \geq 0$ , namely

$$-F(x_{k_0} + \beta\rho^m d_{k_0})^T d_{k_0} < \sigma\beta\rho^m \|d_{k_0}\|^2.$$

By employing the continuity of  $F$  with the fact that  $\rho \in (0, 1)$ , letting the integer  $m$  grow to infinity namely,  $m \rightarrow \infty$ , we obtain

$$-F(x_{k_0})^T d_{k_0} \leq 0. \quad (21)$$

From (19), we have

$$-F(x_{k_0})^T d_{k_0} \geq \tau \|F(x_{k_0})\|^2 > 0. \quad (22)$$

Clearly (21) and (22) do not agree. So, a contradiction is obtained, which establishes the proof.  $\square$

**Lemma 3.** Let the sequences  $\{x_k\}$  and  $\{z_k\}$  be generated by Algorithm 1, then

$$t_k \geq \min \left\{ \beta, \frac{\rho \tau \|F_k\|^2}{(L + \sigma) \|d_k\|^2} \right\}. \quad (23)$$

*Proof.* By (16), we see that if  $t_k = \beta$ , then (16) is satisfied. Conversely, if  $t_k \neq \beta$  then  $t_k = \frac{t_k}{\rho}$  will not satisfy (16), i.e.,

$$-F(x_k + \frac{t_k}{\rho} d_k)^T d_k < \sigma \frac{t_k}{\rho} \|d_k\|^2. \quad (24)$$

By Assumption (ii) and (19), we can write

$$\begin{aligned} \tau \|F_k\| &\leq -F_k^T d_k \\ &= (F(x_k + \frac{t_k}{\rho} d_k) - F(x_k))^T d_k - F(x_k + \frac{t_k}{\rho} d_k)^T d_k \\ &\leq L \frac{t_k}{\rho} \|d_k\|^2 - \sigma \frac{t_k}{\rho} \|d_k\|^2 \\ &= \frac{t_k}{\rho} (L + \sigma) \|d_k\|^2, \end{aligned} \quad (25)$$

which ultimately yields the desired inequality and the proof is completed.  $\square$

**Lemma 4.** Let conditions (i) and (ii) in section 2 hold. Then the sequences  $\{x_k\}$  and  $\{z_k\}$  generated by Algorithm 1 are bounded and

$$\lim_{k \rightarrow \infty} t_k \|d_k\| = 0. \quad (26)$$

*Proof.* First, we prove boundedness of the sequences  $\{x_k\}$  and  $\{z_k\}$ . Let  $\tilde{x} \in \Phi$  be a solution of (1). Then by (2), we have

$$\begin{aligned} (x_k - \tilde{x})^T F(z_k) &= (x_k - z_k + z_k - \tilde{x})^T F(z_k) \\ &= (x_k - z_k)^T F(z_k) + (z_k - \tilde{x})^T F(z_k) \\ &\geq (x_k - z_k)^T F(z_k) + (z_k - \tilde{x})^T F(\tilde{x}) \\ &= (x_k - z_k)^T F(z_k). \end{aligned} \quad (27)$$

Also from (10), (18), and the fact that  $0 < \varsigma < 2$ , we have

$$\begin{aligned}
\|x_{k+1} - \tilde{x}\|^2 &= \|P_\Phi(x_k - \varsigma\mu_k F(z_k)) - \tilde{x}\|^2 \\
&\leq \|x_k - \varsigma\mu_k F(z_k) - \tilde{x}\|^2 \\
&= \|(x_k - \tilde{x}) - \varsigma\mu_k F(z_k)\|^2 \\
&= \|x_k - \tilde{x}\|^2 - 2\varsigma\mu_k F(z_k)^T(x_k - \tilde{x}) + \varsigma^2\mu_k^2\|F(z_k)\|^2 \\
&\leq \|x_k - \tilde{x}\|^2 - 2\varsigma\mu_k F(z_k)^T(x_k - z_k) + \varsigma^2\mu_k^2\|F(z_k)\|^2 \\
&= \|x_k - \tilde{x}\|^2 - \varsigma(2 - \varsigma) \frac{(F(z_k)^T(x_k - z_k))^2}{\|F(z_k)\|^2} \\
&\leq \|x_k - \tilde{x}\|^2,
\end{aligned} \tag{28}$$

which consequently yields

$$\|x_{k+1} - \tilde{x}\| \leq \|x_k - \tilde{x}\|, \quad \forall k \geq 0. \tag{29}$$

And in a recursive manner, (29) implies that  $\|x_k - \tilde{x}\| \leq \|x_0 - \tilde{x}\|$ . So, the sequence  $\{\|x_k - \tilde{x}\|\}$  is decreasing and convergent, which means that  $\{x_k\}$  is bounded. Also, by assumption (i), (9) and (29) we have

$$\|F(x_k)\| = \|F(x_k) - F(\tilde{x})\| \leq L\|x_k - \tilde{x}\| \leq L\|x_0 - \tilde{x}\|.$$

Setting  $L\|x_0 - \tilde{x}\| = \pi$ , we obtain that

$$\|F(x_k)\| \leq \pi. \tag{30}$$

Also, from (16) and definition of  $z_k$ , we get

$$F(z_k)^T(x_k - z_k) = -t_k F(z_k)^T d_k \geq \sigma t_k^2 \|d_k\|^2 = \sigma \|x_k - z_k\|^2. \tag{31}$$

By employing (2) and the Cauchy-Schwartz inequality, we can write

$$\begin{aligned}
F(z_k)^T(x_k - z_k) &= (F(z_k) - F(x_k))^T(x_k - z_k) + F(x_k)^T(x_k - z_k) \\
&\leq \|F(x_k)\| \|x_k - z_k\|.
\end{aligned} \tag{32}$$

By (30), (31) and (32) we can write

$$\sigma \|x_k - z_k\|^2 \leq \|F(x_k)\| \|x_k - z_k\|,$$

which leads to

$$\|x_k - z_k\| \leq \frac{\pi}{\sigma}.$$

Hence, the sequence  $\{z_k\}$  is also bounded. Now, the boundedness of  $\{z_k\}$ , implies that  $\{\|z_k - \tilde{x}\|\}$  is bounded, i.e., there exists  $\pi_2 > 0$  such that for any  $\tilde{x} \in \Phi$

$$\|z_k - \tilde{x}\| \leq \pi_2. \tag{33}$$

Similarly, from (9) and (33), we have

$$\|F(z_k)\| = \|F(z_k) - F(\tilde{x})\| \leq L\|z_k - \tilde{x}\| \leq L\pi_2.$$

Hence, setting  $\pi_3 = L\pi_2$ , we obtain

$$\|F(z_k)\| \leq \pi_3. \quad (34)$$

Also, using (16), we have

$$\sigma^2 t_k^4 \|d_k\|^4 \leq t_k^2 (F(z_k)^T d_k)^2. \quad (35)$$

By combining (28) and (35), we obtain

$$\sigma^2 t_k^4 \|d_k\|^4 \leq \frac{\|F(z_k)\|^2}{\varsigma(2-\varsigma)} (\|x_k - \tilde{x}\|^2 - \|x_{k+1} - \tilde{x}\|^2). \quad (36)$$

Now, by (29) we have that the sequence  $\{\|x_k - \tilde{x}\|\}$  is convergent, and also by (34)  $\{F(z_k)\}$  is bounded. Hence, taking limits of both sides of (36) as  $k$  approaches infinity, we have

$$\sigma^2 \lim_{k \rightarrow \infty} t_k^4 \|d_k\|^4 \leq 0,$$

which consequently leads to the desired result, i.e.,

$$\lim_{k \rightarrow \infty} t_k \|d_k\| = 0. \quad (37)$$

□

**Lemma 5.** Let the sequence of search directions  $\{d_k\}$  be generated by Algorithm 1. Then  $\{d_k\}$  is bounded, namely, a constant  $\vartheta > 0$  exists such that

$$\|d_k\| \leq \vartheta, \quad \forall k \text{ positive}. \quad (38)$$

*Proof.* From (11), (13), (30), and the Cauchy-Schwartz inequality, we obtain

$$\begin{aligned} \|d_k\| &= \left\| -\gamma_k F_k + \frac{F_k^T \bar{y}_{k-1} d_{k-1} - F_k^T d_{k-1} \bar{y}_{k-1}}{\max\{-d_{k-1}^T F_{k-1}, \zeta_1 \|\bar{y}_{k-1}\| \|d_{k-1}\|\}} \right\| \\ &\leq \gamma_k \|F_k\| + \frac{\|F_k\| \|\bar{y}_{k-1}\| \|d_{k-1}\| + \|F_k\| \|\bar{y}_{k-1}\| \|d_{k-1}\|}{\max\{-d_{k-1}^T F_{k-1}, \zeta_1 \|\bar{y}_{k-1}\| \|d_{k-1}\|\}} \\ &\leq \gamma_k \|F_k\| + \frac{2\|F_k\| \|\bar{y}_{k-1}\| \|d_{k-1}\|}{\zeta_1 \|\bar{y}_{k-1}\| \|d_{k-1}\|} \\ &= \gamma_k \|F_k\| + \frac{2\|F_k\|}{\zeta_1} \\ &\leq \left( \kappa + \frac{2}{\zeta_1} \right) \|F_k\| \end{aligned}$$

$$\leq \left( \kappa + \frac{2}{\zeta_1} \right) \pi. \quad (39)$$

By setting  $\vartheta = \left( \kappa + \frac{2}{\zeta_1} \right) \pi$ , the proof is established.  $\square$

The following theorem establishes global convergence of Algorithm 1.

**Theorem 1.** Given that conditions (i) and (ii) hold. Consider the sequences  $\{x_k\}$  and  $\{z_k\}$  generated by Algorithm 1. Then

$$\liminf_{k \rightarrow \infty} \|F_k\| = 0.$$

*Proof.* For the proof, we assume that the conclusion is not true. Then it implies that a constant  $\bar{c} > 0$  exists for which

$$\|F_k\| \geq \bar{c}, \quad \forall k \geq 0. \quad (40)$$

Utilizing (19), (40) and Cauchy Schwartz inequality yields

$$\|d_k\| \geq \tau \bar{c}, \quad \forall k \geq 0. \quad (41)$$

Similarly, by employing the inequalities (23), (39), (40), (41), and for all  $k$  sufficiently large, we get

$$\begin{aligned} t_k \|d_k\| &\geq \min \left\{ \beta, \frac{\rho \tau \|F_k\|^2}{(L + \sigma) \|d_k\|^2} \right\} \|d_k\| \\ &\geq \min \left\{ \beta \tau \bar{c}, \frac{\rho \tau \bar{c}^2}{(L + \sigma) \vartheta} \right\} > 0. \end{aligned} \quad (42)$$

Clearly, the second inequality in (42) contradicts (37). Therefore, we conclude that  $\liminf_{k \rightarrow \infty} \|F_k\| = 0$ .  $\square$

## 4 Numerical experiments and discussions

Here, we investigate the effectiveness of Algorithm 1 by comparing its performance with the two methods presented in [16, 18]. The experiments with all the three algorithms were conducted using the backtracking line search (16). For the other two methods, which we label as *MLSCD* and *HCGP* for simplicity, we set the parameters as they are used in the respective papers. For Algorithm 1, we set  $\rho = 0.6$ ,  $\beta = 1$ ,  $\sigma = 10^{-3}$ ,  $\varsigma = 1.6$ ,  $r = 1$ ,  $\zeta_1 = 0.5$ ,  $\zeta_2 = 0.5$ ,  $\zeta_3 = 0.6$ . Codes for the algorithms were written using Matlab *R2014a* and run on a PC (2.30GHZ CPU, 4GB RAM). The iteration is set to stop for all the methods when the number of iterations exceed 1000 or whenever any of the following inequalities is satisfied:

$$\|F(x_k)\| \leq 10^{-8},$$

$$\|F(z_k)\| \leq 10^{-8}.$$

The following test problems were used to test the three methods.

**Problem 4.1.** This is a modification of the problem obtained from [34]. The mapping  $F$  takes the form  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_i(x) = (e^{x_i})^2 + 3 \sin x_i - 1, \quad i = 2, \dots, n-1,$$

with  $\Phi = \mathbf{R}_+^n$ .

**Problem 4.2.** Exponential Function II obtained from [19]. The mapping  $F$  takes the form  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_1(x) = e^{x_1} - 1, \quad i = 2, 3, \dots, n,$$

$$f_i(x) = \frac{i}{10} (e^{x_i} + x_{i-1} - 1),$$

with  $\Phi = \mathbf{R}_+^n$ .

**Problem 4.3.** Non-smooth Function obtained from [20]. The mapping  $F$  takes the form  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_i(x) = 2x_i - \sin |x_i|, \quad i = 1, 2, \dots, n,$$

with  $\Phi = \left\{ x \in \mathbf{R}^n : \sum_{i=1}^n x_i \leq n, \quad x_i \geq 0, \quad i = 1, 2, \dots, n \right\}$ .

**Problem 4.4.** Strictly Convex Function obtained from [34]. The mapping  $F$  takes the form  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_i(x) = e^{x_i} - 1, \quad i = 1, 2, \dots, n,$$

with  $\Phi = \mathbf{R}_+^n$ .

**Problem 4.5.** Tridiagonal Exponential Function obtained from [23]. The mapping  $F$  takes the form  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_1(x) = x_1 - e^{\left(\cos \frac{x_1+x_2}{n+1}\right)},$$

$$f_i(x) = x_i - e^{\left(\cos \frac{x_{i-1}+x_i+x_{i+1}}{n+1}\right)}, \quad i = 2, 3, \dots, n-1,$$

$$f_n(x) = x_n - e^{\left(\cos \frac{x_{n-1}+x_n}{n+1}\right)}.$$

with  $\Phi = \mathbf{R}_+^n$ .

**Problem 4.6.** Non-smooth Function obtained from [46]. The mapping  $F$  takes the form  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_i(x) = x_i - \sin |x_i - 1|, \quad i = 1, 2, \dots, n,$$

with  $\Phi = \left\{ x \in \mathbf{R}^n : \sum_{i=1}^n x_i \leq n, \quad x_i \geq 0, \quad i = 1, 2, \dots, n \right\}$ .

**Problem 4.7** The problem is obtained from [19]. The mapping  $F$  takes the form  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$\begin{aligned}
f_1(x) &= e^{x_1} - 1, \\
f_i(x) &= e^{x_i} + x_{i-1} - 1, \quad i = 2, \dots, n-1, \\
\text{with } \Phi &= \mathbf{R}_+^n.
\end{aligned}$$

**Problem 4.8** Modified version of the non-smooth Function in Problem 4.6.

The mapping  $F$  takes the form  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$\begin{aligned}
f_i(x) &= x_i - 2 \sin |x_i - 1|, \quad i = 1, 2, \dots, n, \\
\text{with } \Phi &= \left\{ x \in \mathbf{R}^n : \sum_{i=1}^n x_i \leq n, \quad x_i \geq 0, \quad i = 1, 2, \dots, n \right\}.
\end{aligned}$$

For each of the above test functions, 24 numerical experiments were performed with variables 1000, 10,000 50,000, and the following starting points:

$$\begin{aligned}
x_0^1 &= (1, 1, \dots, 1)^T, x_0^2 = (2, 2, \dots, 2)^T, x_0^3 = (3, 3, \dots, 3)^T, x_0^4 = (4, 4, \dots, 4)^T, \\
x_0^5 &= (5, 5, \dots, 5)^T, x_0^6 = (6, 6, \dots, 6)^T, x_0^7 = (7, 7, \dots, 7)^T, x_0^8 = (8, 8, \dots, 8)^T.
\end{aligned}$$

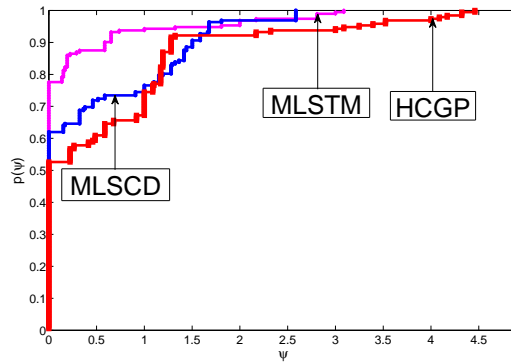


Figure 1: Performance profile with respect to number of iterations

Furthermore, we adopt Dolan and Moré [6] performance profile in order to present a graphical view of the performance of each of the three schemes considered in the experiments. In line with this, three figures were plotted with respect to three performance metrics, namely, number of iterations, function evaluations and processing time. For each figure, the *vertical-axis* corresponds to the percentage of the problems solved by any one of the algorithms with the least value of any of the metric under consideration; the right side, represents the percentage of problems solved successfully by each algorithm. Also, the topmost curve in each figure corresponds to the algorithm that solved the most problems in the experiments. It can be observed from Fig. 1, that the *MLSTM* algorithm solved 78% of problems with least number of iterations compared to the *MLSCD* and *HCGP* algorithms that recorded 62% and 54% respectively. We have to note here, that this percentage values as shown in the figure, represent sums of the percentages recorded by each algorithm with least number of iterations and the ties recorded for

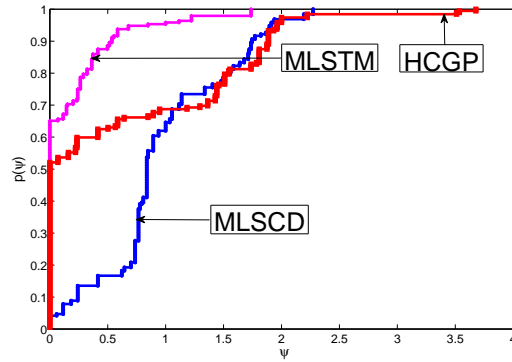


Figure 2: Performance profile with respect to function evaluation

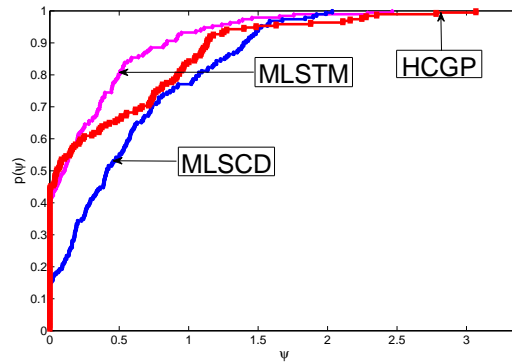


Figure 3: Performance profile with respect to CPU time

the same metric. From Fig. 2 it is observed that the *MLSTM* algorithm solved 65% of the problems with least function evaluations compared to the *HCGP* and *MLSCD* algorithms that recorded 52% and 4% respectively. As in the earlier case, here too the percentage values are sums of the values recorded by each algorithm with least function values and the ties it recorded with any one or two other algorithms. Based on least processing time metric, Fig. 3 indicated that the *HCGP* algorithm has an edge over the *MLSTM* and *MLSCD* algorithms as it solved 45% of the problems with minimum processing time, while the other two algorithms recorded 40% and 18% respectively. Moreover, it is observed that the topmost curve in all the three figures corresponds to the *MLSTM* algorithm. Hence, considering the graphical representations in Figs 1, 2, and 3, and the above analysis, it can be concluded that the *MLSTM* algorithm is more effective for solving the



problem represented in (1) than the *MLSCD* and *HCGP* algorithms.

## 5 Application of the proposed scheme

Obtaining sparse solutions to ill-conditioned linear systems of equations is the interest in most signal and image processing problems. Typically, this involves minimizing the following  $\ell_1 - \ell_2$  norm problem

$$\min_x \frac{1}{2} \|Hx - w\|_2^2 + \delta \|x\|_1, \quad (43)$$

where  $\delta$  is a nonnegative parameter,  $x \in \mathbf{R}^n$ ,  $w \in \mathbf{R}^k$  is an observed value,  $H \in \mathbf{R}^{k \times n}$  ( $k < n$ ) denotes a linear mapping, while  $\|x\|_1$  and  $\|x\|_2$  represents the  $\ell_1$  and  $\ell_2$  norms respectively. Clearly, (43) represents a convex unconstrained optimization problem.

In order to solve (43), Figueiredo et al. [7] reformulated it as a convex quadratic problem, where each vector  $x \in \mathbf{R}^n$  is split into two parts and written as

$$x = a - b, \quad a \geq 0, \quad b \geq 0, \quad a, b \in \mathbf{R}^n. \quad (44)$$

with  $a_i = (x_i)_+$ ,  $b_i = (-x_i)_+$ ,  $\forall i = 1, 2, \dots, n$  and  $(\cdot)_+ = \max\{0, x\}$ . Applying the above representation to (43), we obtain

$$\min_{a,b} \frac{1}{2} \|H(a - b) - w\|_2^2 + \delta e_n^T a + \delta e_n^T b, \quad (45)$$

where  $e_n = (1, 1, \dots, 1)^T \in \mathbf{R}^n$ . Going by Figueiredo et al. [7], the problem in (45) is reformulated as

$$\min_z \frac{1}{2} z^T A z + D^T z, \quad z \geq 0, \quad (46)$$

which is a quadratic program problem with

$$z = \begin{pmatrix} a \\ b \end{pmatrix}, \quad D = \delta e_{2n} + \begin{pmatrix} -y \\ y \end{pmatrix}, \quad y = H^T w, \quad A = \begin{pmatrix} H^T H & -H^T H \\ -H^T H & H^T H \end{pmatrix}. \quad (47)$$

In [45], the quadratic program problem in (46) was reformulated and shown to be equivalent to

$$F(z) = \min\{z, Az + D\} = 0, \quad (48)$$

where  $F$  represents a vector-valued mapping. Also, since  $F$  is monotone and Lipschitz continuous (see [28, 45]), the *MLSTM* scheme can conveniently be applied to solve it.

## 5.1 Image restoration experiment

Here, we conduct some experiments with the *MLSTM* scheme to further demonstrate its effectiveness and application in image reconstruction. For the experiments, four images are employed, which includes *Barbera*, *Lena*, *Einstein*, and *Cameraman*. As in the previous experiments, all codes are generated on MATLAB *R2014a* with the same configuration and parameter values set as applied in the earlier experiments. Also, we test performance of the *MLSTM* method with the *CGDESCENT* [46] solver, which is used in image restoration problems. The same values of the parameters used by the author were also applied in the experiments. The performance of both schemes are compared in terms of final objective function value (Obj), mean square error (MSE), signal to noise ratio (SNR), which is given by

$$SNR = 20 \times \log_{10} \left( \frac{\|x\|}{\|\tilde{x} - x\|} \right),$$

and the structural similarity index (SSIM), which computes the similarity between original image and the restored one in each of the experiments conducted. Results of the experiments conducted are presented in table 1, while Fig. 4 displays the original, blurred, and reconstructed images obtained by the *MLSTM* and *CGDESCENT* schemes. Fig. 4 reveals that the quality of reconstructed images by the *MLSTM* method for all the images considered, is somewhat better than that of *CGDESCENT* scheme. Also, table 1 showed that the *MLSTM* algorithm performed much better regarding Obj, MSE, SNR and SSIM metrics than the *CGDESCENT* scheme. However, the *CGDESCENT* scheme is much faster as it recorded less processing time than the *MLSTM* algorithm. Hence, going by these results, it can be concluded that the *MLSTM* algorithm is promising for reconstruction of the images considered.

Table 1: Image restoration results for MLSTM and CGDESCENT

IMAGE	SIZE	MLSTM					CGDESCENT				
		Obj	MSE	SNR	PT	SSIM	Obj	MSE	SNR	PT	SSIM
BARBERA	256 × 256	1.523 × 10 <sup>6</sup>	1.5267 × 10 <sup>2</sup>	20.85	10.75	0.80	1.585 × 10 <sup>6</sup>	2.0627 × 10 <sup>2</sup>	19.55	1.45	0.75
LENA	256 × 256	1.448 × 10 <sup>6</sup>	6.4566 × 10 <sup>1</sup>	24.37	17.81	0.90	1.513 × 10 <sup>6</sup>	9.0025 × 10 <sup>1</sup>	22.93	1.81	0.87
EINSTEIN	256 × 256	1.012 × 10 <sup>6</sup>	7.0780 × 10 <sup>1</sup>	21.39	10.33	0.86	1.045 × 10 <sup>6</sup>	8.7884 × 10 <sup>1</sup>	20.45	3.59	0.83
CAMERAMAN	256 × 256	1.430 × 10 <sup>6</sup>	1.4128 × 10 <sup>2</sup>	21.05	9.59	0.85	1.47 × 10 <sup>6</sup>	1.78 × 10 <sup>2</sup>	20.05	3.14	0.83

## 6 Conclusion

An efficient modified Liu-Storey scheme (*MLSTM*) for solving constrained system of monotone nonlinear equations was presented in this article. The

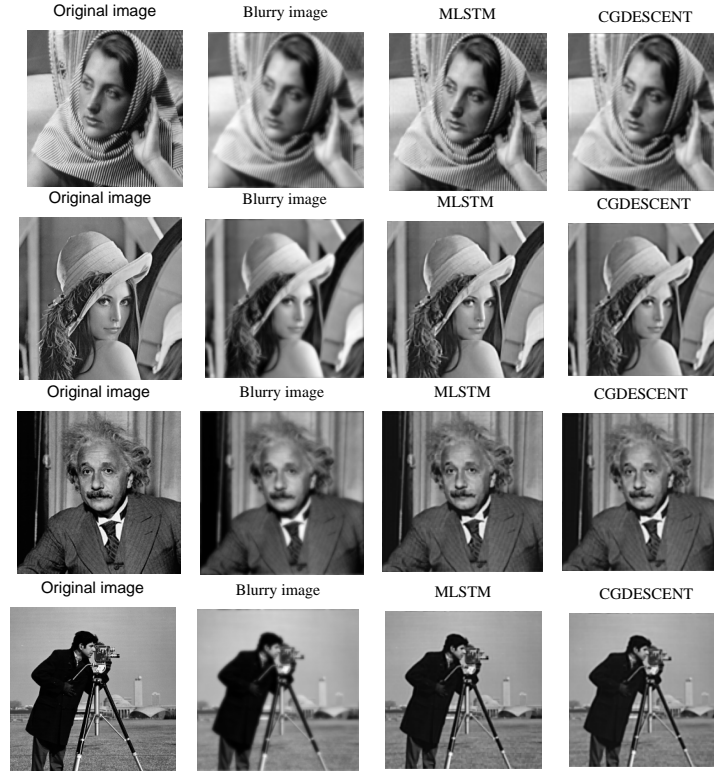


Figure 4: Original and blurred images (First and second columns from the left). Restored images by MLSTM and CGDESCENT methods (last two columns ).

scheme is ideal for large dimension problems as well as non-smooth functions because it avoids computing derivatives and requires less memory to implement. Apart from inheriting numerical efficiency of the classical  $LS$  scheme, the new method satisfies the important condition for global convergence. The scheme's global convergence was established by employing basic assumptions. Also, numerical experiments conducted with some test problems indicate that the proposed scheme is promising as it is competitive and more efficient compared to the  $MLSCD$  and  $HCGP$  methods in [16, 18]. Furthermore, an interesting novelty of the scheme is its application to solving the regularized  $\ell_1$  norm problem in compressed sensing. By conducting some experiments to recover blurry images, the scheme proved to be effective as it competes with and produces better results than the popular  $CGDESCENT$  scheme in

[46]. As a further research, we intend to explore application of the *MLSTM* scheme and its modified version to other area of interest.

**Conflict of Interest** The authors declare that they have no conflict of interest.

**Acknowledgements** The authors are grateful for the helpful and constructive comments by the anonymous reviewers and editors.

## References

- [1] Ahookhosh, M. Amini, K. and Bahrami, S. *Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations*, Numer. Algor. 64(2013), 21-42.
- [2] Barzilai, J. and Borwein, J.M. *Two point step size gradient method*, IMA J. Numer. Anal. 8(1988),141-148.
- [3] Birgin, E.G. and Martinez, J.M. *A spectral conjugate gradient method for unconstrained optimization*, Appl. Math. Optim. 43 (2001), 117-128.
- [4] Dennis, J. and Moré, J. *Quasi-Newton methods, motivation and theory*, SIAM Review, Soc. Ind. Appl. Math. 19(1)(1977), 46-89.
- [5] Dirkse, S.P. and Ferris, M.C. *A collection of nonlinear mixed complementarity problems*, Optim. Methods Softw. 5(1995)319-345.
- [6] Dolan, E.D. and Moré, J.J. *Benchmarking optimization software with performance profiles*, Math. Program. 91(2002), 201-2013.
- [7] Figueiredo, M. Nowak, R. and Wright, S.J. *Gradient projection for sparse reconstruction, application to compressed sensing and other inverse problems*, IEEE J-STSP IEEE Press, Piscataway, NJ. (2007), 586-597.
- [8] Fletcher, R. *Practical method of Optimization*, Volume 1: Unconstrained Optimization, 2nd ed., Wiley, New York, 1997
- [9] Gao, P. and He, C. *An efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints*, Calcolo 55(53)(2018),1-17.
- [10] Halilu, A.S. Majumder, A. Waziri, M.Y. Awwal, A.M. and Ahmed, K. *On solving double direction methods for convex constrained monotone nonlinear equations with image restoration*, Comput. Appl. Math. 40, 239 (2021).
- [11] Halilu, A.S. Majumder, A. Waziri, M.Y. and Ahmed, K. *Signal recovery with convex constrained nonlinear monotone equations through conjugate gradient hybrid approach*, Math. comput. Simulation, <http://doi.org/10.1016/j.matcom.2021.03.020>. (2021).

- [12] Halilu, A.S. Majumder, A. Waziri, M.Y. Ahmed, K. and Awwal, A.M. *Motion control of the two joint planar robotic manipulators through accelerated Dai-Liao method for solving system of nonlinear equations*, Eng. Comput. <https://doi.org/10.1108/EC-06-2021-0317>
- [13] Hager, W.W. and Zhang, H. *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM J. Optim. 16 (2005), 170-192.
- [14] Hestenes, M.R. and Stiefel, E.L. *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49(1952), 409-436.
- [15] Hively, G.A. *On a class of nonlinear integral equations arising in transport theory*, SIAM J. Numer. Anal. 9 (1978), 787-792.
- [16] Ibrahim, A.H. Deepho, J. Abubakar, A.B. Aremu, K.O. *A Modified Liu-Storey-conjugate descent hybrid projection method for convex constrained nonlinear equations and image restoration*, Numer. Alg. Cont. Optim. doi:10.3934/naco.2021022.
- [17] Kelly, C. *Iterative methods for optimization*, Frontiers Appl. Math. 1999, DOI:10.1137/1.9781611970920 Corpus ID: 123596970.
- [18] Koorapetse, M. Kaelo, P. *An efficient hybrid conjugate gradient-based projection method for convex constrained nonlinear monotone equations*, J. inter. math. 22 (6)(2019), 1031-1050.
- [19] La Cruz, W. Martínez, J.M. and Raydan, M. *Spectral residual method without gradient information for solving large-scale nonlinear systems*, Theory and Experiments, Citeseer, Technical Report RT-04-08(2004).
- [20] La Cruz, W. *A Spectral algorithm for large-scale systems of nonlinear monotone equations*, Numer. Algor. DOI 10.1007/s1107s-017-0299-8. (2017).
- [21] Li, X. Shi, J. Dong, X. and Yu, J. *A new conjugate gradient method based on Quasi-Newton equation for unconstrained optimization*, J. Comput. Appl. Math. <https://doi.org/10.1016/j.cam.2018.10.035>
- [22] Liu, Y. and Storey, C. *Efficient generalized conjugate gradient algorithms*, Part 1: Theory, J. Optim. Theory Appl. 69(1991), 129-137.
- [23] Liu, J.K. and Li, S.J. *A projection method for convex constrained monotone nonlinear equations with applications*, Comput. Math. Appl. 70(10)(2015), 2442-2453.
- [24] Meintjes, K. and Morgan, A.P. *A methodology for solving chemical equilibrium systems*, Appl. Math. Comput. 22(1987), 333-361.

- [25] Mompoti, S. and Kaelo, P. *Globally convergent three-term conjugate gradient projection methods for solving nonlinear monotone equations*, Arab. J. Math. 7(2018), 289-301.
- [26] Muhammad, L. and Waziri, M.Y. *An Alternative three-term conjugate gradient algorithm for systems of nonlinear equations*, Intern. J. Math. Model. Comput. 07(02)(2017),145-157.
- [27] Nakamura, W. Narushima, Y. and Yabe, H. *Nonlinear conjugate gradient methods with sufficient descent properties for unconstrained optimization.*, J. Ind. Manag. Optim. 9(3)(2013),595-619.
- [28] Pang, J.S. *Inexact Newton methods for the nonlinear complementarity problem*, Math. Program. 36 (1986), 54-71.
- [29] Polak, E. and Ribière, G. *Note Sur la convergence de directions conjuguées*, Rev. Francaise Informat. Recherche Operationelle, 3e Année, 16 (1969), 35-43.
- [30] Polyak, B.T. *The conjugate gradient method in extreme problems*, USSR Comp. Math. Math. Phys. 9 (1969), 94-112.
- [31] Sabi'u, J. Shah, A. Waziri, M.Y. and Ahmed, K. *Modified Hager-Zhang conjugate gradient methods via singular value analysis for solving monotone nonlinear equations with convex constraint*, Int. J. Comput. Methods. [hptt://doi.org/10.1142/S0219876220500437](https://doi.org/10.1142/S0219876220500437) (2020).
- [32] Solodov, M.V. and Svaiter, B.F. *A globally convergent inexact Newton method for systems of monotone equations*, in: Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Kluwer Academic Publishers, 1998, pp. 355-369.
- [33] Sugiki, K. Narushima, Y. and Yabe, H. *Globally convergent three-term conjugate gradient methods that use secant conditions and generate descent search directions for unconstrained optimization*, J. Optim. Theory Appl. 153(2012), 733-757.
- [34] Wang, C. Wang, Y. and Xu, C. *A projection method for a system of nonlinear monotone equations with convex constraints*, Math. Methods Oper. Res. 66(1)(2007), 33-46.
- [35] Wang, X.Y. Li, X.J. and Kou, X.P. *A self-adaptive three-term conjugate gradient method for monotone nonlinear equations with convex constraints*, Calcolo DOI 10.1007/s10092-015-0140-5.
- [36] Wang, Z. Li, P. Li, X. and Pham, H. *A modified three-term type CD conjugate gradient algorithm for unconstrained optimization problems*, Hindawi Mathematical Problems in Engineering Volume 2020, Article ID 4381515, 14 pages <https://doi.org/10.1155/2020/4381515>.

- [37] Waziri, M.Y. Ahmed, K. and Sabi'u, J. *A family of Hager-Zhang conjugate gradient methods for system of monotone nonlinear equations*, Appl. Math. Comput. 361(2019), 645-660.
- [38] Waziri, M.Y. Ahmed, K. and Sabi'u, J. *A Dai-Liao conjugate gradient method via modified secant equation for system of nonlinear equations*, Arab. J. Math. 9(2020), 443-457.
- [39] Waziri, M.Y. Ahmed, K. and Sabi'u, *Descent Perry conjugate gradient methods for systems of monotone nonlinear equations*, Numer. Algor. 85(2020), 763-785.
- [40] Waziri, M.Y. Ahmed, K. Sabi'u, J. and Halilu, A.S. *Enhanced Dai-Liao conjugate gradient methods for systems of monotone nonlinear equations*, SeMA J. 78(2020), 15-51.
- [41] Waziri, M.Y. Usman, H. Halilu, A.S. and Ahmed, K. *Modified matrix-free methods for solving systems of nonlinear equations*, Optimization. 70(2021), 2321-2340
- [42] Waziri, M.Y. and Ahmed, K. *Two descent Dai-Yuan conjugate gradient methods for systems of monotone nonlinear equations*, J. Sci. Comput. (2022) 90:36. <https://doi.org/10.1007/s10915-021-01713-7>.
- [43] Waziri, M.Y. Ahmed, K. Halilu, A.S. Awwal, A.M. *Modified Dai-Yuan iterative scheme for nonlinear systems and its Application*, Numer. Alg. Control Optim. doi:10.3934/naco.2021044.
- [44] Waziri, M.Y. Ahmed, K. and Halilu, A.S. *Adaptive three-term family of conjugate residual methods for system of monotone nonlinear equations*, Sao Paulo J. Math. Sci. <https://doi.org/10.1007/s40863-022-00293-0>
- [45] Xiao, Y. Wang, Q. and Hu, Q. *Non-smooth equations based method for  $l_1$ -norm problems with applications to compressed sensing*, Nonlinear Anal. Theory Methods Appl. 74(11)(2011), 3570-3577.
- [46] Xiao, Y. and Zhu, H. *A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing*, J. Math. Anal. Appl. 405(1)(2013), 310-319.
- [47] Yuan, G. *Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems*, Optim. Lett. 3(2009), 11-21.

#### How to cite this article

Kiri, A.I., Waziri, M.Y. and Ahmed, K., A modified Liu-Storey scheme for nonlinear systems with an application to image recovery. *Iran. j. numer. anal. optim.*, 2023; 13(1): 38-58. <https://doi.org/10.22067/ijnao.2022.75413.1107>.



## An improvised technique of quintic hermite splines to discretize generalized Burgers–Huxley type equations

I. Kaur, S. Arora\* and I. Bala

### Abstract

A mathematical collocation solution for generalized Burgers–Huxley and generalized Burgers–Fisher equations has been monitored using the weighted residual method with Hermite splines. In the space direction, quintic Hermite splines are introduced, while the time direction is discretized using a finite difference approach. The technique is determined to be unconditionally stable, with order  $(h^4 + \Delta t)$  convergence. The technique's efficacy is tested using nonlinear partial differential equations. Two problems of the generalized Burgers–Huxley and Burgers–Fisher equations have been solved using a finite difference scheme as well as the quintic Hermite collocation method (FDQHCM) with varying impacts. The FDQHCM computer codes are written in MATLAB without transforming the nonlinear term to a linear term. The numerical findings are reported in weighted norms and in discrete form. To assess the technique's applicability, numerical and exact values are compared, and a reasonably good agreement is recognized between the two.

**AMS subject classifications (2020):** 65D07; 65N06; 65N35; 65N12

\*Corresponding author

Received 16 March 2022; revised 28 April 2022; accepted 6 May 2022

Inderpreet Kaur

Chitkara University Institute of Engineering and Technology, Department of Applied Sciences, Chitkara University, Patiala, Punjab, INDIA. e-mail: [inderpreet.kaurchittkara.edu.in](mailto:inderpreet.kaurchittkara.edu.in)

Shelly Arora

Department of Mathematics, Punjabi University, Patiala, Punjab-147002, INDIA. e-mail: [aoshellygmail.com](mailto:aoshellygmail.com)

Indu Bala

Department of Mathematics, Punjabi University, Patiala, Punjab-147002, INDIA. e-mail: [indu13121994gmail.com](mailto:indu13121994gmail.com)



**Keywords:** Quintic Hermite splines, Forward finite difference scheme, collocation method, stability analysis.

## 1 Introduction

Nonlinear partial differential equations nowadays turn out to be basic mathematical methodologies to study multifarious structures like turbulence in fluid dynamics, convection-diffusion, flow through a shock wave roaming in viscid fluid, number theory, continuous stochastic processes, and so on. The diversity of physical phenomena in basic and applied sciences such as physics, chemistry, biology, computer science, electronics, and so on can be preeminently described by these nonlinear equations. Various authors have used these nonlinear partial differential equations in different fields such as hydrodynamics, solid mechanics, and so on; see [7, 9, 10, 11, 18, 25]. The Navier–Stokes equation is a fundamental fluid dynamics equation that may be simplified into a number of mathematical phenomena. By omitting the pressure factor, the generalized Burgers–Huxley equation (GBHE) simplifies this intricate equation. This equation explains how reaction, diffusion, and convection processes interact. Special instances of the GBHE include generalized Burgers–Fisher equation (GBFE) and generalized Burgers equation. These equations are widely studied in fluid dynamics, gas dynamics, traveling wave solutions, and traffic flow. The one-dimensional GBHE, which refers to nerve pulse transmission in nerve fibers and waves in fluid crystals, can be written in the following form:

$$\frac{\partial u}{\partial t} = \varepsilon \frac{\partial^2 u}{\partial x^2} - \mu u^\delta \frac{\partial u}{\partial x} + f(u), \quad (x, t) \in \Omega \times (0, T]. \quad (1)$$

The initial and boundary conditions are presented as

$$u(x, 0) = g(x), \quad (2)$$

$$u(0, t) = f_1(t), \quad \text{and} \quad u(1, t) = f_2(t), \quad (3)$$

where  $g(x)$ ,  $f_1(t)$ , and  $f_2(t)$  are continuous functions in  $x$  and  $t$ , respectively. Moreover,  $f(u)$  is a smooth function with a nonlinear nature that is defined on  $\Omega \times (0, T)$ . In the theory of traveling wave solutions, it is significant. For  $f(u) = 0$ , equation (1) reduced to modified Burgers equation. On the other hand, in [24], it has studied the generalized Burgers equation with  $f(u) \neq 0$ , although in a different way than Burgers–Huxley and Burgers–Fisher. For the GBHE,  $f(u) = \beta u(1-u^\delta)(u^\delta - \bar{\gamma})$  and for the GBFE,  $f(u) = \beta u(1-u^\delta)$ , where  $\mu, \delta, \beta$ , and  $\bar{\gamma}$  are real parameters. The viscosity factor  $\varepsilon$  is another distinction between the Burgers equation and Burgers–Huxley equation. The viscosity factor is typically assumed to be 1 in the Burgers–Huxley equation. However, it plays a vital role in turbulence theory for the Burgers equation. A specific nonlinear evolution equation is represented by the change in parameters.

For example, equation (1) presents the modified Burgers equation to explain wave propagation in nonlinear dissipative systems and various other physical contexts, such as sound waves in a viscous medium, where  $\beta = 0$ . Therefore, a significant study of GBHE with different case studies helps to analyze the behavior of different nonlinear equations from a wide perspective.

Due to the wide applications of GBHE and GBFE, these equations are studied extensively by many investigators. Hammad and El-Azab [13] have used a collocation method with a  $2N$ -order compact finite difference scheme, and Kushner and Matviychuk [17] proposed finite-dimensional dynamics to find the exact solution of Burgers–Huxley equation. Celik [8] proposed the Haar wavelet method, whereas Alharbi and Fahmy [1] have proposed an ADM-pade method to study the behavior of Burgers–Huxley equation. Javidi [16] followed a spectral collocation method to solve the GBHE, whereas Saha Ray and Gupta [21] followed a wavelet collocation for the same. The implicit exponential finite difference method has been followed in [14] to find the numerical solution to Burgers–Huxley equation.

To solve the GBHE and GBFE, the present work proposes a quintic Hermite collocation with the forward finite difference technique (FDQHCM). Quintic Hermite collocation is a weighted residual approach with a Hermite basis, whereas finite difference scheme is the variational method. Both approaches are coupled to get numerical outputs that are compatible with numerical codes and are stable.

## 2 Quintic Hermite collocation

Quintic Hermite collocation method [4, 6, 5] is one of the weighted residual methods. Instead of Jacobi orthogonal polynomials, quintic Hermite interpolating polynomials are used as the foundation function in this approach. Over the area  $\Omega$ , which is considered to be the union of intervals  $[x_{i-1}, x_i]$ , the solution function is approximated by an approximation function that uses quintic Hermite interpolating polynomials as its basis.

Hermite interpolating polynomials, which are an extension of Lagrangian interpolating polynomials, are of order  $2k + 1$ , where  $k$  is a positive integer. Hermite interpolating polynomials, on the other hand, are superior to Lagrangian interpolating polynomials in terms of applicability since they interpolate both the function and its  $k$ th-order derivative. Furthermore, Lagrangian interpolating polynomials need a requirement of continuity at node locations that are not required by Hermite interpolating polynomials. As a result, quintic Hermite interpolating polynomials with  $k = 2$  yield quintic Hermite interpolating polynomials that interpolate the function as well as its first- and second-order derivatives. Arora and Kaur [5] discussed the behavior and structure of quintic Hermite polynomials in depth. Let  $u^\gamma(x, t)$  be the approximating function to be adjusted to eq. (1). Although the first- and second-order derivatives are interpolated by quintic Hermite interpolat-

ing polynomials, the boundary constraints are satisfied at boundary points. The approximating function is

$$u(x, t) = \sum_{i=1}^2 \left( P_i(x) a_i(t) + \bar{P}_i(x) b_i(t) + \bar{\bar{P}}_i(x) c_i(t) \right), \quad (4)$$

where  $a_i, b_i$ , and  $c_i$ 's are continuous functions of  $t$  and  $P_i, \bar{P}_i$ , and  $\bar{\bar{P}}_i$  are smooth functions of  $x$  and expressed as

$$P_i(x) = \begin{cases} 6 \left( \frac{x_{j+1}-x}{x_{j+1}-x_j} \right)^5 - 15 \left( \frac{x_{j+1}-x}{x_{j+1}-x_j} \right)^4 + 10 \left( \frac{x_{j+1}-x}{x_{j+1}-x_j} \right)^3, & x_j \leq x \leq x_{j+1}, \\ 6 \left( \frac{x-x_{j-1}}{x_j-x_{j-1}} \right)^5 - 15 \left( \frac{x-x_{j-1}}{x_j-x_{j-1}} \right)^4 + 10 \left( \frac{x-x_{j-1}}{x_j-x_{j-1}} \right)^3, & x_{j-1} \leq x \leq x_j, \\ 0 & \text{elsewhere,} \end{cases}$$

$$\bar{P}_i(x) = \begin{cases} 3 \frac{(x_{j+1}-x)^5}{(x_{j+1}-x_j)^4} - 7 \frac{(x_{j+1}-x)^4}{(x_{j+1}-x_j)^3} + 4 \frac{(x_{j+1}-x)^3}{(x_{j+1}-x_j)^2}, & x_j \leq x \leq x_{j+1}, \\ -3 \frac{(x-x_{j-1})^5}{(x_j-x_{j-1})^4} + 7 \frac{(x-x_{j-1})^4}{(x_j-x_{j-1})^3} - 4 \frac{(x-x_{j-1})^3}{(x_j-x_{j-1})^2}, & x_{j-1} \leq x \leq x_j, \\ 0 & \text{elsewhere,} \end{cases}$$

$$\bar{\bar{P}}_i(x) = \begin{cases} 0.5 \frac{(x_{j+1}-x)^5}{(x_{j+1}-x_j)^3} - \frac{(x_{j+1}-x)^4}{(x_{j+1}-x_j)^2} + 0.5 \frac{(x_{j+1}-x)^3}{(x_{j+1}-x_j)}, & x_j \leq x \leq x_{j+1}, \\ 0.5 \frac{(x-x_{j-1})^5}{(x_j-x_{j-1})^3} - \frac{(x-x_{j-1})^4}{(x_j-x_{j-1})^2} + 0.5 \frac{(x-x_{j-1})^3}{(x_j-x_{j-1})}, & x_{j-1} \leq x \leq x_j, \\ 0 & \text{elsewhere,} \end{cases}$$

where

$$\begin{aligned} P_i(x_j) &= \delta_{ji}, & P'_i(x_j) &= 0, & P''_i(x_j) &= 0, & i, j &= 1, 2, \dots, 6, \\ \bar{P}_i(x_j) &= 0, & \bar{P}'_i(x_j) &= \delta_{ji}, & \bar{P}''_i(x_j) &= 0, & i, j &= 1, 2, \dots, 6, \\ \bar{\bar{P}}_i(x_j) &= 0, & \bar{\bar{P}}'_i(x_j) &= 0, & \bar{\bar{P}}''_i(x_j) &= \delta_{ji}, & i, j &= 1, 2, \dots, 6. \end{aligned}$$

The points  $x_j$ 's are the node points or mesh points, which are taken equidistant to get uniform mesh grid. The principle of collocation is applied between two consecutive node points, that is, on  $[x_j, x_{j+1}]$ . The details are given in [2, 4, 5]. Therefore, to apply the principle of collocation, a new variable  $\xi$  is introduced within each sub-interval  $[x_j, x_{j+1}]$  in such a way that as  $x$  varies from  $x_j$  to  $x_{j+1}$ ,  $\xi$  varies from 0 to 1, and  $h$  is the length of sub-interval  $[x_j, x_{j+1}]$ . To reduce the complexity of system of equations, the interval length  $h$  is taken to be uniform. Therefore, after rearranging  $P_i, \bar{P}_i$ , and  $\bar{\bar{P}}_i$ , quintic Hermite polynomials take the form as given in Table 1.

Table 1: Presentation of quintic Hermite splines

$i$	$H_i$	$H'_i$	$H''_i$
1	$1 - 10\xi^3 + 15\xi^4 - 6\xi^5$	$-30\xi^2 + 60\xi^3 - 30\xi^4$	$-60\xi + 180\xi^2 - 120\xi^3$
2	$h(\xi - 6\xi^3 + 8\xi^4 - 3\xi^5)$	$h(1 - 18\xi^2 + 32\xi^3 - 15\xi^4)$	$h(-36\xi + 96\xi^2 - 60\xi^3)$
3	$\frac{h^2}{2}(\xi^2 - 3\xi^3 + 3\xi^4 - \xi^5)$	$\frac{h^2}{2}(2\xi - 9\xi^2 + 12\xi^3 - 5\xi^4)$	$\frac{h^2}{2}(2 - 18\xi + 36\xi^2 - 20\xi^3)$
4	$\frac{h^2}{2}(\xi^3 - 2\xi^4 + \xi^5)$	$\frac{h^2}{2}(3\xi^2 - 8\xi^3 + 5\xi^4)$	$\frac{h^2}{2}(6\xi - 24\xi^2 + 20\xi^3)$
5	$(10\xi^3 - 15\xi^4 + 6\xi^5)$	$(30\xi^2 - 60\xi^3 + 30\xi^4)$	$(60\xi - 180\xi^2 + 120\xi^3)$
6	$h(-4\xi^3 + 7\xi^4 - 3\xi^5)$	$h(-12\xi^2 + 28\xi^3 - 15\xi^4)$	$h(-24\xi + 84\xi^2 - 60\xi^3)$

The values of quintic Hermite splines at  $x = 0$  and  $x = 1$  are given in Tables 2 and 3. It is analyzed from these tables that these polynomial are either 0 or 1 at these points, which helps the approximating function to satisfy the Dirichlet or Neumann type boundary conditions. These quintic Hermite splines have the properties  $H_1(\xi) = H_5(1 - \xi)$ ,  $H_2(\xi) = -H_6(1 - \xi)$ , and  $H_3(\xi) = H_4(1 - \xi)$ .

Table 2: Presentation of quintic Hermite splines and its corresponding first- and second-order derivatives at  $\xi=0$

$H_1$	1	$H'_1$	0	$H''_1$	0
$H_2$	0	$H'_2$	$h$	$H''_2$	0
$H_3$	0	$H'_3$	0	$H''_3$	$h^2$
$H_4$	0	$H'_4$	0	$H''_4$	0
$H_5$	0	$H'_5$	0	$H''_5$	0
$H_6$	0	$H'_6$	0	$H''_6$	0

Table 3: Presentation of quintic Hermite splines and its corresponding first- and second-order derivatives at  $\xi=1$

$H_1$	0	$H'_1$	0	$H''_1$	0
$H_2$	0	$H'_2$	0	$H''_2$	0
$H_3$	0	$H'_3$	0	$H''_3$	0
$H_4$	0	$H'_4$	0	$H''_4$	$h^2$
$H_5$	1	$H'_5$	0	$H''_5$	0
$H_6$	0	$H'_6$	$h$	$H''_6$	0

Therefore, equation (4) can be rewritten as

$$u^\gamma(\xi, t) = \sum_{m=1}^6 H_m(\xi) a_m^\gamma(t), \quad (5)$$

where  $H_m(\xi)$ 's are quintic Hermite interpolating polynomials and  $a_m^\gamma(t)$ 's are continuous functions of  $t$  with  $\gamma$  being the number of sub-divisions. To apply orthogonal collocation, zeros of fourth-order shifted Legendre polynomials have been taken as collocation points. These polynomials give better results on center as well as on average [2]. Therefore, the zeros of these polynomials have been chosen as collocation points. The diagrammatic representation of these splines is given in Figure 1. In Figure 2, the application of quintic Hermite splines has been shown. Further details of collocation points and the technique of quintic Hermite collocation are given in [5, 2].

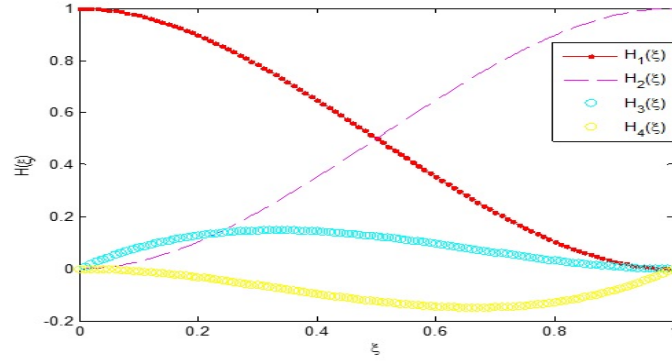


Figure 1: Diagrammatic behavior of quintic Hermite polynomials

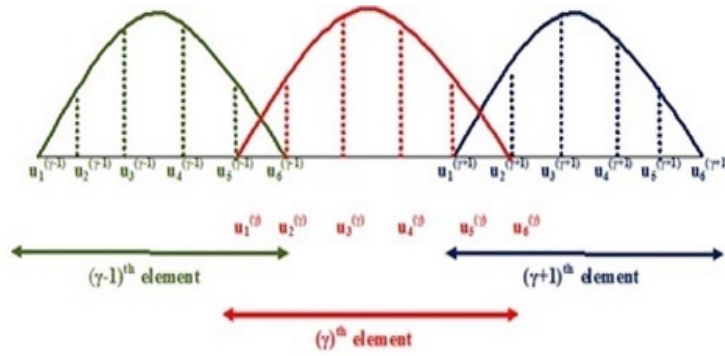


Figure 2: Diagrammatic representation of quintic Hermite collocation scheme

### 3 Implementation of FDQHCM

To implement the scheme of FDQHCM, equation (1) is discretized in time direction using forward finite difference scheme with step-size  $\Delta t$

$$\frac{u_{j+1} - u_j}{\Delta t} = u_{xxj} - \mu u_j^\delta u_{xj} + f_j, \quad (6)$$

where  $f_j$  is  $f(u_j)$ . Initial and boundary conditions can be discretized in the following way at  $t = t_j$ ,  $u_0(x) = g(x)$ ;  $u_j(0) = f_{1j}$  and  $u_j(1) = f_{2j}$ , where  $f_{1j} = f_1(t_j)$  and  $f_{2j} = f_2(t_j)$ . By simplifying, equation (6) converts into

$$u_{j+1} = \Delta t u_{xxj} - \Delta t \mu u_j^\delta u_{xj} + \Delta t f_j + u_j. \quad (7)$$

After applying quintic Hermite collocation on the variable  $u_j$  in the space direction, equation (6) takes the following form:

$$\begin{aligned} \sum_{m=1}^6 H_m(\xi) a_m^\gamma(t_{j+1}) &= \frac{\Delta t}{h^2} \sum_{m=1}^6 H_m''(\xi) a_m^\gamma(t_j) - \mu \frac{\Delta t}{h} \left( \sum_{m=1}^6 H_m(\xi) a_m^\gamma(t_j) \right)^\delta \\ &\quad \times \left( \sum_{m=1}^6 H_m'(\xi) a_m^\gamma(t_j) \right) + \sum_{m=1}^6 H_m(\xi) a_m^\gamma(t_j) + \Delta t f_j, \\ &\quad m = 1, 2, \dots, 6, \quad j = 1, 2, \dots, n_t. \end{aligned} \quad (8)$$

At the  $k$ th collocation point, equation (8) can be written as

$$\begin{aligned} \sum_{m=1}^6 H_{km} a_{m,j+1}^\gamma &= \frac{\Delta t}{h^2} \sum_{m=1}^6 B_{km} a_{m,j}^\gamma - \mu \frac{\Delta t}{h} \left( \sum_{m=1}^6 H_{km} a_{m,j}^\gamma \right)^\delta \times \sum_{m=1}^6 A_{km} a_{m,j}^\gamma \\ &\quad + \sum_{m=1}^6 H_{km} a_{m,j}^\gamma + \Delta t f_{k,j}, \\ &\quad m = 1, 2, \dots, 6, \quad k = 2, 3, 4, 5, \quad j = 1, 2, \dots, n_t, \end{aligned} \quad (9)$$

where  $H_{km}$  is the  $m$ th interpolating polynomial at  $k$ th collocation point and  $B_{km}$  and  $A_{km}$  are, respectively, the second-order and the first-order derivatives of  $m$ th interpolating polynomial at  $k$ th collocation point. Also,  $a_{m,j}^\gamma$  is the  $m$ th collocation function at  $j$ th time step in  $\gamma^{th}$  sub-domain and  $f_{k,j}$  is the discretized function  $f(u)$  at the  $j$ th time step and the  $k$ th collocation point.

Hence, after implementation of quintic Hermite collocation with forward finite difference scheme, collocation equations reduce to the following matrix form:

$$H \bar{a}_{j+1} = M_1 \bar{a}_j + (H \bar{a}_j)^\delta (M_2 \bar{a}_j) + \bar{f} + K, \quad (10)$$

where  $H = [H_m(\xi_k)]$ ,  $M_1 = [\frac{\Delta t}{h^2} B_m(\xi_k) + H_m(\xi_k)]$ , and  $M_2 = [\frac{\Delta t}{h} A_m(\xi_k)]$ . Moreover,  $K$  is obtained from the boundary conditions and  $\bar{f} = [\Delta t f_{k,j}]$ . Also,  $\bar{a}_{j+1}$  and  $\bar{a}_j$  being the unknown collocation vectors at the  $(j+1)$ th and  $j$ th time steps, respectively. In collective form, left-hand side coefficient matrix can be written as

$$J = M_1 \bar{a}_j + (H \bar{a}_j)^\delta (M_2 \bar{a}_j) + \bar{f} + K$$

and  $\bar{A} = [\bar{a}_{j+1}]$ .

Hence, in the combined form, equation (10) can be written as

$$\bar{A} = H^{-1} J. \quad (11)$$

Details of these matrices are given in [5, 3]. In the case of the first and last elements, the bandwidth is  $4 \times 5$ , whereas for the remaining elements, bandwidth is  $4 \times 6$ . Therefore, after combining all the collocation equations, a set of  $4ne \times 4ne$  equations appear, with  $ne$  being the total number of subdivisions. The discretized set of equations has been solved in MATLAB. For initial approximation of  $\bar{a}_0$ , the initial condition of  $u$  at  $t = 0$  has been taken at different collocation points. Then the loop was introduced to calculate  $\bar{a}_1$ ,  $\bar{a}_2$ , and so on upto desired accuracy. The details of algorithm are mentioned hereunder.

**Algorithm:**

The algorithm of the solution technique is

- (i) Define the problem.
  - (ii) Apply the finite difference technique in the time direction.
  - (iii) Discretize the problem by applying quintic Hermite interpolating polynomials in space direction.
  - (iv) Apply collocation points on the interpolating equations.
  - (v) Solve the collocation equations using the following code.
- ```

for ii = 1 : 1 : ti/j
P(:,ii) = B - mu * A.*(Udelta) + U + beta * j.*U.*(1 - (Udelta)).*((Udelta) - gamma);
Z(:,ii) = inv(H) * P(:,ii);
U0 = Z(:,ii);
end

```

## 4 Stability analysis

To study the stability of any partial differential equation, it is necessary that it should be linear. In the case of nonlinear partial differential equations, it is first linearized, and then the stability behavior is analyzed. Therefore, to study the stability of equation (1), it is first linearized by taking  $\bar{v} = \max(u^\delta)$  and  $f(u) = \beta u \bar{f}$ , where  $\bar{f} = \max((u^\delta - 1)(\gamma - u^\delta))$ , for all  $(x, t) \in \Omega \times (0, T)$ . As given in [20], a method is said to be  $A$ -stable if the region of absolute stability includes the region of  $Re(\lambda \Delta t) < 0$ .

The region of absolute stability for FDQHCM is the set of all complex numbers  $\lambda \Delta t$  such that

$$|u_j| \leq C \quad \text{as} \quad t_j \rightarrow \infty. \quad (12)$$

Let  $\lambda_j$ ,  $j = 1, 2, \dots, ne$ , be the set of eigenvalues of the coefficient matrix  $a_{m,j}^\gamma$ . The scheme of FDQHCM is said to be stable if  $Re(\lambda_j, \Delta t) < 0$  for all values of  $\mu, \beta, \gamma$ , and  $\delta$ . Then

$$\begin{aligned} \sum_{m=1}^6 H_m(\xi) a_{m,j+1}^\gamma &= \frac{\Delta t}{h^2} \sum_{m=1}^6 H_m''(\xi) a_{m,j}^\gamma - \mu \frac{\Delta t}{h} \bar{v} \sum_{m=1}^6 H_m'(\xi) a_{m,j}^\gamma \\ &+ \sum_{m=1}^6 H_m(\xi) a_{m,j}^\gamma + \Delta t \beta \sum_{m=1}^6 H_m(\xi) a_{m,j}^\gamma \bar{f}. \end{aligned} \quad (13)$$

The stability of the given technique can be checked on a linear equation. Therefore, equation (13) represents the quasi-linearized form of equation (8) to check the stability of the given technique. From Figure 3, it is observed that  $Re(\lambda_j, \Delta t) \in [-1, 0]$  for all values of  $\mu, \beta, \gamma$ , and  $\delta$ , which justifies the stability of the numerical scheme.

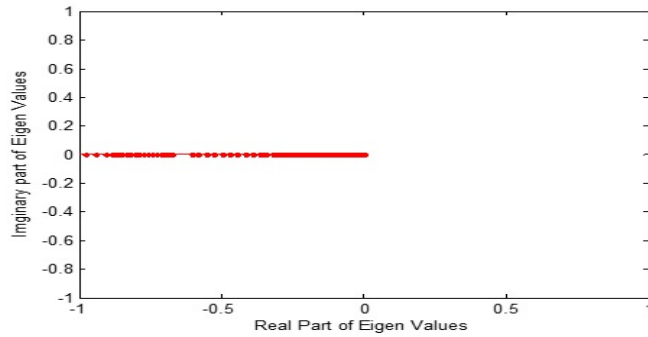


Figure 3: Behavior of eigenvalues for coefficient matrix given in equation (13)

## 5 Convergence analysis

The operator  $\mathfrak{L}$  defined by  $\frac{\partial^2}{\partial x^2}$  in spatial and time domains, is positive definite in  $L_2(0, 1)$ , the space of all real valued Lebesgue measurable functions square integrable  $(0, 1)$ , for all  $t > 0$ . The definition given by [19] is quoted here:

*Consider a family of mathematical problems parametrized by singular perturbation parameter  $\varepsilon$ , where  $\varepsilon$  lies in the semi open interval  $0 < \varepsilon \leq 1$ . Assume that each problem in the family has the unique solution denoted by  $u_\varepsilon$  and that each  $u_\varepsilon$  is approximated by a sequence of numerical solutions  $(U_\varepsilon, \bar{\Omega}^N)_{N=1}^\infty$ , where  $U_\varepsilon$  is defined on the  $\bar{\Omega}^N$ , representing the set of points in  $\mathbf{R}$  and  $N$  is the discretization parameter. Then the numerical solutions  $U_\varepsilon$  are said to converge to the exact solution  $u_\varepsilon$ , if there exist a positive integer  $N_0$  and positive numbers  $C$  and  $p$ , where  $N_0$ ,  $C$ , and  $p$  are all independent of  $N$  and  $\varepsilon$ , such that for all  $N \geq N_0$ , we have*

$$\sup_{0 < \varepsilon \leq 1} \|U_\varepsilon - u_\varepsilon\|_{\bar{\Omega}^N} \leq CN^{-p}.$$



Here  $p$  is the rate of convergence and  $C$  is the error constant.

**Theorem 1** (Maximum Principle). [19] Let  $u(x)$  be a solution of an advection diffusion equation with  $u(0) \geq 0$  and  $u(1) \geq 0$ . Then  $\mathbb{L}(u(x)) \geq 0$  for all  $x$  on the domain  $\Omega$ ,  $\mathbb{L}$  be the operator. Hence  $u(x) \geq 0$  for all  $x$  in  $\bar{\Omega}$

**Lemma 1.** [5, 3] Let  $\mathbf{H}$  be the space of all Hermite interpolating polynomials of order 5 defined on the interval  $0 \leq x \leq 1$ . Then

$$\sum_{m=1}^6 |H_m(x)| \leq 6, \quad \text{for all } x \in [0, 1].$$

**Theorem 2.** [12] Let  $\mathbf{H}$  be the space of all quintic Hermite interpolating polynomials  $H$  of function  $u(x)$  defined on  $[a, b]$ . Then the rate of convergence of quintic Hermite interpolation on  $[a, b]$  is of order 6. Moreover,

$$\|u^{(n)}(x) - H^{(n)}(x)\| \leq C\gamma_n h^{6-n}, \quad n = 0, 1, \dots, 5 \quad (14)$$

where, the values of  $\gamma_n$  are given in [12].

**Theorem 3.** [5, 3] Let  $U(x, t_j)$  be the quintic Hermite spline interpolate of  $u(x, t)$  from the space  $\mathbf{H}$  of all Hermite interpolating polynomials of order 5 defined on the interval  $0 \leq x \leq 1$  be the quintic Hermite spline interpolate of  $u(x, t)$ , such that  $P_t(x, t_j) \in C^6([a, b])$ . Then the uniform error estimate is given by

$$\|u(x, t_j) - U(x, t_j)\|_{\infty} \leq C(\Delta t + h^4). \quad (15)$$

## 6 Numerical implementation

To check the applicability of FDQHCM, numerical results have been compared to the analytic results. Stability of the proposed technique has been checked by  $L_2$ -norm and  $L_{\infty}$ -norm as follows:

$$L_2\text{-norm, } \|u\|_2 = \sqrt{\sum_{\gamma} h \sum_i w_i (u(\xi_i, t) - u^{\gamma}(\xi_i, t))^2}, \quad (16)$$

where  $w_i$ 's represents the weight function corresponding to the collocation points and

$$L_{\infty}\text{-norm, } \|u\|_{\infty} = \max_{0 \leq \xi \leq 1} |u(\xi_i, t) - u^{\gamma}(\xi_i, t)|, \quad (17)$$

where  $u(\xi_i, t)$  is the analytic solution and  $u^{\gamma}(\xi_i, t)$  is the numerical solution obtained from FDQHCM.

**Problem 1.** Consider the GBHE given in equation (1) with  $f(u) = \beta u(1 - u^\delta)(u^\delta - \bar{\gamma})$ . The exact solution to the given equation is

$$u(x, t) = \left( \frac{\bar{\gamma}}{2} + \frac{\bar{\gamma}}{2} \tanh(a_1(x - a_2 t)) \right)^{1/\delta}, \quad (18)$$

where  $a_1 = \frac{\bar{\gamma}}{4(1+\delta)}(-\mu\delta + \delta\sqrt{\mu^2 + 4\beta(1+\delta)})$  and

$$a_2 = \frac{2\mu\bar{\gamma} - (1+\delta-\bar{\gamma})(-\mu + \sqrt{\mu^2 + 4\beta(1+\delta)})}{2(1+\delta)}.$$

The initial and boundary conditions can be derived from the exact solution. From Tables 4–8, numerical values obtained from the FDQHCM have been compared to the exact values and of the results obtained from literature [13, 15, 23].

Table 4: Comparison of results for  $\mu = 1$ ,  $\beta = 1$ ,  $\delta = 1$ , and  $\gamma = 0.001$

| $x$ | $t$  | Exact       | FDQHCM       | Absolute Error          | [13]                    | [23]                     | [15]                    |
|-----|------|-------------|--------------|-------------------------|-------------------------|--------------------------|-------------------------|
| 0.1 | 0.05 | 0.000500019 | 0.0005000110 | $7.9750 \times 10^{-9}$ | $7.7006 \times 10^{-9}$ | $7.72768 \times 10^{-9}$ | $1.9372 \times 10^{-7}$ |
|     | 0.1  | 0.000500025 | 0.0005000137 | $1.1300 \times 10^{-8}$ | $1.1268 \times 10^{-8}$ | $1.12968 \times 10^{-8}$ | $3.8743 \times 10^{-7}$ |
|     | 1    | 0.000500137 | 0.0005001205 | $1.6478 \times 10^{-8}$ | $1.6863 \times 10^{-8}$ | $1.68647 \times 10^{-8}$ | $3.8750 \times 10^{-6}$ |
| 0.5 | 0.05 | 0.000500069 | 0.0005000514 | $1.7607 \times 10^{-8}$ | $1.7284 \times 10^{-8}$ | $1.73534 \times 10^{-8}$ | $1.9373 \times 10^{-7}$ |
|     | 0.1  | 0.000500075 | 0.0005000462 | $2.8837 \times 10^{-8}$ | $2.8738 \times 10^{-8}$ | $2.88305 \times 10^{-8}$ | $3.8746 \times 10^{-7}$ |
|     | 1    | 0.000500187 | 0.0005001405 | $4.6463 \times 10^{-8}$ | $4.6841 \times 10^{-8}$ | $4.68491 \times 10^{-8}$ | $3.8753 \times 10^{-6}$ |
| 0.9 | 0.05 | 0.000500119 | 0.0005001110 | $7.9760 \times 10^{-9}$ | $7.7006 \times 10^{-9}$ | $7.72823 \times 10^{-9}$ | $1.9375 \times 10^{-7}$ |
|     | 0.1  | 0.000500125 | 0.0005001137 | $1.1301 \times 10^{-8}$ | $1.1268 \times 10^{-8}$ | $1.12980 \times 10^{-8}$ | $3.8749 \times 10^{-7}$ |
|     | 1    | 0.000500237 | 0.0005002205 | $1.6481 \times 10^{-8}$ | $1.6863 \times 10^{-8}$ | $1.68669 \times 10^{-8}$ | $3.8756 \times 10^{-6}$ |

Table 5: Comparison of results for  $\mu = 1$ ,  $\beta = 1$ , and  $\bar{\gamma} = 0.001$

| $x$ | $t$  | $\delta = 2$ |              |                          | $\delta = 3$ |              |                          |
|-----|------|--------------|--------------|--------------------------|--------------|--------------|--------------------------|
|     |      | Exact        | FDQHCM       | Absolute Error           | Exact        | FDQHCM       | Absolute Error           |
| 0.1 | 0.05 | 0.0223614813 | 0.0223611210 | $3.60349 \times 10^{-7}$ | 0.0793728109 | 0.0793714961 | $1.31479 \times 10^{-6}$ |
|     | 0.1  | 0.0223617974 | 0.0223612705 | $5.26908 \times 10^{-7}$ | 0.0793740199 | 0.0793720972 | $1.92270 \times 10^{-6}$ |
|     | 1    | 0.0223674857 | 0.0223666991 | $7.86609 \times 10^{-7}$ | 0.0793957760 | 0.0793929065 | $2.86954 \times 10^{-6}$ |
| 0.5 | 0.05 | 0.0223634233 | 0.0223626133 | $8.09958 \times 10^{-7}$ | 0.0793790070 | 0.0793760514 | $2.95560 \times 10^{-6}$ |
|     | 0.1  | 0.0223637393 | 0.0223623937 | $1.34559 \times 10^{-6}$ | 0.0793802158 | 0.0793753057 | $4.91013 \times 10^{-6}$ |
|     | 1    | 0.0223694271 | 0.0223672410 | $2.18610 \times 10^{-6}$ | 0.0794019686 | 0.0793939936 | $7.97497 \times 10^{-6}$ |
| 0.9 | 0.05 | 0.0223653650 | 0.0223650047 | $3.60299 \times 10^{-7}$ | 0.0793852021 | 0.0793838875 | $1.31464 \times 10^{-6}$ |
|     | 0.1  | 0.0223656810 | 0.0223651541 | $5.26903 \times 10^{-7}$ | 0.0793864108 | 0.0793844882 | $1.92259 \times 10^{-6}$ |
|     | 1    | 0.0223713683 | 0.0223705817 | $7.86614 \times 10^{-7}$ | 0.0794081601 | 0.0794052905 | $2.86962 \times 10^{-6}$ |

It is observed that results obtained from the FDQHCM are at par with [13] and are better than [15, 23]. It is also observed that the FDQHCM gives error varying from  $10^{-9}$  to  $10^{-5}$  for varying values of  $\mu, \beta, \delta$ , and  $\bar{\gamma}$ . The results obtained from the FDQHCM appear to be more consistent than of [13]. In Table 9,  $L_2$ -norm and  $L_\infty$ -norm have been calculated for  $\mu = \beta = 1$ ,

Table 6: Comparison of results for  $\mu = 1$ ,  $\beta = 1$ , and  $\bar{\gamma} = 0.001$ 

| $x$ | $t$  | $\delta = 4$ |             |                          | $\delta = 8$ |             |                           |
|-----|------|--------------|-------------|--------------------------|--------------|-------------|---------------------------|
|     |      | Exact        | FDQHCM      | Absolute Error           | Exact        | FDQHCM      | Absolute Error            |
| 0.1 | 0.05 | 0.1495399548 | 0.149537428 | $2.52679 \times 10^{-6}$ | 0.38670979   | 0.386702959 | $6.83447 \times 10^{-6}$  |
|     | 0.1  | 0.1495423528 | 0.149538658 | $3.69481 \times 10^{-6}$ | 0.38671673   | 0.386706734 | $9.99399 \times 10^{-6}$  |
|     | 1    | 0.1495854975 | 0.149579985 | $5.51247 \times 10^{-6}$ | 0.38684140   | 0.386826506 | $1.48943 \times 10^{-5}$  |
| 0.5 | 0.05 | 0.1495506669 | 0.149544987 | $5.67995 \times 10^{-6}$ | 0.38673162   | 0.386716259 | $1.53640 \times 10^{-5}$  |
|     | 0.1  | 0.1495530645 | 0.149543629 | $9.43545 \times 10^{-6}$ | 0.38673855   | 0.386713032 | $2.55228 \times 10^{-5}$  |
|     | 1    | 0.1495961998 | 0.149580879 | $1.53208 \times 10^{-5}$ | 0.38686318   | 0.386821783 | $4.13948 \times 10^{-5}$  |
| 0.9 | 0.05 | 0.1495613768 | 0.149558851 | $2.52579 \times 10^{-6}$ | 0.38675344   | 0.386746612 | $6.83197 \times 10^{-6}$  |
|     | 0.1  | 0.1495637738 | 0.149560080 | $3.69379 \times 10^{-6}$ | 0.38676037   | 0.386750381 | $9.99200 \times 10^{-6}$  |
|     | 1    | 0.1496068999 | 0.149601387 | $5.51287 \times 10^{-6}$ | 0.38688495   | 0.386870055 | $1.489155 \times 10^{-5}$ |

Table 7: Comparison of results for  $\mu = 1$ ,  $\beta = 1$ ,  $\delta = 16$ , and  $\bar{\gamma} = 0.001$ 

| $x$ | $t$  | Exact     | FDQHCM    | Absolute Error           |
|-----|------|-----------|-----------|--------------------------|
| 0.1 | 0.05 | 0.6218689 | 0.6218574 | $1.14208 \times 10^{-5}$ |
|     | 0.1  | 0.6218811 | 0.6218644 | $1.66988 \times 10^{-5}$ |
|     | 1    | 0.6221000 | 0.6220752 | $2.48186 \times 10^{-5}$ |
| 0.5 | 0.05 | 0.6218956 | 0.6218699 | $2.56702 \times 10^{-5}$ |
|     | 0.1  | 0.6219078 | 0.6218651 | $4.26434 \times 10^{-5}$ |
|     | 1    | 0.6221265 | 0.6220576 | $6.89814 \times 10^{-5}$ |
| 0.9 | 0.05 | 0.6219223 | 0.6219108 | $1.14113 \times 10^{-5}$ |
|     | 0.1  | 0.6219344 | 0.6219178 | $1.66897 \times 10^{-5}$ |
|     | 1    | 0.6221531 | 0.6221283 | $2.48109 \times 10^{-5}$ |

Table 8: Comparison of results for  $\mu = 0$ ,  $\beta = 1$ ,  $\delta = 1$ , and  $\bar{\gamma} = 0.001$ 

| $x$ | $t$  | Exact       | FDQHCM        | Absolute Error          | [13]                    | [23]                    | [15]                    |
|-----|------|-------------|---------------|-------------------------|-------------------------|-------------------------|-------------------------|
| 0.1 | 0.05 | 0.000500030 | 0.00050001989 | $1.0286 \times 10^{-8}$ | $1.0269 \times 10^{-8}$ | $1.0303 \times 10^{-8}$ | $1.8747 \times 10^{-7}$ |
|     | 0.1  | 0.000500043 | 0.00050002762 | $1.5044 \times 10^{-8}$ | $1.5027 \times 10^{-8}$ | $1.5063 \times 10^{-8}$ | $3.7493 \times 10^{-7}$ |
|     | 1    | 0.000500268 | 0.00050024509 | $2.2466 \times 10^{-8}$ | $2.2488 \times 10^{-8}$ | $2.2488 \times 10^{-8}$ | $3.7500 \times 10^{-6}$ |
| 0.5 | 0.05 | 0.000500101 | 0.00050007775 | $2.3132 \times 10^{-8}$ | $2.3049 \times 10^{-8}$ | $2.3138 \times 10^{-8}$ | $1.8749 \times 10^{-7}$ |
|     | 0.1  | 0.000500113 | 0.00050007495 | $3.8429 \times 10^{-8}$ | $3.8324 \times 10^{-8}$ | $3.8441 \times 10^{-8}$ | $3.7498 \times 10^{-7}$ |
|     | 1    | 0.000500338 | 0.00050027582 | $6.2443 \times 10^{-8}$ | $6.2465 \times 10^{-8}$ | $6.2465 \times 10^{-8}$ | $3.7504 \times 10^{-6}$ |
| 0.9 | 0.05 | 0.000500172 | 0.00050016131 | $1.0283 \times 10^{-8}$ | $1.0269 \times 10^{-8}$ | $1.0303 \times 10^{-8}$ | $1.8751 \times 10^{-7}$ |
|     | 0.1  | 0.000500184 | 0.00050016904 | $1.5047 \times 10^{-8}$ | $1.5027 \times 10^{-8}$ | $1.5063 \times 10^{-8}$ | $3.7502 \times 10^{-7}$ |
|     | 1    | 0.000500409 | 0.00050038651 | $2.2464 \times 10^{-8}$ | $2.2488 \times 10^{-8}$ | $2.2488 \times 10^{-8}$ | $3.7509 \times 10^{-6}$ |

$\bar{\gamma} = 0.001$ , and for  $\delta = 1, 2, 3$ , respectively. Both the norms are lying within the range of  $10^{-8}$  to  $10^{-6}$ .

Table 9:  $L_2$ -norm and  $L_\infty$ -norm for  $\mu = 1$ ,  $\beta = 1$ , and  $\bar{\gamma} = 0.001$ 

| $t$  | $L_2$ -norm               |                           |                           | $L_\infty$ -norm          |                           |                           |
|------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
|      | $\delta = 1$              | $\delta = 2$              | $\delta = 3$              | $\delta = 1$              | $\delta = 2$              | $\delta = 3$              |
| 0.1  | $6.328299 \times 10^{-9}$ | $2.954021 \times 10^{-7}$ | $1.077846 \times 10^{-6}$ | $2.880650 \times 10^{-8}$ | $1.344790 \times 10^{-6}$ | $4.906733 \times 10^{-6}$ |
| 0.25 | $9.224090 \times 10^{-9}$ | $4.305708 \times 10^{-7}$ | $1.571002 \times 10^{-6}$ | $4.271075 \times 10^{-8}$ | $1.993759 \times 10^{-6}$ | $7.274619 \times 10^{-6}$ |
| 0.5  | $1.000496 \times 10^{-8}$ | $4.669971 \times 10^{-7}$ | $1.703806 \times 10^{-6}$ | $4.645850 \times 10^{-8}$ | $2.168583 \times 10^{-6}$ | $7.911991 \times 10^{-6}$ |

From Figures 4–7, three-dimensional behavior of  $u(x, t)$  has been presented for  $\mu = \beta = 1$ ,  $\bar{\gamma} = 0.001$ , and  $\delta = 1, 2, 3, 4$ , respectively, which shows that the values are lying between 0 and 1 for time ranging from 0 to 0.01.

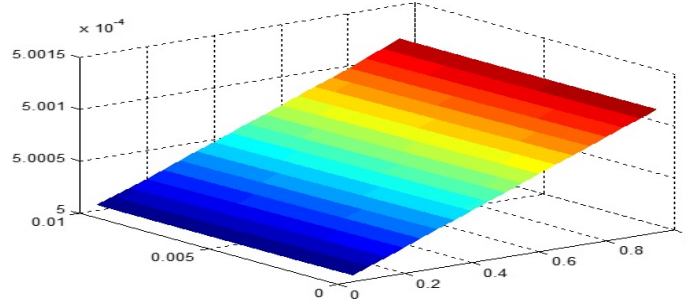


Figure 4: Three-dimensional behavior of  $u(x, t)$  for  $\mu = 1$ ,  $\beta = 1$ ,  $\gamma = 0.001$ , and  $\delta = 1$ .

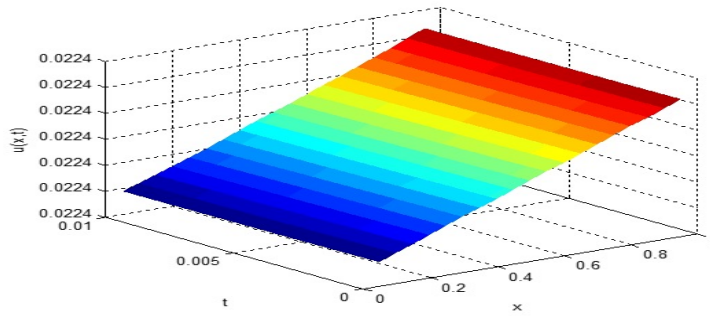
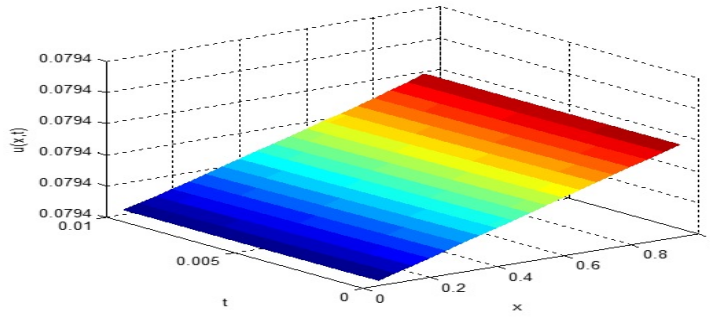
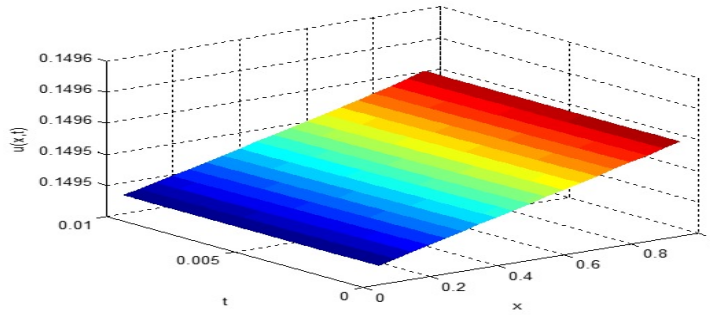


Figure 5: Three-dimensional behavior of  $u(x, t)$  for  $\mu = 1$ ,  $\beta = 1$ ,  $\gamma = 0.001$ , and  $\delta = 2$ .

**Problem 2.** Consider the GBFE given in equation (1) for  $\varepsilon = 1$  with  $f(u) = \beta u(1 - u^\delta)$ . The exact solution for the GBFE is

$$u(x, t) = \left( \frac{1}{2} + \frac{1}{2} \tanh(a_3(x - a_4 t)) \right)^{1/\delta}, \quad (19)$$

where  $a_3 = \frac{-\mu\delta}{2(1+\delta)}$  and  $a_4 = \frac{\mu^2 + \beta(1+\delta)^2}{\mu(1+\delta)}$ . The initial condition and boundary conditions can be obtained from the exact solution. From Tables 10–17, numerical values obtained from the FDQHCM have been compared to the exact values and the results obtained from [13, 23]. It is observed that results obtained from the FDQHCM are at par with the results obtained from [13] and are better than [22]. However, in Tables 10 and 17, results by [13] are slightly better than the FDQHCM, but this is overcome by the simplicity of

Figure 6: Three-dimensional behavior of  $u(x, t)$  for  $\mu = 1$ ,  $\beta = 1$ ,  $\gamma = 0.001$ , and  $\delta = 3$ .Figure 7: Three-dimensional behavior of  $u(x, t)$  for  $\mu = 1$ ,  $\beta = 1$ ,  $\gamma = 0.001$ , and  $\delta = 4$ .

the FDQHCM as compared to the technique of former. The absolute error varies from  $10^{-11}$  to  $10^{-5}$  for varying values of  $\mu, \beta$ , and  $\delta$ . In Table 18,  $L_2$ -norm and  $L_\infty$ -norm have been calculated for  $\mu = \beta = 0.001$  and  $\delta = 1, 2, 3$ , respectively.

Table 10: Comparison of results for  $\mu = 0.001$ ,  $\beta = 0.001$ , and  $\delta = 1$ 

| $x$ | $t$   | Exact    | FDQHCM   | Absolute Error          | [13]                     | [22]                  | [15]                     |
|-----|-------|----------|----------|-------------------------|--------------------------|-----------------------|--------------------------|
| 0.1 | 0.001 | 0.499988 | 0.499988 | $9.999 \times 10^{-10}$ | $5.8147 \times 10^{-11}$ | $1.01 \times 10^{-7}$ | $9.68763 \times 10^{-6}$ |
|     | 0.005 | 0.499989 | 0.499989 | $6.999 \times 10^{-9}$  | $2.6058 \times 10^{-10}$ | $4.38 \times 10^{-7}$ | $1.93753 \times 10^{-6}$ |
|     | 0.01  | 0.499990 | 0.499990 | $1.100 \times 10^{-8}$  | $4.4599 \times 10^{-10}$ | $7.53 \times 10^{-7}$ | $1.93752 \times 10^{-5}$ |
| 0.5 | 0.001 | 0.499938 | 0.499938 | $4.460 \times 10^{-13}$ | $5.6241 \times 10^{-11}$ | $1.04 \times 10^{-7}$ | $9.68691 \times 10^{-6}$ |
|     | 0.005 | 0.499939 | 0.499939 | $9.310 \times 10^{-13}$ | $3.0621 \times 10^{-10}$ | $5.21 \times 10^{-7}$ | $1.93738 \times 10^{-6}$ |
|     | 0.01  | 0.499940 | 0.499940 | $9.985 \times 10^{-10}$ | $6.1867 \times 10^{-10}$ | $1.04 \times 10^{-6}$ | $1.93738 \times 10^{-5}$ |
| 0.9 | 0.001 | 0.499888 | 0.499888 | $9.980 \times 10^{-10}$ | $5.8135 \times 10^{-11}$ | $1.01 \times 10^{-7}$ | $9.68619 \times 10^{-6}$ |
|     | 0.005 | 0.499889 | 0.499889 | $6.998 \times 10^{-9}$  | $2.6053 \times 10^{-10}$ | $4.38 \times 10^{-7}$ | $1.93724 \times 10^{-6}$ |
|     | 0.01  | 0.499890 | 0.499890 | $1.100 \times 10^{-8}$  | $4.4591 \times 10^{-10}$ | $7.53 \times 10^{-7}$ | $1.93724 \times 10^{-5}$ |

Table 11: Comparison of results for  $\mu = 0.001$ ,  $\beta = 0.001$ 

| $x$ | $t$   | $\delta = 2$ |            |                           | $\delta = 3$ |            |                          |
|-----|-------|--------------|------------|---------------------------|--------------|------------|--------------------------|
|     |       | Exact        | FDQHCM     | Absolute Error            | Exact        | FDQHCM     | Absolute Error           |
| 0.1 | 0.001 | 0.70709535   | 0.70709535 | $4.65367 \times 10^{-10}$ | 0.79369100   | 0.79369100 | $5.3640 \times 10^{-10}$ |
|     | 0.005 | 0.70709676   | 0.70709677 | $1.02299 \times 10^{-8}$  | 0.79369259   | 0.79369260 | $1.1100 \times 10^{-8}$  |
|     | 0.01  | 0.70709853   | 0.70709855 | $1.54395 \times 10^{-8}$  | 0.79369457   | 0.79369459 | $1.7813 \times 10^{-8}$  |
| 0.5 | 0.001 | 0.70704821   | 0.70704821 | $2.5057 \times 10^{-10}$  | 0.79365131   | 0.79365131 | $4.9914 \times 10^{-10}$ |
|     | 0.005 | 0.70704962   | 0.70704962 | $7.9174 \times 10^{-11}$  | 0.79365290   | 0.79365290 | $9.5943 \times 10^{-11}$ |
|     | 0.01  | 0.70705139   | 0.70705139 | $1.2627 \times 10^{-11}$  | 0.79365489   | 0.79365489 | $4.1912 \times 10^{-10}$ |
| 0.9 | 0.001 | 0.70700106   | 0.70700106 | $1.17716 \times 10^{-9}$  | 0.79361162   | 0.79361162 | $1.4294 \times 10^{-9}$  |
|     | 0.005 | 0.70700248   | 0.70700249 | $1.07532 \times 10^{-8}$  | 0.79361321   | 0.79361322 | $1.1675 \times 10^{-8}$  |
|     | 0.01  | 0.70700424   | 0.70700426 | $1.57272 \times 10^{-8}$  | 0.79361519   | 0.79361521 | $1.6992 \times 10^{-8}$  |

Table 12: Comparison of results for  $\mu = 0.001$ ,  $\beta = 0.001$ , and  $\delta = 4$ 

| $x$ | $t$   | Exact      | FDQHCM     | Absolute Error           | [22]                  |
|-----|-------|------------|------------|--------------------------|-----------------------|
| 0.1 | 0.001 | 0.84088843 | 0.84088843 | $1.37598 \times 10^{-9}$ | $1.75 \times 10^{-8}$ |
|     | 0.005 | 0.84089011 | 0.84089012 | $1.25399 \times 10^{-8}$ | $7.37 \times 10^{-7}$ |
|     | 0.01  | 0.84089221 | 0.84089223 | $1.82591 \times 10^{-8}$ | $1.27 \times 10^{-6}$ |
| 0.5 | 0.001 | 0.84085479 | 0.84085479 | $2.0918 \times 10^{-10}$ | $1.75 \times 10^{-8}$ |
|     | 0.005 | 0.84085647 | 0.84085647 | $1.7133 \times 10^{-10}$ | $8.77 \times 10^{-7}$ |
|     | 0.01  | 0.84085857 | 0.84085857 | $6.3821 \times 10^{-10}$ | $1.75 \times 10^{-6}$ |
| 0.9 | 0.001 | 0.84082114 | 0.84082115 | $1.0779 \times 10^{-9}$  | $1.75 \times 10^{-8}$ |
|     | 0.005 | 0.84082283 | 0.84082284 | $1.1838 \times 10^{-8}$  | $7.38 \times 10^{-7}$ |
|     | 0.01  | 0.84082493 | 0.84082495 | $1.8053 \times 10^{-8}$  | $1.27 \times 10^{-6}$ |

Table 13: Comparison of results for  $\mu = 0.001$  and  $\beta = 0.001$ 

| $x$ | $t$   | $\delta = 8$ |            |                            | $\delta = 16$ |           |                         |
|-----|-------|--------------|------------|----------------------------|---------------|-----------|-------------------------|
|     |       | Exact        | FDQHCM     | Absolute Error             | Exact         | FDQHCM    | Absolute Error          |
| 0.1 | 0.001 | 0.91699941   | 0.91699941 | $8.421940 \times 10^{-10}$ | 0.9576009     | 0.9576009 | $1.023 \times 10^{-9}$  |
|     | 0.005 | 0.91700124   | 0.91697903 | $2.222702 \times 10^{-5}$  | 0.9576028     | 0.9576029 | $4.177 \times 10^{-8}$  |
|     | 0.01  | 0.91700353   | 0.91695864 | $4.491036 \times 10^{-5}$  | 0.9576052     | 0.9576053 | $4.778 \times 10^{-8}$  |
| 0.5 | 0.001 | 0.91697903   | 0.91700124 | $2.221422 \times 10^{-5}$  | 0.9575897     | 0.9575897 | $2.435 \times 10^{-8}$  |
|     | 0.005 | 0.91698086   | 0.91698086 | $3.297860 \times 10^{-10}$ | 0.9575916     | 0.9575916 | $2.357 \times 10^{-10}$ |
|     | 0.01  | 0.91698315   | 0.91696048 | $2.267573 \times 10^{-5}$  | 0.9575940     | 0.9575940 | $1.435 \times 10^{-8}$  |
| 0.9 | 0.001 | 0.91695864   | 0.91700353 | $4.488876 \times 10^{-5}$  | 0.9575784     | 0.9575784 | $6.336 \times 10^{-9}$  |
|     | 0.005 | 0.91696048   | 0.91698315 | $2.266323 \times 10^{-5}$  | 0.9578803     | 0.9575803 | $2.227 \times 10^{-8}$  |
|     | 0.01  | 0.91696277   | 0.91696277 | $1.947571 \times 10^{-8}$  | 0.9575827     | 0.9575827 | $2.092 \times 10^{-8}$  |

Table 14: Comparison of absolute errors for  $\mu = 0.1$  and  $\beta = -0.0025$ 

| $x$ | $t$ | $\delta = 2$            |                          |                        | $\delta = 4$            |                          |                        |
|-----|-----|-------------------------|--------------------------|------------------------|-------------------------|--------------------------|------------------------|
|     |     | FDQHCM                  | [13] &                   | [22]                   | FDQHCM                  | [13]                     | [22]                   |
| 0.1 | 0.1 | $1.9313 \times 10^{-5}$ | $1.76638 \times 10^{-5}$ | $1.121 \times 10^{-5}$ | $5.9667 \times 10^{-6}$ | $1.26230 \times 10^{-5}$ | $1.343 \times 10^{-5}$ |
|     | 0.3 | $7.0917 \times 10^{-5}$ | $2.51379 \times 10^{-5}$ | $1.600 \times 10^{-5}$ | $2.1779 \times 10^{-5}$ | $1.79797 \times 10^{-5}$ | $1.919 \times 10^{-5}$ |
|     | 0.5 | $1.2556 \times 10^{-5}$ | $2.61751 \times 10^{-5}$ | $1.667 \times 10^{-5}$ | $3.8540 \times 10^{-5}$ | $1.87212 \times 10^{-5}$ | $2.001 \times 10^{-5}$ |
| 0.5 | 0.1 | $6.4389 \times 10^{-6}$ | $4.49179 \times 10^{-5}$ | $2.904 \times 10^{-5}$ | $2.0158 \times 10^{-6}$ | $3.21358 \times 10^{-5}$ | $3.489 \times 10^{-5}$ |
|     | 0.3 | $5.0445 \times 10^{-5}$ | $6.91014 \times 10^{-5}$ | $4.468 \times 10^{-5}$ | $1.5604 \times 10^{-5}$ | $4.94694 \times 10^{-5}$ | $5.373 \times 10^{-5}$ |
|     | 0.5 | $1.0433 \times 10^{-4}$ | $7.24595 \times 10^{-5}$ | $4.687 \times 10^{-5}$ | $3.2214 \times 10^{-5}$ | $5.18702 \times 10^{-5}$ | $5.641 \times 10^{-5}$ |
| 0.9 | 0.1 | $1.9529 \times 10^{-5}$ | $1.75391 \times 10^{-5}$ | $1.154 \times 10^{-5}$ | $6.1067 \times 10^{-6}$ | $1.25646 \times 10^{-5}$ | $1.393 \times 10^{-5}$ |
|     | 0.3 | $7.1767 \times 10^{-5}$ | $2.50111 \times 10^{-5}$ | $1.643 \times 10^{-5}$ | $2.2305 \times 10^{-5}$ | $1.7920 \times 10^{-5}$  | $1.981 \times 10^{-5}$ |
|     | 0.5 | $1.2709 \times 10^{-4}$ | $2.60482 \times 10^{-5}$ | $1.711 \times 10^{-5}$ | $3.9456 \times 10^{-5}$ | $1.86610 \times 10^{-5}$ | $2.065 \times 10^{-5}$ |

Table 15: Comparison of absolute errors for  $\mu = 0.1$ ,  $\beta = -0.0025$ , and  $\delta = 8$ 

| $x$ | $t$ | FDQHCM                  | [13]                     | [22]                   |
|-----|-----|-------------------------|--------------------------|------------------------|
| 0.1 | 0.1 | $4.8295 \times 10^{-7}$ | $7.65875 \times 10^{-6}$ | $1.471 \times 10^{-5}$ |
|     | 0.3 | $1.5645 \times 10^{-6}$ | $1.09129 \times 10^{-5}$ | $2.107 \times 10^{-5}$ |
|     | 0.5 | $2.6947 \times 10^{-6}$ | $1.13630 \times 10^{-5}$ | $2.203 \times 10^{-5}$ |
| 0.5 | 0.1 | $1.8448 \times 10^{-7}$ | $1.95143 \times 10^{-5}$ | $3.832 \times 10^{-5}$ |
|     | 0.3 | $1.1484 \times 10^{-6}$ | $3.00445 \times 10^{-5}$ | $5.911 \times 10^{-5}$ |
|     | 0.5 | $2.2738 \times 10^{-6}$ | $3.15019 \times 10^{-5}$ | $6.218 \times 10^{-5}$ |
| 0.9 | 0.1 | $4.9302 \times 10^{-7}$ | $7.63553 \times 10^{-5}$ | $1.533 \times 10^{-5}$ |
|     | 0.3 | $1.6078 \times 10^{-6}$ | $1.08887 \times 10^{-5}$ | $2.183 \times 10^{-5}$ |
|     | 0.5 | $2.7772 \times 10^{-6}$ | $1.13383 \times 10^{-5}$ | $2.280 \times 10^{-5}$ |

Table 16: Comparison of absolute errors for  $\mu = 1$  and  $\beta = 0$ 

| $x$ | $t$    | $\delta = 3$             |                          | $\delta = 8$              |                          |
|-----|--------|--------------------------|--------------------------|---------------------------|--------------------------|
|     |        | FDQHCM                   | [13]                     | FDQHCM                    | [13]                     |
| 0.1 | 0.0001 | $2.3559 \times 10^{-10}$ | $5.7870 \times 10^{-11}$ | $8.59409 \times 10^{-11}$ | $2.7729 \times 10^{-11}$ |
|     | 0.0005 | $2.6056 \times 10^{-8}$  | $2.0870 \times 10^{-5}$  | $1.35615 \times 10^{-8}$  | $1.0782 \times 10^{-5}$  |
|     | 0.001  | $6.7400 \times 10^{-7}$  | $4.5521 \times 10^{-5}$  | $3.86425 \times 10^{-7}$  | $2.3520 \times 10^{-5}$  |
| 0.5 | 0.0001 | $4.6311 \times 10^{-10}$ | $5.1671 \times 10^{-11}$ | $9.37 \times 10^{-11}$    | $2.0409 \times 10^{-11}$ |
|     | 0.0005 | $2.6853 \times 10^{-10}$ | $1.7895 \times 10^{-5}$  | $4.82443 \times 10^{-10}$ | $9.4066 \times 10^{-6}$  |
|     | 0.001  | $4.8165 \times 10^{-10}$ | $4.0262 \times 10^{-5}$  | $2.49371 \times 10^{-10}$ | $2.1163 \times 10^{-5}$  |
| 0.9 | 0.0001 | $4.5141 \times 10^{-10}$ | $5.3143 \times 10^{-11}$ | $1.40942 \times 10^{-10}$ | $1.8090 \times 10^{-11}$ |
|     | 0.0005 | $2.9347 \times 10^{-8}$  | $1.6546 \times 10^{-5}$  | $1.7873 \times 10^{-8}$   | $8.7305 \times 10^{-6}$  |
|     | 0.001  | $7.5936 \times 10^{-7}$  | $3.6150 \times 10^{-5}$  | $4.92825 \times 10^{-7}$  | $1.9075 \times 10^{-5}$  |

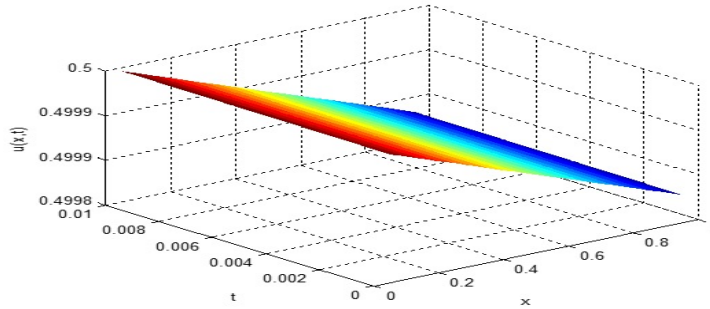
Table 17: Comparison of absolute errors for  $\mu = 1$  and  $\beta = 1$ 

| $x$ | $t$    | $\delta = 1$             |                           | $\delta = 2$             |                       |
|-----|--------|--------------------------|---------------------------|--------------------------|-----------------------|
|     |        | FDQHCM                   | FDQHCM                    | [13]                     | [22]                  |
| 0.1 | 0.0001 | $1.2969 \times 10^{-6}$  | $3.03026 \times 10^{-10}$ | $4.4788 \times 10^{-10}$ | $1.55 \times 10^{-5}$ |
|     | 0.0005 | $5.9286 \times 10^{-5}$  | $9.30236 \times 10^{-8}$  | $2.4700 \times 10^{-5}$  | $7.62 \times 10^{-5}$ |
|     | 0.001  | $1.9592 \times 10^{-4}$  | $1.62366 \times 10^{-6}$  | $5.3872 \times 10^{-5}$  | $1.50 \times 10^{-4}$ |
| 0.5 | 0.0001 | $5.9153 \times 10^{-10}$ | $1.99555 \times 10^{-10}$ | $7.1278 \times 10^{-11}$ | $1.83 \times 10^{-5}$ |
|     | 0.0005 | $1.5519 \times 10^{-10}$ | $2.98812 \times 10^{-10}$ | $2.0953 \times 10^{-5}$  | $9.14 \times 10^{-5}$ |
|     | 0.001  | $9.1902 \times 10^{-8}$  | $2.91233 \times 10^{-10}$ | $4.7151 \times 10^{-5}$  | $1.83 \times 10^{-4}$ |
| 0.9 | 0.0001 | $1.1642 \times 10^{-6}$  | $2.34912 \times 10^{-10}$ | $5.5348 \times 10^{-10}$ | $2.07 \times 10^{-5}$ |
|     | 0.0005 | $5.3457 \times 10^{-5}$  | $9.77826 \times 10^{-8}$  | $1.9314 \times 10^{-5}$  | $1.02 \times 10^{-4}$ |
|     | 0.001  | $1.7683 \times 10^{-4}$  | $1.70078 \times 10^{-6}$  | $4.2211 \times 10^{-5}$  | $2.00 \times 10^{-4}$ |

Table 18:  $L_2$ -norm and  $L_\infty$ -norm for  $\mu = 0.001$  and  $\beta = 0.001$ 

| $t$  | $L_2$ -norm               |                           |                           | $L_\infty$ -norm          |                           |                           |
|------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
|      | $\delta = 1$              | $\delta = 2$              | $\delta = 3$              | $\delta = 1$              | $\delta = 2$              | $\delta = 3$              |
| 0.01 | $6.996123 \times 10^{-9}$ | $8.760110 \times 10^{-9}$ | $9.773331 \times 10^{-9}$ | $4.937500 \times 10^{-8}$ | $6.633549 \times 10^{-8}$ | $7.056646 \times 10^{-8}$ |
| 0.05 | $1.243733 \times 10^{-8}$ | $1.776084 \times 10^{-8}$ | $2.003356 \times 10^{-8}$ | $6.687500 \times 10^{-8}$ | $1.007198 \times 10^{-7}$ | $1.096476 \times 10^{-7}$ |
| 0.1  | $1.680553 \times 10^{-8}$ | $2.385368 \times 10^{-8}$ | $2.735586 \times 10^{-8}$ | $7.375000 \times 10^{-8}$ | $1.090978 \times 10^{-7}$ | $1.229183 \times 10^{-7}$ |

Both the norms are lying within the range of  $10^{-8}$  to  $10^{-7}$ . From Figures 8–11, three-dimensional behavior of  $u(x, t)$  has been presented for  $\mu = \beta = 0.001$  and  $\delta = 1, 2, 3, 4$ , which shows that the values are lying between 0 and 1 for time ranging from 0 to 0.01.

Figure 8: Three-dimensional behavior of  $u(x, t)$  for  $\mu = 0.001$ ,  $\beta = 0.001$ ,  $\gamma = 0.001$ , and  $\delta = 1$ .



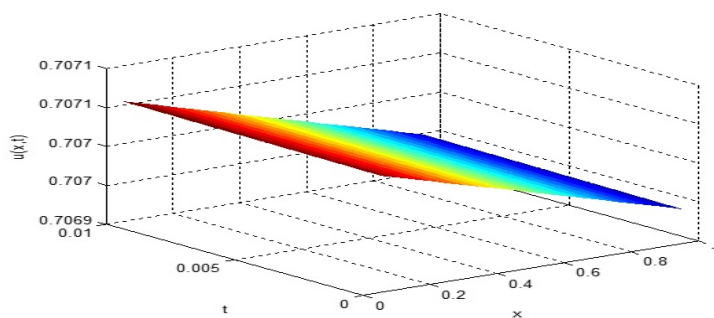


Figure 9: Three-dimensional behavior of  $u(x,t)$  for  $\mu = 0.001$ ,  $\beta = 0.001$ ,  $\gamma = 0.001$ , and  $\delta = 2$ .

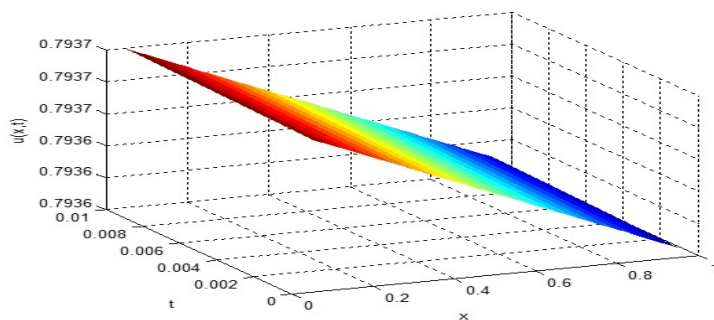


Figure 10: Three-dimensional behavior of  $u(x,t)$  for  $\mu = 0.001$ ,  $\beta = 0.001$ ,  $\gamma = 0.001$ , and  $\delta = 3$ .

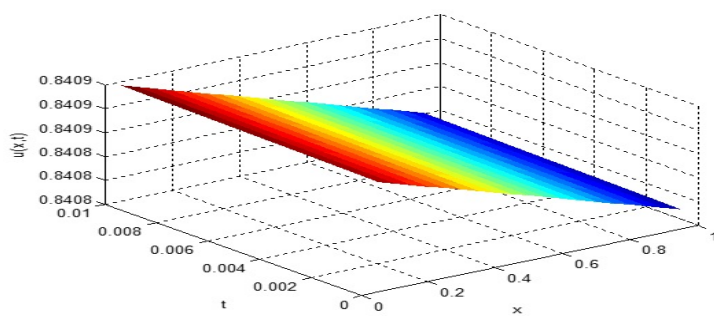


Figure 11: Three-dimensional behavior of  $u(x,t)$  for  $\mu = 0.001$ ,  $\beta = 0.001$ ,  $\gamma = 0.001$ , and  $\delta = 4$ .

## 7 Conclusions

The technique of the FDQHCM has been implemented successfully on the GBHE and GBFE. The technique is a combination of the weighted residual method and the finite difference method, which gives stability to the numerical results and is easily adaptable to computer codes. Numerical results have been calculated for a vast range of parameters, even for those parameters not been calculated earlier. The technique is found to be more consistent than [13, 15, 22, 23]. It was also shown that the FDQHCM could be applied to nonlinear partial differential equations of higher order too.

## References

- [1] Alharbi, A. and Fahmy, E.S. *ADM-Pade solutions for generalized Burgers and Burgers-Huxley systems with two coupled equations*, J. Comput. Appl. Math. 233 (2010) 2071–2080.
- [2] Arora, S., Dhaliwal, S.S. and Kukreja, V.K. *Application of orthogonal collocation on finite elements for solving non-linear boundary value problems*, Appl. Math. Comput. 180 (2006) 516–523.
- [3] Arora, S., Jain, R. and Kukreja, V.K. *Solution of Benjamin-Bona-Mahony-Burgers equation using collocation method with quintic Hermite splines*, Appl. Numer. Math. 154 (2020) 1–16.
- [4] Arora, S. and Kaur, I. *An efficient scheme for numerical solution of burgers' equation using quintic Hermite interpolating polynomials*, Arab. J. Math. 5 (2016) 23–34.
- [5] Arora, S. and Kaur, I. *Applications of quintic Hermite collocation with time discretization to singularly perturbed problems*, Appl. Math. Comput. 316 (2018) 409–421.
- [6] Arora, S., Kaur, I., Kumar, H. and Kukreja, V.K. *A robust technique of cubic Hermite collocation for solution of two phase nonlinear model*, J. King Saud Univ. Eng. Sci. 29 (2017) 159–165.
- [7] Asogwa, K., Mebarek-Oudina, F. and Animasaun, I., *Comparative investigation of water-based Al<sub>2</sub>O<sub>3</sub> nanoparticles through water-based CuO nanoparticles over an exponentially accelerated radiative Riga plate surface via heat transport*. Arab. J. Sci. Eng. (2022) 1–18.
- [8] Celik, I. *Haar wavelet method for solving generalized Burgers-Huxley equation*, Arab J. Math. Sci. 18 (2012) 25–37.

- [9] Chabani, I., Mebarek Oudina, F. and Ismail, A.I. *MHD flow of a Hybrid nano-fluid in a triangular enclosure with zigzags and an elliptic obstacle*. Micromachines, 13 (2022) 224.
- [10] Djebali, R., Mebarek-Oudina, F. and Choudhari, R. *Similarity solution analysis of dynamic and thermal boundary layers: Further formulation along a vertical flat plate*. Phys. Scr. 96 (2021) 085206.
- [11] Farhan, M., Omar, Z., Mebarek-Oudina, F., Raza, J., Shah, Z., Choudhari, R.V. and Makinde, O.D. *Implementation of one step one Hybrid block method on nonlinear equation of the circular sector oscillator*. Comput. Math. Model. 31 (2020) 116–132.
- [12] Hall, C. *On error bounds for spline interpolation*, J. Approx. Theory. 1 (1968) 209–218.
- [13] Hammad, D. A. and El-Azab, M. S. *2N order compact finite difference scheme with collocation method for solving the generalized Burger's-Huxley and Burger's-Fisher equations*, Appl. Math. Comput. 258 (2015) 296–311.
- [14] Inan, B. and Bahadir, A. R. *Numerical solution of the generalized Burgers Huxley equation by implicit exponential finite difference method*, J. Appl. Math. Inform. 11 (2015), 57–67.
- [15] Ismail, H.N.A., Raslan, K. and Rabboh, A.A.A. *Adomain-decomposition method for Burger's Huxley and Burger's Fisher equations*, Appl. Numer. Math. 159 (2004) 291–301.
- [16] Javidi, M. *A numerical solution of the generalized Burgers-Huxley equation by spectral collocation method*, Appl. Math. Comput. 178 (2006) 338–344.
- [17] Kushner, A.G. and Matviychuk, R.I. *Finite dimensional dynamics and exact solutions of Burgers-Huxley-equation*, Twelfth International Conference "Management of large scale system development", Moscow, Russia, (2019) 1–3.
- [18] Marzougui, S., Mebarek-Oudina, F., Mchirgui, A. and Magherbi, M. *Entropy generation and heat transport of Cu-water nanoliquid in porous lid-driven cavity through magnetic field*. Int. J. Numer. Methods Heat Fluid Flow, (2021).
- [19] Miller, J.J.H., O'Riordan, R.E. and Shishkin, G. I. *Fitted numerical methods for singular perturbation problems*, World Scientific, Singapore, 1996.
- [20] Rathish Kumar, B. V., Vivek, S., Murthy, S.V.S.S.N.V.G.K. and Nigam, M. *A numerical study of singularly perturbed generalized Burgers-Huxley*

- equation using three-step Taylor–Galerkin method*, Comput. Math. Appl. 62 (2011) 776–786.
- [21] Saha Ray, S. and Gupta, A.K. *On the solution of Burgers–Huxley and Huxley equation using wavelet collocation method*, Comput. Model. Eng. Sci. 91 (2013) 409–424.
  - [22] Sari, M., Gurarslan, G. and Dag, I. *A compact finite difference method for the solution of the generalized Burgers–Fisher equation*, Numer. Methods Partial Differ. Equ. 26 (2009) 125–134.
  - [23] Sari, M., Gurarslan, G. and Zeytinoglu, A. *High-order finite difference schemes for numerical solutions of the generalized Burger–Huxley equation*, Numer. Methods Partial Differ. Equ. 27 (2011) 1313–1326.
  - [24] Tersenov, A.S. *On the generalized Burgers equation*, Nonlinear Differ. Equ. Appl. 17 (2010) 437–452.
  - [25] Warke, A.S., Ramesh, K., Mebarek-Oudina, F. and Abidi, A. *Numerical investigation of nonlinear radiation with Magnetomicroscopic Stagnation point flow past a heated stretching sheet*. J. Therm. Anal. Calorim. 135 (2021) 533–549.

#### How to cite this article

Kaur, I., Arora, S. and Bala, I., An improvised technique of quintic hermite splines to discretize generalized Burgers–Huxley type equations. *Iran. j. numer. anal. optim.*, 2023; 13(1): 59–79. <https://doi.org/10.22067/ijnao.2022.75871.1120>.



## Generalization of equitable efficiency in multiobjective optimization problems by the direct sum of matrices

F. Ahmadi, A. R. Salajegheh and D. Foroutannia\*

### Abstract

We suggest an a priori method by introducing the concept of  $A_P$ -equitable efficiency. The preferences matrix  $A_P$ , which is based on the partition  $P$  of the index set of the objective functions, is given by the decision-maker. We state the certain conditions on the matrix  $A_P$  that guarantee the preference relation  $\preceq_{eA_P}$  to satisfy the strict monotonicity and strict  $P$ -transfer principle axioms.

A problem most frequently encountered in multiobjective optimization is that the set of Pareto optimal solutions provided by the optimization process is a large set. Hence, the decision-making based on selecting a unique preferred solution becomes difficult. Considering models with  $A_P^r$ -equitable efficiency and  $A_P^\infty$ -equitable efficiency can help the decision-maker for overcoming this difficulty, by shrinking the solution set.

**AMS subject classifications (2020):** Primary 45D05; Secondary 42C10, 65G99.

**Keywords:** Nondominated, equitable efficiency,  $A_P$ -equitable efficiency, Multiobjective programming.

---

\*Corresponding author

Received 8 April 2021; revised 24 April 2022; accepted 7 May 2022

F. Ahmadi

Department of Mathematics, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran.  
e-mail: fatemeh.Ahmadi@stu.vru.ac.ir

A. R. Salajegheh

Department of Mathematics, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran.  
e-mail: Salajegheh@stu.vru.ac.ir

D. Foroutannia

Department of Mathematics, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran.  
e-mail: foroutan@vru.ac.ir

## 1 Introduction

A problem that sometimes occurs in classical multiobjective optimization is that the set of efficient solutions is a large set. By using a priori methods, we can generate finite sets of Pareto optimal solutions, which can help the decision-maker in the task of selecting the most appropriate solution. A priori methods are based on the preferences matrix, which evaluates how to combine the objective functions by the decision-maker to introduce a preference function. Note that in a priori methods, the preferences are expressed by the decision-maker before the solution process (e.g., setting goals or weights to the objective functions). The criticism about a priori methods is that it is very difficult for the decision-maker to beforehand define and accurately quantify his preferences; see [4].

The concept of equitable efficiency is a specific refinement of the Pareto efficiency. While the Pareto efficiency assumes that the criteria are uncomparable (not measured on a common scale), the equitability is based on the assumption that the criteria are comparable, impartial (anonymous), and that the Pigou–Dalton principle of transfer holds. The impartiality axiom makes the distribution of outcomes among the criteria more important than the assignment of outcomes to specific criteria. Therefore models are the equitable allocation of resources.

The equitable preference was first known as the generalized Lorenz dominance [8, 10]. Kostreva and Ogryczak [6] and Kostreva, Ogryczak, and Wierzbicki [7] are the first ones who introduced the concept of equitability into multiobjective programming. They analyzed solution properties and approaches to generating equitably efficient solutions. A complete preference structure of equitability is derived by Bataar and Wiecek [1]. Furthermore, the concept of equitability in multiobjective programming is generalized within a framework of convex cones by Mut and Wiecek [11]. They introduced the concept of  $A$ -equitable efficiency for solving the multiobjective optimization problems, where  $A$  is an arbitrary matrix with nonnegative entries, and they also showed that the preference relation  $\preceq_{eA}$  satisfies the axioms of reflexivity, transitivity, and impartiality while the weak principle of transfer requires a condition on the matrix  $A$ . Because the preference relation  $\preceq_{eA}$  does not satisfy the strict monotonicity and strict principle of transfer axioms in general, the set of  $A$ -equitably efficient solutions does not contain within the set of equitably efficient solutions and the set of Pareto optimal solutions for the same problem. Foroutannia and Merati [3] stated new conditions on the matrix  $A$  that guarantee to hold these axioms by the preference relation  $\preceq_{eA}$ .

Let the partition  $P$  of the index set of objective functions be given by the decision-maker according to the importance of the objective functions. The equitable rational preference relation is extended to  $P$ -equitable rational preference relations by Mahmodinejad and Foroutannia [9]. They showed that the concept of  $P$ -equitably efficient solutions is a specific refinement

of Pareto optimality by adding the  $P$ -impartiality and  $P$ -transfer axioms. Moreover, they obtained the  $P$ -equitably efficient solutions by decomposing the original problem into a collection of smaller subproblems and then solved the subproblems by the concept of equitable efficiency.

The equitable optimization method is applied to problems such as portfolio, location, telecommunications, and resource allocation [12, 13, 14, 15, 16]. It should be noted that some authors have used the term “fair” rather than “equitable”.

In this paper, we investigate a priori technique for attaining the decision-maker preferences by introducing the concept of  $A_P$ -equitable efficiency, where the preferences matrix  $A_P$  is based on the partition  $P$  of the index set of objective functions given by the decision-maker. The current study is an extension of some results obtained in [3, 9, 11].

The paper is organized as follows. Terminology and basic concepts are presented in Section 2. In Section 3, we introduce the concept of  $A_P$ -equitable efficiency and give some conditions that ensure that the preference relation  $\preceq_{eA_P}$  is a  $P$ -equitable rational preference relation. In Section 4, the concept of  $A_P^r$ -equitably efficiency is examined to generate a subset of Pareto optimal solutions, for  $r = 1, 2, \dots$ . In addition, a numerical example is provided to confirm the efficiency of this method. Finally, Section 5 concludes the paper.

## 2 Terminology and review of the equitable preference

Let  $\mathbb{R}^m$  be the Euclidean vector space and let  $y', y'' \in \mathbb{R}^m$ . Then  $y' \leq y''$  means  $y'_i \leq y''_i$  for  $i = 1, \dots, m$  and  $y' < y''$  means  $y'_i < y''_i$  for  $i = 1, \dots, m$ , and also  $y' \leq y''$  stands for  $y' \leq y''$  but  $y' \neq y''$ .

Consider a decision problem defined as an optimization problem with  $m$  objective functions. For simplification, we assume, without loss of generality, that the objective functions are to be minimized. The problem can be formulated as follows:

$$\begin{aligned} & \min (f_1(x), f_2(x), \dots, f_m(x)), \\ & \text{subject to } x \in X, \end{aligned} \tag{1}$$

where  $x$  denotes the vector of decision variables in the feasible set  $X$  and  $f(x) = (f_1(x), f_2(x), \dots, f_m(x))$  is the vector function that maps the feasible set  $X$  into the objective (criterion) space  $\mathbb{R}^m$ . We refer to the elements of the objective space as outcome vectors. An outcome vector  $y$  is attainable if it expresses outcomes of a feasible solution, that is,  $y = f(x)$  for some  $x \in X$ . The set of all attainable outcome vectors is denoted by  $Y = f(X)$ .

In the single objective minimization problems, we compare the objective values at different feasible decisions to select the best decision. Decisions are ranked according to the objective values of those decisions, and any decision with the smallest objective value is called an optimal solution. Similarly, to

make the multiobjective optimization model operational, one needs to assume some solution concepts specifying what it means to minimize multiobjective functions. The solution concepts are defined by the properties of the corresponding preference model. We assume that solution concepts depend only on the evaluation of the outcome vectors while not taking into account any other solution properties not represented within the outcome vectors. Thus, we can limit our considerations to the preferred model in the objective space  $Y$ .

In the rest of the section, some basic concepts and definitions of preference relations are reviewed from [3, 6, 9, 11]. Preferences are represented by a weak preference relation with the notation,  $\preceq$ , which allows us to compare pairs of outcome vectors  $y'$  and  $y''$  in the objective space  $Y$ . We say  $y' \preceq y''$  if and only if “ $y'$  is at least as good as  $y''$ ” or “ $y'$  is weakly preferred to  $y''$ ”. In other words,  $y' \preceq y''$  means that the decision-maker thinks that the outcome vector  $y'$  is at least as good as the outcome vector  $y''$ . From  $\preceq$ , we can derive two other important relations on  $Y$ .

**Definition 1.** Let  $y', y'' \in \mathbb{R}^m$  and let  $\preceq$  be a relation of weak preference defined on  $\mathbb{R}^m \times \mathbb{R}^m$ . The strict preference relation,  $\prec$ , is defined by

$$y' \prec y'' \Leftrightarrow (y' \preceq y'' \text{ and not } y'' \preceq y'), \quad (2)$$

and read  $y'$  is strictly preferred to  $y''$ . Also the indifference relation,  $\simeq$ , is defined by

$$y' \simeq y'' \Leftrightarrow (y' \preceq y'' \text{ and } y'' \preceq y'), \quad (3)$$

and read  $y'$  is indifferent to  $y''$ .

**Definition 2.** Preference relations satisfying the following axioms are called equitable rational preference relations:

1. Reflexivity: for all  $y \in \mathbb{R}^m$ ,  $y \preceq y$ .
2. Transitivity: for all  $y', y'', y''' \in \mathbb{R}^m$ ,  $y' \preceq y''$  and  $y'' \preceq y''' \Rightarrow y' \preceq y'''$ .
3. Strict monotonicity: for all  $y \in \mathbb{R}^m$ ,  $y - \epsilon e_i \prec y$  for all  $\epsilon > 0$ , where  $e_i$  denotes the  $i$ th unit vector in  $\mathbb{R}^m$ , for all  $i \in \{1, 2, \dots, m\}$ .
4. Impartial: for all  $y \in \mathbb{R}^m$

$$(y_1, y_2, \dots, y_m) \simeq (y_{\tau(1)}, y_{\tau(2)}, \dots, y_{\tau(m)}),$$

where  $\tau$  stands for an arbitrary permutation of components of  $y$ .

5. Strict transfer principle: for all  $y \in \mathbb{R}^m$  and for all  $i, j \in \{1, 2, \dots, m\}$

$$y_i > y_j \Rightarrow y - \epsilon e_i + \epsilon e_j \prec y,$$

where  $0 < \epsilon < y_i - y_j$ .



A preference relation with the axioms reflexivity, transitivity, and strict monotonicity is called rational preference relation. For  $y', y'' \in Y$ , we say that  $y'$  rationally dominates  $y''$ , and denote by  $y' \prec_r y''$  if and only if  $y' \prec y''$  for all rational preference relations  $\preceq$ . An outcome vector  $y$  is rationally nondominated if and only if there exist no other outcome vector  $y'$  such that  $y'$  rationally dominates  $y$ . Analogously, a feasible solution  $x \in X$  is an efficient or Pareto optimal solution to the multiobjective problem (1) if and only if  $y = f(x)$  is rationally nondominated. It has been shown in [6] that  $y' \prec_r y''$  if and only if  $y' \leq y''$ . As a consequence, we can state that a feasible solution  $x \in X$  is a Pareto optimal solution to the multiobjective problem (1) if and only if there exist no  $x' \in X$  such that  $f_i(x') \leq f_i(x)$  for  $i = 1, 2, \dots, m$ , where at least one strict inequality holds.

The set of all Pareto optimal solutions  $x \in X$  is denoted by  $X_E$  and called the efficient set. The set of all rationally nondominated points  $y = f(x) \in Y$ , where  $x \in X_E$ , is denoted by  $Y_N$  and called the nondominated set.

The equitable rational preference relations allow us to define the concept of equitably efficient solution.

**Definition 3.** Let  $y', y'' \in Y$ . We say that  $y'$  equitably dominates  $y''$ , and denote by  $y' \prec_e y''$  if and only if  $y' \prec y''$  for all equitable rational preference relations  $\preceq$ . An outcome vector  $y$  is equitably nondominated if and only if there exist no other outcome vector  $y'$  such that  $y'$  equitably dominates  $y$ . Analogously, a feasible solution  $x$  is called an equitably efficient solution of the multiobjective problem (1) if and only if  $y = f(x)$  is equitably nondominated.

The set of all equitably efficient solutions  $x \in X$  is denoted by  $X_{eE}$  and called the equitably efficient set. The set of all equitably nondominated points  $y = f(x) \in Y$ , where  $x \in X_{eE}$ , is denoted by  $Y_{eN}$  and called the equitably nondominated set.

**Definition 4.** Let  $y \in \mathbb{R}^m$ .

1. The function  $\theta : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is called an ordering map if and only if  $\theta(y) = (\theta_1(y), \theta_2(y), \dots, \theta_m(y))$ , where  $\theta_1(y) \geq \theta_2(y) \geq \dots \geq \theta_m(y)$  in which  $\theta_i(y) = y_{\tau(i)}$  for  $i = 1, 2, \dots, m$ , and  $\tau$  is a permutation of the set  $\{1, 2, \dots, m\}$ .
2. The function  $\bar{\theta} : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is called a cumulative ordering map if and only if  $\bar{\theta}(y) = (\bar{\theta}_1(y), \bar{\theta}_2(y), \dots, \bar{\theta}_m(y))$ , where  $\bar{\theta}_i(y) = \sum_{j=1}^i \theta_j(y)$  for  $i = 1, 2, \dots, m$  and the ordering map  $\theta$  is given by part (1).

Note that  $\bar{\theta}(y) = \Delta\theta(y)$ , where

$$\Delta = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}.$$

is an  $m \times m$  lower-triangular matrix and relates it to the equitable preference.

A relationship between the weak equitable preference relation  $\preceq_e$  and the Pareto relation has been established in [6]. The following proposition shows finding nondominated points with respect to the weak equitable preference relation  $\preceq_e$  can be done by means of Pareto preference.

**Proposition 1.** [6, Proposition 2.3] For any two vectors  $y', y'' \in Y$ , we have

$$y' \preceq_e y'' \Leftrightarrow \bar{\theta}(y') \leq \bar{\theta}(y'') \Leftrightarrow \Delta\theta(y') \leq \Delta\theta(y''),$$

where the ordering map  $\theta$  and the cumulative ordering map  $\bar{\theta}$  are given by Definition 2.

Now, we review the concept equitably with respect to any matrix  $A$ , which was introduced by Mut and Wiecek [11]. Assume that  $A = (a_{ij})$  is an  $m \times m$  matrix with real entries. Then the cumulative map  $A(\theta) : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is defined by

$$A(\theta(y)) = \left( \sum_{j=1}^m a_{1j}\theta_j(y), \sum_{j=1}^p a_{2j}\theta_j(y), \dots, \sum_{j=1}^m a_{pj}\theta_j(y) \right).$$

**Definition 5.** Let  $y', y'' \in Y$ . We say that  $y'$   $A$ -equitably dominates  $y''$ , and denote by  $y' \prec_{eA} y''$  if and only if  $A(\theta(y')) \leq A(\theta(y''))$ . An outcome vector  $y$  is  $A$ -equitably nondominated if and only if there exist no other outcome vector  $y'$  such that  $y'$   $A$ -equitably dominates  $y$ . Analogously, a feasible solution  $x$  is called an  $A$ -equitably efficient solution of the multiobjective problem (1) if and only if  $y = f(x)$  is  $A$ -equitably nondominated.

The set of all  $A$ -equitably efficient solutions  $x \in X$  is denoted by  $X_{eAE}$  and called the  $A$ -equitably efficient set. The set of all  $A$ -equitably nondominated points  $y = f(x) \in Y$ , where  $x \in X_{eAE}$ , is denoted by  $Y_{eAN}$  and called the  $A$ -equitably nondominated set.

Mut and Wiecek [11, Section 5] examined relationships between cone representations and the axioms of preference relation  $\preceq_{eA}$ . They showed that the relation  $\preceq_{eA}$  satisfies the axioms of reflexivity, transitivity, and impartiality while the weak principle of transfer requires the following condition on the matrix  $A$ .

**Weak transfer principle:** For all  $y \in \mathbb{R}^m$  and for all  $i, j \in \{1, 2, \dots, m\}$

$$y_i > y_j \Rightarrow y - \epsilon e_i + \epsilon e_j \preceq y,$$

where  $0 \leq \epsilon \leq y_i - y_j$ .

**Proposition 2.** [11, Corollary 5.11] Let  $A = [a_1, a_2, \dots, a_p]$ , where  $a_i$ 's are the columns of the matrix  $A$ ,  $i = 1, \dots, p$ . The weak principle of transfer axiom for the generalized equitable preference  $\preceq_{eA}$  is equivalent to the condition

$$a_1 \geq a_2 \geq \cdots \geq a_p,$$

on the matrix  $A$ .

The preference relation  $\preceq_{eA}$  does not satisfy the strict monotonicity and strict principle of transfer axioms in general. Therefore the set of  $A$ -equitably efficient solutions is not contained within the set of equitably efficient solutions and the set of Pareto optimal solutions for the same problem. Foroutannia and Merati extended the work done by Mut and Wiecek and stated new conditions on the matrix  $A$  that guarantee to satisfy these axioms by the preference relation  $\preceq_{eA}$ . They showed that the preference relation  $\preceq_{eA}$  is an equitable rational preference relation if and only if

$$a_1 \geq a_2 \geq \cdots \geq a_m \geq 0,$$

where  $a_i$  is the  $i$ th column of the matrix  $A$ .

The concept of  $P$ -equitable rational preference relation has been introduced by Mahmodinejad and Foroutannia [9]. They studied some theoretical and practical aspects of the  $P$ -equitably efficient solutions and showed that the set of  $P$ -equitably efficient solutions is contained within the set of efficient solutions for the same problem.

**Definition 6.** Let  $M = \{1, 2, \dots, m\}$  be the index set of objective functions  $f = (f_1, f_2, \dots, f_m)$  and let  $n$  be a positive integer such that  $n \leq m$ . A collection  $P = \{P_k \subseteq M : k = 1, 2, \dots, n\}$  is called a decomposition of  $M$ , and also it is said a partition of  $M$  if  $\bigcup_{k=1}^n P_k = M$ , and  $P_i \cap P_j = \emptyset$  for all  $i \neq j$ , where  $i, j \in \{1, 2, \dots, n\}$  and  $P_k$  is index set of objective functions in class  $k$ .

**Definition 7.** Rational preference relations satisfying the following axioms are called  $P$ -equitable rational preference relations:

1.  $P$ -impartiality:  $y_{P_k} \simeq y_{\tau P_k}$  for any permutation  $\tau$  of components of  $y_{P_k}$ ,  $k = 1, \dots, n$ .
2. Strict  $P$ -transfer principle:

$$y_i > y_j \Rightarrow y - \epsilon e_i + \epsilon e_j \prec y,$$

where  $0 < \epsilon < y_i - y_j$  and  $i, j \in P_k$  for  $k = 1, \dots, n$ .

When  $n = 1$ , that is,  $P_1 = \{1, \dots, m\}$ , each  $P$ -equitable rational preference relation becomes an equitable rational preference relation. For more details on the  $P$ -equitable rational preference relation, the reader may refer to [9].

**Definition 8.** Let  $y', y'' \in Y$ . We say that  $y'$   $P$ -equitably dominates  $y''$ , and denote by  $y' \prec_{Pe} y''$  if and only if  $\bar{\theta}(y'_{P_k}) \leq \bar{\theta}(y''_{P_k})$  for  $k = 1, \dots, n$  and  $\bar{\theta}(y'_{P_k}) < \bar{\theta}(y''_{P_k})$  for some  $k \in \{1, \dots, n\}$ . An outcome vector  $y$  is  $P$ -equitably

nondominated if and only if there exist no other outcome vector  $y'$  such that  $y'$   $P$ -equitably dominates  $y$ . Analogously, a feasible solution  $x$  is called an  $P$ -equitably efficient solution of the multiobjective problem (1) if and only if  $y = f(x)$  is  $P$ -equitably nondominated.

The set of all  $P$ -equitably efficient solutions  $x \in X$  is denoted by  $X_{PE}$  and called the  $P$ -equitably efficient set. The set of all  $P$ -equitably nondominated points  $y = f(x) \in Y$ , where  $x \in X_{PE}$ , is denoted by  $Y_{PN}$  and called the  $P$ -equitably nondominated set.

### 3 The concept of $A_P$ -equitable efficiency

In this section, we suggest an a priori method that is based on the preferences matrix. The idea behind this is that the decision-maker classifies the objective functions in different classes and determines a partition  $P = \{P_k \subseteq M : k = 1, 2, \dots, n\}$  of  $\{1, 2, \dots, m\}$  according to the importance of objective functions. The decision-maker should give a preferences matrix  $A_k$  for objective functions in class  $P_k$  for  $k = 1, 2, \dots, n$ . We introduce the matrix  $A_P = A_1 \oplus A_2 \oplus \dots \oplus A_n$ , which is the direct sum of the matrices  $A_1, A_2, \dots, A_n$ , that is,

$$A_P = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_n \end{bmatrix}.$$

The pairwise comparison matrix and its decompositions are one of the ways which the decision-maker can use to provide a preference matrix  $A_k$  for objective functions in the class  $P_k$  ( $k = 1, 2, \dots, n$ ). A pairwise comparison matrix is used to compute for relative priorities of objective functions. The entry  $(i, j)$  of a pairwise comparison matrix expresses the degree of the preference of the  $i$ th objective over the  $j$ th objective. For more details, the reader is referred to [5].

By the matrix  $A_P$ , the cumulative map  $A_P(\theta) : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is defined as

$$A_P(\theta(y)) = (A_1(\theta(y_{P_1})), A_2(\theta(y_{P_2})), \dots, A_n(\theta(y_{P_n}))),$$

for  $y \in Y$ , where  $y_{P_k} = (y_j)_{j \in P_k}$  for  $k = 1, 2, \dots, n$ . Note that  $A_k$  is a  $|P_k| \times |P_k|$  matrix and  $|P_k|$  is the cardinal of the set  $P_k$ .

Suppose that  $y', y'' \in Y$  are two outcome vectors. Throughout this paper, the following notations is used:

$$A_P(\theta(y')) \leq A_P(\theta(y'')) \Leftrightarrow A_k(\theta(y'_{P_k})) \leq A_k(\theta(y''_{P_k})) \quad (k = 1, 2, \dots, n),$$

$$A_P(\theta(y')) \leq A_P(\theta(y'')) \Leftrightarrow (A_P(\theta(y')) \leq A_P(\theta(y'')) \text{ and not } A_P(\theta(y'')) \leq A_P(\theta(y'))),$$

and also

$$A_P(\theta(y')) = A_P(\theta(y'')) \Leftrightarrow (A_P(\theta(y')) \leq A_P(\theta(y'')) \text{ and } A_P(\theta(y'')) \leq A_P(\theta(y'))).$$

The following definitions are necessary for the solution concepts of this paper.

**Definition 9.** Suppose that  $y', y'' \in Y$  are two outcome vectors. We say that  $y'$   $A_P$ -equitably dominates  $y''$  if and only if  $A_P(\theta(y')) \leq A_P(\theta(y''))$ , and that is denoted by  $y' \prec_{eA_P} y''$ . Also we say that  $y$  is an  $A_P$ -equitably nondominated point if and only if there exit no  $y'$  such that  $y' \prec_{eA_P} y$ . A feasible solution  $x \in X$  is an  $A_P$ -equitably efficient solution to the multiobjective problem (1) if and only if  $y = f(x)$  is an  $A_P$ -equitably nondominated point.

The set of all  $A_P$ -equitably efficient solutions  $x \in X$  is denoted by  $X_{eA_P E}$  and called the  $A_P$ -equitably efficient set. The set of all  $A_P$ -equitably nondominated points is denoted by  $Y_{eA_P N}$  and called the  $A_P$ -equitably nondominated set.

Note that the relation  $\prec_{eA_P}$  becomes the equitable relation when  $A_1 = \Delta$  and  $P_1 = \{1, 2, \dots, m\}$ . Moreover, if  $A$  is an arbitrary matrix and  $P_1 = \{1, 2, \dots, m\}$ , then Definition 5 holds. Also for  $A_k = \Delta_{|P_k| \times |P_k|}$  ( $k = 1, 2, \dots, n$ ), Definition 8 holds.

Similar to the relation of  $A_P$ -equitable dominance, we can define the relation of  $A_P$ -equitable indifference,  $\simeq_{eA_P}$ , and the relation of  $A_P$ -equitable weak dominance,  $\preceq_{eA_P}$ . We say that  $y' \simeq_{eA_P} y''$  if and only if  $A_P(\theta(y')) = A_P(\theta(y''))$ , and also that  $y' \preceq_{eA_P} y''$  if and only if  $A_P(\theta(y')) \leq A_P(\theta(y''))$ .

It is clear that the preference relation  $\preceq_{eA_P}$  satisfies the reflexivity, transitivity, and  $P$ -impartiality axioms. In continue, we express some conditions that guarantee the relation  $\preceq_{eA_P}$  is a  $P$ -equitable rational preference relation. Throughout this section, we assume that  $e_i^k \in \mathbb{R}^k$  is the unit vector with the  $i$ th component equal to one and the remaining ones equal to zero, where  $k = 1, 2, \dots$  and  $i \in \{1, 2, \dots, k\}$ .

**Theorem 1.** The strict monotonicity axiom for the preference  $\preceq_{eA_P}$  is equivalent to the condition

$$a_i^k \geq 0 \quad (i = 1, 2, \dots, |P_k|), \quad (4)$$

where  $a_i^k$  is the  $i$ th column of the matrix  $A_k$  and  $k = 1, 2, \dots, n$ .

*Proof.* We first prove that if the matrix  $A_p$  satisfies condition (4), then the strict monotonicity axiom holds for the preference  $\preceq_{eA_P}$ . Let  $y \in Y$ ,  $i \in \{1, 2, \dots, m\}$  and  $y' = y - \epsilon e_i^m$ , for  $\epsilon > 0$ . We show that  $y' \prec_{eA_P} y$ , this means that  $A_j(\theta(y'_{P_j})) \leq A_j(\theta(y_{P_j}))$  for  $j = 1, 2, \dots, n$  and  $A_j(\theta(y'_{P_j})) \leq A_j(\theta(y_{P_j}))$ , for some  $j \in \{1, 2, \dots, n\}$ . There exists an index  $k \in \{1, 2, \dots, n\}$

such that  $i \in P_k$ . For  $j \in \{1, 2, \dots, n\} - \{k\}$ , we have  $y'_{P_j} = y_{P_j}$ , so  $A_j(\theta(y'_{P_j})) = A_j(\theta(y_{P_j}))$ . Since  $y'_{P_k} = y_{P_k} - \epsilon e_i^{|P_k|}$ , we have  $y'_{P_k} \leq y_{P_k}$ . Hence

$$\theta(y'_{P_k}) \leq \theta(y_{P_k}).$$

Because  $a_j^k \geq 0$  for  $j = 1, 2, \dots, |P_k|$ , we obtain

$$\sum_{j=1}^{|P_k|} a_{ij}^k \theta_j(y'_{P_k}) \leq \sum_{j=1}^{|P_k|} a_{ij}^k \theta_j(y_{P_k}) \quad (i = 1, 2, \dots, |P_k|),$$

and there is  $i' \in \{1, 2, \dots, |P_k|\}$  such that

$$\sum_{j=1}^{|P_k|} a_{i'j}^k \theta_j(y'_{P_k}) < \sum_{j=1}^{|P_k|} a_{i'j}^k \theta_j(y_{P_k}).$$

So,  $A_k(\theta(y'_{P_k})) \leq A_k(\theta(y_{P_k}))$  and the proof is complete.

Conversely, suppose that the strict monotonicity axiom holds for the preference  $\preceq_{e_{A_P}}$ . For any  $k \in \{1, 2, \dots, n\}$ , we define the vector  $y^j \in \mathbb{R}^m$  such that

$$y_{P_i}^j = \begin{cases} e_1^{|P_i|} + \dots + e_j^{|P_i|} & \text{for } j \leq |P_i| \\ 0 & \text{otherwise,} \end{cases}$$

for  $j = 1, 2, \dots, \max_{k=1,2,\dots,n} |P_k|$  and  $i = 1, 2, \dots, n$ . Let  $e \in \mathbb{R}^m$  be defined as  $e_{P_k} = e_1^{|P_k|}$ , for  $k = 1, 2, \dots, n$ . The strict monotonicity property implies that

$$y^j - e \prec_{e_{A_P}} y^j, \quad (j = 1, 2, \dots, \max_{k=1,2,\dots,n} |P_k|),$$

which concludes that  $a_j^k \geq 0$  for  $j = 1, 2, \dots, |P_k|$  and  $k = 1, 2, \dots, n$ . Hence, the matrix  $A_P$  fulfills condition (4).  $\square$

**Remark 1.** If  $n = 1$  and  $P_1 = \{1, 2, \dots, m\}$ , then Theorem 3.1 in [3] holds.

To establish the strict  $P$ -transfer principle for the preference  $\preceq_{e_{A_P}}$ , we need the following statement.

**Proposition 3.** [3, Proposition 3.1] Let  $x = (x_1, x_2, \dots, x_m)$  and  $y = (y_1, y_2, \dots, y_m)$  be two vectors in  $\mathbb{R}^m$  such that

$$\sum_{j=1}^i x_j \leq \sum_{j=1}^i y_j \quad (i = 1, 2, \dots, m),$$

where the strict inequality holds at least once. Also let  $W = [w^1, w^2, \dots, w^m]$  be a matrix  $m \times m$  and let  $w^i$ 's be the columns of the matrix  $W$  for  $i = 1, \dots, m$ . If

$$w^1 \geq w^2 \geq \dots \geq w^m \geq 0, \quad (5)$$

then

$$\sum_{j=1}^m w_{ij}x_j \leq \sum_{j=1}^m w_{ij}y_j, \quad (i = 1, 2, \dots, m),$$

where the strict inequality holds at least once.

**Corollary 1.** Let  $x$  and  $y$  be vectors in  $\mathbb{R}^m$ . If  $w_1 \geq w_2 \geq \dots \geq w_m \geq 0$  and

$$\sum_{i=1}^n x_i \leq \sum_{i=1}^n y_i \quad (n = 1, 2, \dots, m),$$

then

$$\sum_{i=1}^m w_i x_i \leq \sum_{i=1}^m w_i y_i.$$

*Proof.* Let the matrix  $W = (w_{ij})$  be defined by  $w_{1j} = w_j$  for  $j = 1, 2, \dots, m$  and  $w_{ij} = 0$  for  $i = 2, \dots, m$ . Using Proposition 3, the proof is obvious.  $\square$

**Theorem 2.** The strict  $P$ -transfer principle for the preference  $\preceq_{eA_P}$  is equivalent to the following condition:

$$a_1^k \geq a_2^k \geq \dots \geq a_{|P_k|}^k, \quad (6)$$

where  $a_i^k$ 's are the columns of the matrix  $A_k$ , for  $i = 1, 2, \dots, |P_k|$  and  $k = 1, \dots, n$ .

*Proof.* Let  $y \in Y$ ,  $i, j \in P_k$ ,  $y_i > y_j$ , and  $y' = y - \epsilon e_i^m + \epsilon e_j^m$ , where  $0 < \epsilon < y_i - y_j$ . We show that  $y' \prec_{eA_P} y$ . This means that  $A_l(\theta(y'_{P_l})) \leq A_l(\theta(y_{P_l}))$  for  $l = 1, 2, \dots, n$  and  $A_l(\theta(y'_{P_l})) \leq A_l(\theta(y_{P_l}))$ , for some  $l \in \{1, 2, \dots, n\}$ . If  $l \in \{1, 2, \dots, n\} - \{k\}$ , then  $y'_{P_l} = y_{P_l}$ , so  $A_l(\theta(y'_{P_l})) = A_l(\theta(y_{P_l}))$ .

Let  $\alpha \in R$  be such that

$$a_1^k + \alpha \geq a_2^k + \alpha \geq \dots \geq a_{|P_k|}^k + \alpha \geq 0. \quad (7)$$

Since  $y'_{P_k} = y_{P_k} - \epsilon e_i^{|P_k|} + \epsilon e_j^{|P_k|}$  and the equitable preference  $\preceq_e$  satisfies the strict transfer principle, we have  $\bar{\theta}(y'_{P_k}) \leq \bar{\theta}(y_{P_k})$ . Hence, by Proposition 3 and (7), we have

$$\sum_{t=1}^{|P_k|} (a_{st}^k + \alpha) \theta_t(y'_{P_k}) \leq \sum_{t=1}^{|P_k|} (a_{st}^k + \alpha) \theta_t(y_{P_k}) \quad (s = 1, 2, \dots, |P_k|),$$

where the strict inequality holds at least  $s$ . On other hand,  $\sum_{t=1}^{|P_k|} \theta_t(y'_{P_k}) = \sum_{t=1}^{|P_k|} \theta_t(y_{P_k})$  implies that

$$\sum_{t=1}^{|P_k|} a_{st}^k \theta_t(y'_{P_k}) \leq \sum_{t=1}^{|P_k|} a_{st}^k \theta_t(y_{P_k}) \quad (s = 1, 2, \dots, |P_k|),$$

where the strict inequality holds at least  $s$ . Hence, we have the desired result.

Conversely, suppose that the strict  $P$ -transfer axiom holds for the preference  $\preceq_{e_{AP}}$ . We define the vector  $y^j \in \mathbb{R}^m$  such that

$$y_{P_i}^j = \begin{cases} 2e_1^{|P_i|} + \dots + 2e_j^{|P_i|} & \text{for } j \leq |P_i| - 1 \\ 0 & \text{otherwise,} \end{cases}$$

for  $j = 1, 2, \dots, \max_{i=1,2,\dots,n} |P_i|$  and  $i = 1, 2, \dots, n$ .

Let  $e^j \in \mathbb{R}^m$  be defined as  $e_{P_i}^j = e_j^{|P_i|}$  for  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, \max_{i=1,2,\dots,n} |P_i|$ . The strict  $P$ -transfer property implies that

$$y^j - e^j + e^{j+1} \prec_{e_{AP}} y^j, \quad (j = 1, 2, \dots, \max_{i=1,2,\dots,n} |P_i|),$$

which conclude the desired result.  $\square$

**Remark 2.** If  $n = 1$  and  $P_1 = \{1, 2, \dots, m\}$ , then Theorem 3.2 in [3] and Corollary 5.11 in [11] hold.

Theorems 1 and 2 imply that the preference relation  $\preceq_{e_{AP}}$  is a  $P$ -equitable rational preference relation if and only if the matrix  $A_P$  fulfills conditions (4) and (6), that is,

$$a_1^k \geq a_2^k \geq \dots \geq a_{|P_k|}^k \geq 0, \quad (8)$$

for  $k = 1, \dots, n$ .

**Theorem 3.** Suppose that the matrix  $A_P = A_1 \oplus A_2 \oplus \dots \oplus A_n$  satisfies condition (8). If  $x \in X$  is an  $A_P$ -equitably efficient solution of multiobjective problem (1), then it is a  $P$ -equitably efficient solution of multiobjective problem (1). Moreover,  $Y_{e_{AP}N} \subset Y_{PN}$ .

*Proof.* Suppose that  $x$  is an  $A_P$ -equitably efficient solution to (1). If  $x$  is not a  $P$ -equitable efficient solution to (1), then a feasible solution  $x'$  must exist such that the outcome vectors  $y = f(x)$  and  $y' = f(x')$  satisfy  $y' \prec_P y$ , so  $\bar{\theta}(y'_{P_k}) \leq \bar{\theta}(y_{P_k})$  for  $k = 1, \dots, n$ . Using Proposition 3, we deduce that

$$\sum_{j=1}^{|P_k|} a_{ij}^k \theta_j(y'_{P_k}) \leq \sum_{j=1}^{|P_k|} a_{ij}^k \theta_j(y_{P_k}) \quad (i = 1, 2, \dots, |P_k|),$$



where the strict inequality holds at least once. Hence  $y' \prec_{eAP} y$ , which contradicts the equitable  $A_P$ -efficiency of  $x$ .  $\square$

**Remark 3.** If  $n = 1$  and  $P_1 = \{1, 2, \dots, m\}$ , then Theorem 3.3 in [3] holds.

Since the  $P$ -equitably efficient set is contained within the efficient set, by applying Theorem 3, we can conclude  $X_{eAPE} \subset X_{PE} \subset X_E$ , and hence  $Y_{eAPN} \subset Y_{PN} \subset Y_N$ .

In general, the preference relation  $\preceq_{eAP}$  does not satisfy the strict monotonicity and the strict  $P$ -transfer axioms. Also condition (8) is necessary in Theorem 3. The truth of these statements is examined by the following example.

**Example 1.** Let

$$X = Y = \{(y_1, y_2) : y_1^2 + y_2^2 \leq 1 \text{ and } y_2 \geq y_1\}.$$

If  $n = 1$ ,  $A_P = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ , a  $y = \begin{bmatrix} -1/2 \\ 1/2 \end{bmatrix}$ , and  $\epsilon = 1/2$ , then  $y - \frac{1}{2}e_2 \not\prec_{eAP} y$  and  $y - \frac{1}{2}e_2 + \frac{1}{2}e_1 \not\prec_{eAP} y$ . Hence, the preference relation  $\preceq_{eAP}$  does not necessarily satisfy the strict monotonicity and the strict  $P$ -transfer axioms. Also, we have

$$Y_N = \left\{ (y_1, y_2) : y_1^2 + y_2^2 = 1, -1 \leq y_1 \leq \frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}} \leq y_2 \leq 0 \right\},$$

$$Y_{eAN} = \left\{ (y_1, y_2) : y_1^2 + y_2^2 = 1, -1 \leq y_1 \leq 0, 0 \leq y_2 \leq 1 \right\}.$$

Moreover  $Y_{PN} = \{(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}})\}$ . Hence,  $Y_{eAPN} \not\subset Y_{PN}$  and  $Y_{eAPN} \not\subset Y_N$ .

Note that Definition 9 permits one to express  $A_P$ -equitable efficiency for problem (1) in terms of the standard efficiency for the multiobjective problem with objectives  $A_k(\theta(f_{P_k}(x)))$ :

$$\min\{A_P(\theta(f(x))) : x \in X\}. \quad (9)$$

**Theorem 4.** A feasible solution  $x \in X$  is an  $A_P$ -equitably efficient solution to the multiobjective problem (1) if and only if it is an efficient solution to the multiobjective problem (9).

*Proof.* The proof is trivial by Definition 9.  $\square$

**Remark 4.** If  $n = 1$  and  $P_1 = \{1, 2, \dots, m\}$ , then [11, Corollary 5.3] holds. Also, if  $A_k = \Delta_{|P_k| \times |P_k|}$  for all  $k = 1, 2, \dots, n$ , then [9, Theorem 3.2] holds.

## 4 The concept of $A_P^\infty$ -equitably efficiency

In this section, we investigate the inclusion relations among  $A_P^r$ -equitably efficient set,  $P$ -equitably efficient set, and efficient set. Then, we introduce

the concept of  $A_P^\infty$ -equitable efficient to generate a subset of efficient solutions, which aims to offer a limited number of representative solutions to the decision-maker.

Let  $A_P = A_1 \oplus A_2 \oplus \cdots \oplus A_n$  and  $B_P = B_1 \oplus B_2 \oplus \cdots \oplus B_n$  be two  $m \times m$  matrices. The combined cumulative map  $(A_P \circ B_P)(\theta) : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is defined by

$$(A_P \circ B_P)(\theta(y)) = A_P(\theta(B_P(\theta(y)))) ,$$

for  $y \in Y$ . If  $y', y'' \in Y$ , using the combined cumulative map, then we can say that  $y'$   $(A_P \circ B_P)$ -equitably dominates  $y''$  if and only if

$$A_P(\theta(B_P(\theta(y')))) \leq A_P(\theta(B_P(\theta(y'')))) ,$$

and that is denoted by  $y' \prec_{e(A_P \circ B_P)} y''$ . Also we say that  $y$  is an  $(A_P \circ B_P)$ -equitably nondominated point if and only if there exit no  $y'$  such that  $y' \prec_{e(A_P \circ B_P)} y$ . A feasible solution  $x \in X$  is an  $(A_P \circ B_P)$ -equitably efficient solution to the multiobjective problem (1) if and only if  $y = f(x)$  is an  $(A_P \circ B_P)$ -equitably nondominated point.

In order to make calculations easier, we present a condition on the matrix  $B_P$  whereby the vector  $B_P(\theta(y))$  is decreasing for every outcome vector  $y \in Y$ .

**Proposition 4.** The condition

$$r_{ij}^{B_k} \geq r_{(i+1)j}^{B_k} \quad (j = 1, 2, \dots, |P_k|), \quad (10)$$

where  $r_{ij}^{B_k} = \sum_{t=1}^j b_{it}^k$  for  $i = 1, 2, \dots, |P_k| - 1$  and  $k = 1, 2, \dots, n$ , is equivalent to the statement that  $B_k(\theta(y_{P_k}))$  is decreasing for all  $y \in \mathbb{R}^m$ .

*Proof.* Put  $\theta_{|P_k|+1}(y_{P_k}) = 0$ , by the Abel summation

$$\sum_{j=1}^{|P_k|} b_{ij}^k \theta_j(y_{P_k}) = \sum_{j=1}^{|P_k|} r_{ij}^{B_k} (\theta_j(y_{P_k}) - \theta_{j+1}(y_{P_k})),$$

we obtain the desired result.  $\square$

By the above proposition, we conclude that  $\theta(B_P(\theta(y))) = B_P(\theta(y))$  and

$$(A_P \circ B_P)(\theta(y)) = A_P(\theta(B_P(\theta(y)))) = (A_P B_P)(\theta(y)),$$

where  $A_P B_P$  is the product of the matrices  $A_P$  and  $B_P$ , and also

$$A_P B_P = A_1 B_1 \oplus A_2 B_2 \oplus \cdots \oplus A_n B_n. \quad (11)$$

It follows from what has been said above that the relation  $\preceq_{e(A_P \circ B_P)}$  is equivalent to the relation  $\preceq_{eA_P B_P}$ , when the matrix  $B_P$  satisfies condition (10).

In continue, we study the relationship between  $A_P$ -equitably efficient solutions and  $(A_P \circ B_P)$ -equitably efficient solutions. To do this, we require the following statements.

**Proposition 5.** Let  $A = (a_1, a_2, \dots, a_m)$  and  $B = (b_1, b_2, \dots, b_m)$  be two  $m \times m$  matrices, where  $a_j$  and  $b_j$  are the  $j$ th column of the matrices  $A$  and  $B$ , respectively. If  $D = AB = (d_1, d_2, \dots, d_m)$ , where  $d_j$  is the  $j$ th column of the matrix  $D$ , then the following statements hold:

- (i) If  $a_j \geq 0$  and  $b_j \geq 0$  for all  $j = 1, 2, \dots, m$ , then  $d_j \geq 0$  for all  $j = 1, 2, \dots, m$ .
- (ii) If  $a_j \geq 0$  for  $j = 1, 2, \dots, m$  and  $b_j \geq b_{j+1}$  for  $j = 1, 2, \dots, m-1$ , then  $d_j \geq d_{j+1}$  for  $j = 1, 2, \dots, m-1$ .
- (iii) If  $r_{i,j}^A = \sum_{k=1}^j a_{ik}$  and  $r_{i,j}^B = \sum_{k=1}^j b_{ik}$  are decreasing with respect to  $i$  for all  $j = 1, 2, \dots, m$ , and also if  $r_{i,j}^A \geq 0$  for  $i, j = 1, 2, \dots, m$ , then  $r_{i,j}^D = \sum_{k=1}^j d_{ik}$  is decreasing with respect to  $i$  for all  $j = 1, 2, \dots, m$ .

*Proof.* (i) We have

$$d_j = \left( \sum_{k=1}^m a_{ik} b_{kj} \right)_{i=1}^m.$$

The condition  $b_j \geq 0$  implies that  $b_{kj} \geq 0$  for all  $k = 1, 2, \dots, m$  and  $b_{k'j} > 0$  for some  $k' \in \{1, 2, \dots, m\}$ . Also,  $a_{k'} \geq 0$  concludes that  $a_{ik'} \geq 0$  for any  $i = 1, 2, \dots, m$  and  $a_{i'k'} > 0$  for some  $i' \in \{1, 2, \dots, m\}$ . Thus  $a_{ik} b_{kj} \geq 0$  for any  $i, k = 1, 2, \dots, m$  and  $a_{i'k'} b_{k'j} > 0$ , which means that  $d_j \geq 0$ .

(ii) The condition  $b_j \geq b_{j+1}$  implies that  $b_{kj} - b_{k(j+1)} \geq 0$  for all  $k = 1, 2, \dots, m$  and  $b_{k'j} - b_{k'(j+1)} > 0$  for some  $k' \in \{1, 2, \dots, m\}$ . Also,  $a_{k'} \geq 0$  concludes that  $a_{ik'} \geq 0$  for any  $i = 1, 2, \dots, m$  and  $a_{i'k'} > 0$  for some  $i' \in \{1, 2, \dots, m\}$ . Thus  $a_{ik}(b_{kj} - b_{k(j+1)}) \geq 0$  for  $i, k = 1, 2, \dots, m$  and  $a_{i'k'}(b_{k'j} - b_{k'(j+1)}) > 0$ , which means that

$$\sum_{k=1}^m a_{ik} b_{kj} \geq \sum_{k=1}^m a_{ik} b_{k(j+1)} \quad (i = 1, 2, \dots, m),$$

and the strict inequality holds when  $i = i'$ . Hence  $d_j \geq d_{j+1}$  for  $j = 1, 2, \dots, m-1$ .

(iii) Since  $r_{i,j}^B$  is decreasing with respect to  $i$  for  $j = 1, 2, \dots, m$ , we obtain

$$\sum_{t=1}^n b_{it} \geq \sum_{t=1}^n b_{(i+1)t} \quad (n = 1, 2, \dots, m).$$

For all  $j \in \{1, 2, \dots, m\}$ , we set  $w_t = \sum_{k=1}^j a_{tk} = r_{tj}^A$ . Using Corollary 1, we see that

$$\sum_{t=1}^m \sum_{k=1}^j a_{tk} b_{it} \geq \sum_{t=1}^m \sum_{k=1}^j a_{tk} b_{(i+1)t}.$$

Therefore

$$\sum_{k=1}^j \sum_{t=1}^m a_{tk} b_{it} \geq \sum_{k=1}^j \sum_{t=1}^m a_{tk} b_{(i+1)t},$$

and

$$\sum_{k=1}^j d_{ik} \geq \sum_{k=1}^j d_{(i+1)k},$$

for all  $j = 1, 2, \dots, m$ . This completes the proof of part (iii).  $\square$

**Theorem 5.** Let  $A_P = A_1 \oplus A_2 \oplus \dots \oplus A_n$  and  $B_P = B_1 \oplus B_2 \oplus \dots \oplus B_n$  be two  $m \times m$  matrices. We have the following statements.

- (i) If the matrix  $A_P$  satisfies condition (4) and the matrix  $B_P$  satisfies condition (8), then the matrix  $A_P B_P$  fulfills condition (8). Thus, the preference relation  $\preceq_{e(A_P B_P)}$  is a  $P$ -equitable rational preference relation. Moreover, if the matrix  $B_P$  satisfies condition (10), then the preference relation  $\preceq_{e(A_P \circ B_P)}$  is a  $P$ -equitable rational preference relation.
- (ii) If the matrices  $A_P$  and  $B_P$  satisfy condition (10) and also if the matrix  $A_P$  fulfills condition (4), then the matrix  $A_P B_P$  satisfies condition (10).

*Proof.* By using relation (11) and Proposition 5, we obtain the desired results.  $\square$

**Theorem 6.** Let  $A_P = A_1 \oplus A_2 \oplus \dots \oplus A_n$  and  $B_P = B_1 \oplus B_2 \oplus \dots \oplus B_n$  be two  $m \times m$  matrices. Also let the matrix  $A_P$  satisfy condition (4) and the matrix  $B_P$  satisfy condition (8). If  $y'$  and  $y''$  are two outcome vectors, then

$$\begin{aligned} y' \prec_{eB_P} y'' &\implies y' \prec_{e(A_P B_P)} y'', \\ y' \preceq_{eB_P} y'' &\implies y' \preceq_{e(A_P B_P)} y''. \end{aligned}$$

Hence  $Y_{e(A_P B_P)N} \subset Y_{eB_P N}$ , which implies that  $X_{e(A_P B_P)E} \subset X_{eB_P E}$ . Moreover if the matrix  $B_P$  satisfies condition (10), then  $Y_{e(A_P \circ B_P)N} \subset Y_{eB_P N}$  and  $X_{e(A_P \circ B_P)E} \subset X_{eB_P E}$ .

*Proof.* Let  $y', y'' \in Y$  and  $y' \prec_{eB_P} y''$ . Then  $B_k(\theta(y'_{P_k})) \leq B_k(\theta(y''_{P_k}))$  for  $k = 1, 2, \dots, n$  and  $B_{k'}(\theta(y'_{P_{k'}})) \leq B_{k'}(\theta(y''_{P_{k'}}))$  for some  $k' \in \{1, 2, \dots, n\}$ . Hence

$$\sum_{j=1}^{|P_k|} b_{ij}^k \theta_j(y'_{P_k}) \leq \sum_{j=1}^{|P_k|} b_{ij}^k \theta_j(y''_{P_k}) \quad (i = 1, 2, \dots, |P_k| \text{ and } k = 1, 2, \dots, n),$$

and there exists  $i' \in \{1, 2, \dots, |P_{k'}|\}$  such that

$$\sum_{j=1}^{|P_{k'}|} b_{i'j}^{k'} \theta_j(y'_{P_{k'}}) < \sum_{j=1}^{|P_{k'}|} b_{i'j}^{k'} \theta_j(y''_{P_{k'}}).$$

Now according to condition (4), we have  $a_{ti}^k \geq 0$  for  $i = 1, 2, \dots, |P_k|$  and  $k = 1, 2, \dots, n$ , and there exists  $t' \in \{1, 2, \dots, |P_{k'}|\}$  such that  $a_{t'i'}^{k'} > 0$ . This implies that

$$\sum_{j=1}^{|P_k|} (A_k B_k)_{tj} \theta_j(y'_{P_k}) \leq \sum_{j=1}^{|P_k|} (A_k B_k)_{tj} \theta_j(y''_{P_k})$$

$$(t = 1, 2, \dots, |P_k| \text{ and } k = 1, 2, \dots, n),$$

and the strict inequality holds when  $k = k'$  and  $t = t'$ . Therefore  $y' \prec_{e(A_P B_P)} y''$ . Moreover, suppose that the matrix  $B_P$  fulfills condition (10). Since the preference relations  $\prec_{e(A_P B_P)}$  and  $\prec_{e(A_P \circ B_P)}$  are equivalent, the proof is complete.  $\square$

Let  $A_P = A_1 \oplus A_2 \oplus \dots \oplus A_n$  be an  $m \times m$  matrix and let  $r = 1, 2, \dots$ . The cumulative map  $A_P^r(\theta) : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is defined as

$$A_P^r(\theta(y)) = \underbrace{(A_P \circ A_P \circ \dots \circ A_P)}_{r\text{-times}}(\theta(y)),$$

for  $y \in Y$ . If conditions (10) and (4) are satisfied by the matrix  $A_P$ , then

$$A_P^r(\theta(y)) = \underbrace{(A_P A_P \dots A_P)}_{r\text{-times}}(\theta(y)).$$

The following statement states the relationship among  $A_P^r$ -equitable efficient solutions,  $P$ -equitable efficient solutions, and efficient solutions to multiobjective problem (1).

**Corollary 2.** Suppose that the matrix  $A_P$  satisfies conditions (8) and (10). Then  $Y_{eA_P^{r+1}N} \subset Y_{eA_P^rN} \subset Y_{PN} \subset Y_N$ . Moreover,  $X_{eA_P^{r+1}E} \subset X_{eA_P^rE} \subset X_{PE} \subset X_E$ .

*Proof.* The first inclusion follows by replacing  $A_P^r$  instead of  $B_P$ , in Theorem 6. Also, by applying Theorem 3, we deduce the second inclusion.  $\square$

By using Corollary 2, we conclude the following statement for  $P_1 = \{1, 2, \dots, m\}$ .

**Corollary 3.** Suppose that the matrix  $A$  satisfies conditions (8) and (10). Then  $Y_{eA^{r+1}N} \subset Y_{eA^rN} \subset Y_{eN} \subset Y_N$ . Moreover,  $X_{eA^{r+1}E} \subset X_{eA^rE} \subset X_{eE} \subset X_E$ .

Condition (8) in the above results are necessary. To investigate this fact, we give the following example.

**Example 2.** Let  $Y$  and  $A_P$  be defined as in Example 1. Although condition (10) holds, condition (8) does not hold, and we have

$$\begin{aligned} Y_{A^{4r}eN} &= Y_N = \left\{ (y_1, y_2) : y_1^2 + y_2^2 = 1, -1 \leq y_1 \leq \frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}} \leq y_2 \leq 0 \right\}, \\ Y_{A^{4r+1}eN} &= \left\{ (y_1, y_2) : y_1^2 + y_2^2 = 1, -1 \leq y_1 \leq 0, 0 \leq y_2 \leq 1 \right\}, \\ Y_{eA^{4r+2}N} &= \left\{ (y_1, y_2) : y_1^2 + y_2^2 = 1, 0 \leq y_1 \leq \frac{1}{\sqrt{2}}, 0 \leq y_2 \leq 1 \right\}, \\ Y_{A^{4r+3}eN} &= \left\{ (y_1, -y_1) : y_1^2 + y_2^2 = 1, y_2 = y_1 \right\}, \end{aligned}$$

for  $r = 0, 1, 2, \dots$ . We observe that Corollary 2 does not hold.

According to Corollary 2, we offer an algorithm to compute the  $A_P^r$ -equitably efficient solutions to the multiobjective problem (1).

---

**Algorithm 1**

---

Input: Consider the feasible solution  $X$  and the objective functions  $f$  as in problem (1). Determine a partition  $P = \{P_1, P_2, \dots, P_n\}$  of  $\{1, 2, \dots, m\}$ , a matrix  $A_P$ , and an integer  $r \in \{1, 2, \dots\}$ , according to the decision-maker.

Step 1: Put  $X_1 = X$  and  $k = 1$ .

Step 2: Solve the following multiobjective problem

$$\min\{A_P^k(\theta(f(x))) : x \in X_k\}. \quad (12)$$

Step 3: If  $k = r$ , stop. Otherwise, put  $X_{k+1} = X_{eA_P^kE}$  and  $k = k + 1$ , go to Step 2.

Output: The set  $X_r$  is  $A_P^r$ -equitably efficient set.

---

In the first iteration of Algorithm 1, the  $A_P$ -equitably efficient solutions to the multiobjective problem (1) are computed. Then these solutions are gradually reduced in the next iterations. Finally, the  $A_P^r$ -equitably efficient solutions are obtained in the last iteration.

In the following example, we investigate Corollary 2 and Algorithm 1 and show that  $A_P^r$ -equitably efficient sets are reducing when  $r$  is increasing. For this purpose, a large number of random solutions are generated for the scalable test function. From this large set of solutions, efficient solutions,  $P$ -equitably efficient solutions, and  $A_P^r$ -equitably efficient solutions are calculated for  $r = 1, 2, 3$ .

**Example 3.** The test problem considered is the F1 (see [2])

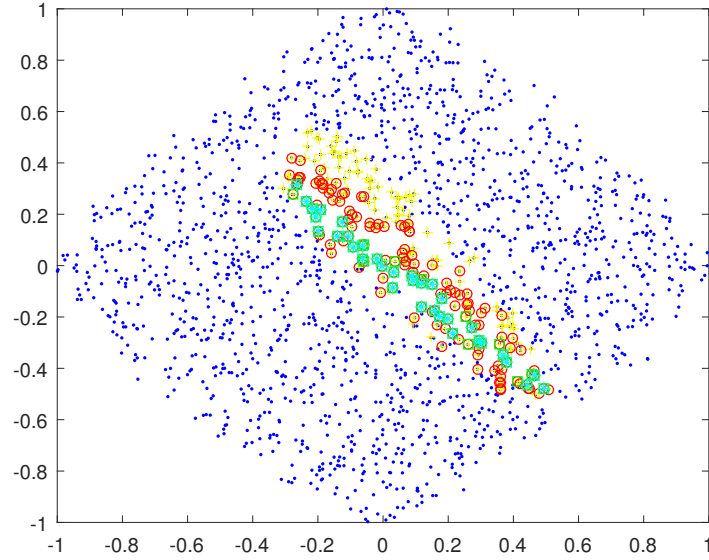


Figure 1: Efficient solutions,  $P$ -equitably efficient solutions, and  $A_P^r$ -equitably efficient solutions of the  $F1$  problem (2 variables and 6 objectives) for  $r = 1, 2, 3$ .

$$\begin{aligned} \min_{x \in R^2} \quad & y = \{f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x)\} \\ & f_1(x) = x_1^2 + (x_2 + 1)^2 \\ & f_2(x) = (x_1 - 0.5)^2 + (x_2 + 0.5)^2 \\ & f_3(x) = (x_1 - 1)^2 + x_2^2 \\ & f_4(x) = (x_1 + 1)^2 + x_2^2 \\ & f_5(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \\ & f_6(x) = x_1^2 + (x_2 - 1)^2 \\ & x_1, x_2 \in [-1, 1]. \end{aligned}$$

In Figure 1 from 3000 random solutions, 1804 solutions (blue point) are efficient. Let  $P_1 = \{1, 2, 3\}$ ,  $P_2 = \{4, 5, 6\}$ , and

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 0.5 & 0 \\ 0.4 & 0.4 & 0.2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 0.4 & 0 \\ 0.4 & 0.3 & 0.2 \end{bmatrix},$$

be given by the decision-maker. We obtain 230  $P$ -equitably efficient solutions, 144  $A_P$ -equitably efficient solutions, 44  $A_P^2$ -equitably efficient solutions, and

34  $A_P^3$ -equitably efficient solutions, which are shown by yellow plus sign, red circles, green square, and cyan star, respectively, in Figure 1.

We assume that the matrix  $A_P$  satisfies conditions (8) and (10). Using the results above, we can define infinite order dominance as follows:

$$\prec_{eA_P^\infty} = \bigcup_{r \in \mathbb{N}} \prec_{eA_P^r},$$

where  $\mathbb{N} = \{1, 2, \dots\}$ . This means that,

$$y' \prec_{eA_P^\infty} y'' \Leftrightarrow y' \prec_{eA_P^r} y'' \quad (\text{for some } r \in \mathbb{N}).$$

**Definition 10.** The outcome vector  $y$  is  $A_P^\infty$ -equitably nondominated if and only if there exist no other outcome vector  $y'$  such that  $y' \prec_{eA_P^\infty} y$ . Analogously, a feasible solution  $x$  is called an  $A_P^\infty$ -equitably efficient solution to the multiobjective problem (1) if and only if  $y = f(x)$  is  $A_P^\infty$ -equitably nondominated.

**Corollary 4.** If the matrix  $A_P$  satisfies conditions (8) and (10), then  $Y_{eA_P^\infty N} = \bigcap_{r \in \mathbb{N}} Y_{eA_P^r N}$  and  $Y_{eA_P^\infty N} \subset Y_{PN} \subset Y_N$ . Moreover,  $X_{eA_P^\infty E} \subset X_{PE} \subset X_E$ .

*Proof.* By applying Definition 10 and Corollary 2, the proof is trivial.  $\square$

Corollary 4 indicates that to reduce Pareto optimal solutions and  $P$ -equitably efficient solutions, we can use  $A_P^\infty$ -equitably efficient solutions.

For  $n = 1$  and  $P_1 = \{1, 2, \dots, m\}$ , by applying Corollary 4, we conclude the following statement.

**Corollary 5.** Suppose that the matrix  $A$  satisfies conditions (8) and (10). Then  $Y_{eA^\infty N} = \bigcap_{r \in \mathbb{N}} Y_{eA^r N}$  and  $Y_{eA^\infty N} \subset Y_{eN} \subset Y_N$ . Moreover,  $X_{eA^\infty E} \subset X_{eE} \subset X_E$ .

## 5 Conclusion

In this paper, we focused on a new concept of rational  $A_P$ -equitable efficiency for solving the multiobjective optimization problems, where the preferences matrix  $A_P$  is given by the decision-maker. This concept was obtained by rational preference relations on the certain cumulative vector  $A_P(\theta(y))$  for  $y \in Y$ . We examined some conditions that ensure the preference relation  $\preceq_{eA_P}$  is a  $P$ -equitable rational preference relation. Moreover, we expressed the concept of  $A_P^r$ -equitable efficiency to generate a subset of Pareto optimal solutions for  $r = 1, 2, \dots$ . Also, we proved that the  $A_P^r$ -equitably efficient sets are decreasing with respect to  $r$  and that the intersection of these sets is the  $A_P^\infty$ -equitably efficient set. Furthermore, an experiment was carried out on randomly generated solutions in order to better compare the efficient



solutions, the  $P$ -equitably efficient solutions, and the  $A_P^r$ -equitably efficient solutions. This experiment indicated that the size of the  $A_P^r$ -equitably efficient sets is considerably smaller than the size of the efficient set.

## References

- [1] Baatar, D. and Wiecek, M.M. *Advancing equitability in multiobjective programming*, Comput. Math. Appl. 52(1-2) (2006), 225–234.
- [2] Farina, M. and Amato, P. *On the optimal solution definition for many criteria optimization problems*, In Proceedings of the NAFIPS-FLINT International Conference (2002), 233–238.
- [3] Foroutannia, D. and Merati, M. *Generalisation of  $A$ -equitable preference in multiobjective optimization problems*, Optimization, 70(9) (2021), 1859–1874.
- [4] Hwang, C.L. and Masud, A. *Multiple objective decision making, methods and applications: A state of the art survey*, Lecture Notes in Economics and Mathematical Systems, vol. 164, Springer-Verlag, Berlin, 1979.
- [5] Jalao, E.R., Wu, T. and Shunk, D. *An intelligent decomposition of pairwise comparison matrices for large-scale decisions*, Eur. J. Oper. Res. 238(1) (2014), 270–280.
- [6] Kostreva, M.M. and Ogryczak, W. *Linear optimization with multiple equitable criteria*, RAIRO Oper. Res. 33(3) (1999), 275–297.
- [7] Kostreva, M.M., Ogryczak, W. and Wierzbicki, A. *equitable aggregations in multiple criteria analysis*, Eur. J. Oper. Res. 158(2) (2004), 362–377.
- [8] Lorenz, M.O. *Methods of measuring the concentration of wealth*, American Statistical Association, New Series, 70 (1905), 209–219.
- [9] Mahmodinejad, A. and Foroutannia, D. *Piecewise equitable efficiency in multiobjective programming*, Oper. Res. Lett. 42 (2014), 522–526.
- [10] Marshall, A.W. and Olkin, I. *Inequalities: Theory of majorization and its applications*, Academic Press, New York, 1979.
- [11] Mut, M. and Wiecek, M.M. *Generalized equitable preference in multiobjective programming*, Eur. J. Oper. Res. 212 (2011), 535–551.
- [12] Ogryczak, W. *Inequality measures and equitable approaches to location problems*, Eur. J. Oper. Res. 122(2) (2000), 374–391.
- [13] Ogryczak, W. *Multiple criteria linear programming model for portfolio selection*, Ann. Oper. Res. 97 (2000), 143–162.

- [14] Ogryczak, W., Luss, H., Pioro, M. and Nace, D. *A. Tomaszewski, Fair optimization and networks: a survey*, J. Appl. Math. 2014 (2014), Article ID 612018, 25 pages.
- [15] Ogryczak, W., Wierzbicki, A. and Milewski, M. *A multi-criteria approach to fair and efficient bandwidth allocation*, Omega, 36 (2008), 451–463.
- [16] Ogryczak, W. and Zawadzki, M. *Conditional median: a parametric solution concept for location problems*, Ann. Oper. Res. 110 (2002), 167–181.

**How to cite this article**

Ahmadi, F., Salajegheh, A. R. and Foroutannia, D., Generalization of equitable efficiency in multiobjective optimization problems by the direct sum of matrices. *Iran. j. numer. anal. optim.*, 2023; 13(1): 80–101. <https://doi.org/10.22067/ijnao.2022.69731.1023>.



# A family of eight-order interval methods for computing rigorous bounds to the solution to nonlinear equations

M. Dehghani-Madiseh

## Abstract

One of the major problems in applied mathematics and engineering sciences is solving nonlinear equations. In this paper, a family of eight-order interval methods for computing rigorous bounds on the simple zeros of nonlinear equations is presented. We present the convergence and error analysis of the introduced methods. Also, the introduced methods are compared with the well-known interval Newton method and interval Ostrowski-type methods. Finally, we propose a technique based on the combination of the newly introduced approach with the extended interval arithmetic to find all of the roots of a nonlinear equation that are located in an initial interval.

**AMS subject classifications (2020):** 65G30; 34G20

**Keywords:** Interval arithmetic, Nonlinear equations, Rigorous bounds, Convergence analysis.

## 1 Introduction

The main motivation for this study is to enclose the simple root  $x^*$  of the nonlinear equation

$$f(x) = 0, \quad (1)$$

by a bounded interval, where  $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$  is a real-valued nonlinear function on the open interval  $D$ .

Nonlinear problems are of interest to engineers, physicists, and many other scientists because most systems are inherently nonlinear in nature.

---

Received 9 January 2022; revised 18 May 2022; accepted 19 May 2022

Marzieh Dehghani-Madiseh

Department of Mathematics, Faculty of Mathematical Sciences and Computer, Shahid Chamran University of Ahvaz, Ahvaz, Iran. e-mail: m.dehghani@scu.ac.ir

Up to now, many modified methods for solving nonlinear equations have been developed to improve the local order of convergence of some classical methods, such as Newton, Chebyshev, Potra-Ptak, and Ostrowski methods; see [19, 18, 7, 8, 6, 3, 4, 9, 10, 14, 2, 23, 5, 13].

An optimal eight-order method for solving nonlinear equation (1) proposed by Bi, Ren, and Wu [2] that is based on King family [14], is given by

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n = y_n - \frac{2f(x_n) - f(y_n)}{2f(x_n) - 5f(y_n)} \frac{f(y_n)}{f'(x_n)}, \\ x_{n+1} = z_n - h(\mu_n) \frac{f(z_n)}{f'(z_n)}, \end{cases} \quad (2)$$

where  $\mu_n = \frac{f(z_n)}{f(x_n)}$  and  $h$  is a real-valued function with  $h(0) = 1$ ,  $h'(0) = 2$  and  $|h''(0)| < \infty$ . Iterative method (2) with eight-order of convergence is very fast compared with many other methods. Solving the problems in floating-point arithmetic is inevitably associated with round-off errors, and so the obtained solution to the problem is accompanied by some errors. Interval analysis is a tool for bounding the errors and providing rigorous bounds on the solution to the problems. The interval extension of the Newton method with quadratic convergence [24, 16], the interval extensions of the Ostrowski method and modified Ostrowski method, respectively, with fourth-order and sixth-order of convergence [1, 11], and the interval extension of the  $n$ -step Traub method with  $(n + 1)$ -order of convergence [21], are examples of the interval methods that give rigorous bounds on the solution to the nonlinear equations.

In this work, we present an interval extension of (2), which has an eight-order of convergence and gives rigorous and outstanding results, that is, interval enclosures with sharp bounds that contain the exact solution. Also, we introduce a technique based on combining the new method and the extended interval arithmetic for enclosing all simple roots that are located in an initial interval. In contrast, many root-finding methods can only find one root of the function in the given initial interval.

Here, we use boldface letters to denote intervals. The set of real intervals is denoted by  $\mathbb{IR} = \{\mathbf{x} = [\underline{\mathbf{x}}, \bar{\mathbf{x}}] : \underline{\mathbf{x}} \leq \bar{\mathbf{x}}\}$ . The midpoint and width of an interval number  $\mathbf{x} = [\underline{\mathbf{x}}, \bar{\mathbf{x}}]$  are defined by  $m(\mathbf{x}) = \frac{\bar{\mathbf{x}} + \underline{\mathbf{x}}}{2}$  and  $w(\mathbf{x}) = \bar{\mathbf{x}} - \underline{\mathbf{x}}$ , respectively. The absolute value of  $\mathbf{x}$  is  $|\mathbf{x}| = \max\{|x| : x \in \mathbf{x}\}$ . The interval extension of real-valued function  $g$  is denoted by its corresponding uppercase and bold letter  $\mathbf{G}$ .

## 2 Description of the methods

Many modified methods for solving nonlinear equation (1) with a high-order of convergence are based on the well-known Newton method. So, we first give a brief description of the interval Newton method.

## 2.1 Interval Newton method

The idea of the interval Newton method for the first time was discussed in [24, 16]. Suppose that the real differentiable function  $f$  in (1) has the inclusion of monotonic interval extension  $\mathbf{F}'(\mathbf{x})$  of its derivative  $f'(x)$  and that  $\mathbf{x}_0$  is an initial point. Then the interval Newton method is

$$\mathbf{x}_{n+1} = \left\{ m(\mathbf{x}_n) - \frac{f(m(\mathbf{x}_n))}{\mathbf{F}'(\mathbf{x}_n)} \right\} \cap \mathbf{x}_n, \quad n = 0, 1, \dots \quad (3)$$

Recursive relation (3) produces a sequence  $\{\mathbf{x}_n\}$  of interval numbers. If the initial interval  $\mathbf{x}_0$  contains a zero  $x^*$  of  $f(x)$  and  $0 \notin \mathbf{F}'(\mathbf{x}_0)$ , then all iterates contain  $x^*$  and the method converges to  $x^*$ .

**Theorem 1.** [17] Let  $f$  be a real rational function of a single real variable  $x$  with rational extensions  $\mathbf{F}$  and  $\mathbf{F}'$  of  $f$  and  $f'$ , respectively, such that  $f$  has a simple zero  $y$  in an interval  $[x_1, x_2]$  for which  $F([x_1, x_2])$  is defined and  $\mathbf{F}'([x_1, x_2])$  is defined and does not contain zero. Then there is an interval  $\mathbf{x}_0 \subseteq [x_1, x_2]$  containing  $y$  and a positive real number  $K$  such that

$$w(\mathbf{x}_{n+1}) \leq K(w(\mathbf{x}_n))^2,$$

therein  $\{\mathbf{x}_n\}$  is the produced interval sequence by (3).

## 2.2 Main results and convergence analysis

In this subsection, a new interval method is introduced to obtain sharp enclosures for the simple zeros of nonlinear equations. First, for theoretical considerations, we present the following lemmas.

**Lemma 1.** [17, 1] For real numbers  $a$  and  $b$  and interval numbers  $\mathbf{x}$  and  $\mathbf{y}$ , we have

- (i)  $w(a\mathbf{x} + b\mathbf{y}) = |a|w(\mathbf{x}) + |b|w(\mathbf{y})$ ,
- (ii)  $w(\mathbf{x}\mathbf{y}) \leq |\mathbf{x}|w(\mathbf{y}) + |\mathbf{y}|w(\mathbf{x})$ .

**Lemma 2.** [17] Every nested sequence  $\{\mathbf{x}_k\}$  converges and has the limit  $\bigcap_{k=1}^{\infty} \mathbf{x}_k$ .

**Lemma 3.** [17] If  $\mathbf{F}$  is a natural interval extension of the real-valued rational function  $f$  with  $\mathbf{F}(\mathbf{x})$  defined for  $\mathbf{x} \subseteq \mathbf{x}_0$ , where  $\mathbf{x}$  and  $\mathbf{x}_0$  are intervals, then there exists a constant  $L$  such that

$$w(\mathbf{F}(\mathbf{x})) \leq Lw(\mathbf{x}).$$

Now we introduce the interval extension of (2) as follows:

$$\begin{cases} \mathbf{y}_n = \mathbf{N}(\mathbf{x}_n) \cap \mathbf{x}_n, \\ \mathbf{z}_n = \mathbf{R}(\mathbf{x}_n, \mathbf{y}_n) \cap \mathbf{x}_n, \\ \mathbf{x}_{n+1} = \mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n) \cap \mathbf{x}_n, \end{cases} \quad (4)$$

where

$$\mathbf{N}(\mathbf{x}) = \mathbf{m}(\mathbf{x}) - \frac{f(\mathbf{m}(\mathbf{x}))}{\mathbf{F}'(\mathbf{x})}, \quad (5)$$

$$\mathbf{R}(\mathbf{x}, \mathbf{y}) = \mathbf{m}(\mathbf{y}) - \frac{2f(\mathbf{m}(\mathbf{x})) - f(\mathbf{m}(\mathbf{y}))}{2f(\mathbf{m}(\mathbf{x})) - 5f(\mathbf{m}(\mathbf{y}))} \frac{f(\mathbf{m}(\mathbf{y}))}{\mathbf{F}'(\mathbf{x})}, \quad (6)$$

$$\mathbf{S}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{m}(\mathbf{z}) - \mathbf{H}(\tilde{\mu}) \frac{f(\mathbf{m}(\mathbf{z}))}{\mathbf{F}'(\mathbf{z})}, \quad \tilde{\mu} = \frac{\mathbf{F}(\mathbf{z})}{f(\mathbf{m}(\mathbf{x}))}, \quad (7)$$

in which  $\mathbf{H}$  is the interval extension of the continuous rational function  $h$ .

Now we are ready to present the theoretical analysis of the proposed method (4).

**Theorem 2.** Assume that  $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$  is continuously differentiable and that  $0 \notin \mathbf{F}'(\mathbf{x}_0)$  for a given  $\mathbf{x}_0 \subseteq D$ . If  $\mathbf{x}_0$  contains a zero  $x^*$  of  $f(x)$ , then so do all  $\mathbf{x}_k$  for  $k = 1, 2, \dots$ , defined by (4). Furthermore, the intervals  $\mathbf{x}_k$  form a nested sequence converging to  $x^*$ .

*Proof.* Using the Taylor expansion around  $x \in \mathbf{x}_0$ , we have

$$0 = f(x^*) = f(x) + (x^* - x)f'(\xi_1),$$

for some  $\xi_1$  between  $x$  and  $x^*$ . Because  $f'(\xi_1) \neq 0$ , we obtain

$$x^* = x - \frac{f(x)}{f'(\xi_1)}, \quad (8)$$

which  $f'(\xi_1) \in \mathbf{F}'(\mathbf{x}_0)$  yields

$$x^* = x - \frac{f(x)}{f'(\xi_1)} \in x - \frac{f(x)}{\mathbf{F}'(\mathbf{x}_0)}.$$

Since  $x \in \mathbf{x}_0$  is arbitrary, so in particular for  $x = \mathbf{m}(\mathbf{x}_0)$ , and taking into account that  $x^* \in \mathbf{x}_0$ , we obtain

$$x^* \in \left\{ \mathbf{m}(\mathbf{x}_0) - \frac{f(\mathbf{m}(\mathbf{x}_0))}{\mathbf{F}'(\mathbf{x}_0)} \right\} \cap \mathbf{x}_0 = \mathbf{N}(\mathbf{x}_0) \cap \mathbf{x}_0 = \mathbf{y}_0.$$

Now again using the Taylor theorem, for  $y \in \mathbf{y}_0$ , we can write

$$f(y) = f(x^*) + (y - x^*)f'(\xi_2),$$

for some  $\xi_2$  between  $y$  and  $x^*$ . Since  $f'(\xi_2) \neq 0$  and taking into account that  $f(x^*) = 0$ , we get

$$x^* = y - \frac{f(y)}{f'(\xi_2)}. \quad (9)$$

As previously mentioned, method (2) is based on the King family. King [14] proposed the following formula for approximating  $f'(y_n)$ :

$$f'(y_n) \approx f'(x_n) \frac{f(x_n) + \gamma f(y_n)}{f(x_n) + \beta f(y_n)}, \quad (10)$$

with  $\gamma = \beta - 2$  to achieve a fourth-order of convergence. Let  $\xi_1$  and  $\xi_2$  be sufficiently close to  $x$  and  $y$ , respectively. Whereas in method (2),  $\beta = -\frac{1}{2}$ , and using (10), we have

$$f'(\xi_2) \approx f'(\xi_1) \frac{2f(x) - 5f(y)}{2f(x) - f(y)}. \quad (11)$$

Substituting (11) into (9) yields

$$x^* = y - \frac{f(y)}{f'(\xi_2)} = y - \frac{2f(x) - f(y)}{2f(x) - 5f(y)} \frac{f(y)}{f'(\xi_1)}. \quad (12)$$

Indeed  $f'(\xi_1) \in \mathbf{F}'(\mathbf{x}_0)$  and (12) holds for any  $x \in \mathbf{x}_0$  and  $y \in \mathbf{y}_0$ , in particular for  $x = m(\mathbf{x}_0)$  and  $y = m(\mathbf{y}_0)$ . So, we obtain

$$x^* \in \left\{ m(\mathbf{y}_0) - \frac{2f(m(\mathbf{x}_0)) - f(m(\mathbf{y}_0))}{2f(m(\mathbf{x}_0)) - 5f(m(\mathbf{y}_0))} \frac{f(m(\mathbf{y}_0))}{\mathbf{F}'(\mathbf{x}_0)} \right\} \cap \mathbf{x}_0 = \mathbf{R}(\mathbf{x}_0, \mathbf{y}_0) \cap \mathbf{x}_0 = \mathbf{z}_0.$$

Now for  $z \in \mathbf{z}_0$ , by the Taylor theorem, we have

$$f(z) = f(x^*) + (z - x^*)f'(\xi_3), \quad (13)$$

for some  $\xi_3$  between  $z$  and  $x^*$ . Using the Taylor expansion for  $h(\mu)$  around zero with  $\mu = \frac{f(z)}{f(x)}$ , we get

$$h(\mu) \approx h(0) + \mu h'(0).$$

Since  $h(0) = 1$  and  $h'(0) = 2$ , we obtain

$$h(\mu) \approx 1 + 2 \frac{f(z)}{f(x)},$$

and so

$$f(z)h(\mu) = f(z) + 2 \frac{f^2(z)}{f(x)}.$$

Because  $z$  is arbitrary, we can assume that  $z$  and  $x^*$  are sufficiently close together and so  $f(z)h(\mu) \approx f(z)$ . Now using (13), we obtain

$$x^* = z - h(\mu) \frac{f(z)}{f'(\xi_3)}. \quad (14)$$

Indeed  $f'(\xi_3) \in \mathbf{F}'(\mathbf{z}_0)$  and (14) holds for any  $x \in \mathbf{x}_0$  and  $z \in \mathbf{z}_0$ , in particular for  $x = m(\mathbf{x}_0)$  and  $z = m(\mathbf{z}_0)$ . Therefore, since  $x^* \in \mathbf{x}_0$ , we obtain

$$x^* \in \left\{ m(\mathbf{z}_0) - \mathbf{H}(\tilde{\mu}_0) \frac{f(m(\mathbf{z}_0))}{\mathbf{F}'(\mathbf{z}_0)} \right\} \cap \mathbf{x}_0 = \mathbf{S}(\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0) \cap \mathbf{x}_0 = \mathbf{x}_1.$$

By continuing this process, we see that

$$x^* \in \mathbf{x}_k, \quad k = 0, 1, \dots \quad (15)$$

Now by formula (4), it is obvious that  $\mathbf{x}_{k+1} \subseteq \mathbf{x}_k$  for  $k = 0, 1, \dots$ , which means that  $\{\mathbf{x}_k\}$  is a nested sequence. By Lemma 2, this sequence is convergent to  $\mathbf{a} = \bigcap_{k=1}^{\infty} \mathbf{x}_k$ . Since  $x^* \in \mathbf{x}_k$  for all  $k$ , then  $x^* \in \mathbf{a}$ . On the other hand,  $m(\mathbf{z}_n)$  is not contained in  $\mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n)$  unless  $f(m(\mathbf{z}_n)) = 0$ . Since  $m(\mathbf{z}_n) \in \mathbf{z}_n \subseteq \mathbf{x}_n$ , we conclude that  $w(\mathbf{x}_{n+1}) < w(\mathbf{x}_n)$ . Therefore  $\mathbf{a} = x^*$ .  $\square$

Note that procedure (4) stops when some stopping criteria are fulfilled, such as  $w(\mathbf{x}_n) < \epsilon$  for a tolerance  $\epsilon$  or  $\mathbf{x}_{n+1} = \mathbf{x}_n$ . The computational scheme of the proposed interval method (4) for enclosing the simple roots of a given nonlinear equation  $f(x) = 0$  is presented in Algorithm 1.

---

**Algorithm 1** The new interval method (4) for enclosing roots of nonlinear equation  $f(x) = 0$

---

```

1: procedure INTERVAL ROOT-FINDING( $f, \mathbf{x}_0, tol$ )
2:    $n = 0$ ;
3:   while  $w(\mathbf{x}_n) \geq tol$  do
4:     Compute  $\mathbf{N}(\mathbf{x}_n)$  from (5);
5:      $\mathbf{y}_n = \text{intersect}(\mathbf{N}(\mathbf{x}_n), \mathbf{x}_n)$ ;
6:     Compute  $\mathbf{R}(\mathbf{x}_n, \mathbf{y}_n)$  from (6);
7:      $\mathbf{z}_n = \text{intersect}(\mathbf{R}(\mathbf{x}_n, \mathbf{y}_n), \mathbf{x}_n)$ ;
8:     Compute  $\mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n)$  from (7);
9:      $\mathbf{x}_{n+1} = \text{intersect}(\mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n), \mathbf{x}_n)$ ;
10:     $n = n + 1$ ;
11:   end while
12:   return  $\mathbf{x}_n$ 
13: end procedure

```

---

**Theorem 3.** Suppose that  $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$  is continuously differentiable and that  $0 \notin \mathbf{F}'(\mathbf{x}_0)$  for a given  $\mathbf{x}_0 \subseteq D$ . If  $x^* \in \mathbf{x}_0$ , then  $\mathbf{x}_k$  contains a unique root of  $f(x)$ , for  $k = 0, 1, \dots$ . Furthermore, if  $\mathbf{S}(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k) \cap \mathbf{x}_k = \emptyset$  for some  $k$ , then  $f(x) \neq 0$  for all  $x \in \mathbf{x}_0$ .



*Proof.* Let  $x^* \in \mathbf{x}_0$ . By Theorem 2, we conclude that  $x^* \in \mathbf{x}_k$  for all  $k$ , which is unique because  $0 \notin \mathbf{F}'(\mathbf{x}_k) \subseteq \mathbf{F}'(\mathbf{x}_0)$ .

Now suppose  $\mathbf{S}(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k) \cap \mathbf{x}_k = \emptyset$  for some  $k$ , but  $x^* \in \mathbf{x}_0$  is a root of  $f(x)$ , so by Theorem 2, we conclude that  $x^* \in \mathbf{x}_n$  for all  $n$ . Particularly  $x^* \in \mathbf{x}_{k+1} = \mathbf{S}(\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k) \cap \mathbf{x}_k$ , which is a contradiction.  $\square$

Theorem 3 is in the category of verification methods. By verifying its assumptions with the aid of a computer, we can detect when a certain interval does not contain a root.

**Theorem 4.** Let  $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$  be continuously differentiable and have a simple zero  $x^*$  in  $\mathbf{x}_0$ . If  $0 \notin \mathbf{F}'(\mathbf{x}_0)$ , then the interval method (4) has an eight-order of convergence, that is, there exists a positive real number  $C$  such that

$$w(\mathbf{x}_{n+1}) \leq C(w(\mathbf{x}_n))^8.$$

*Proof.* Since  $\mathbf{x}_{n+1} \subseteq \mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n)$  so  $w(\mathbf{x}_{n+1}) \leq w(\mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n))$ . By the mean value theorem, we can write

$$f(m(\mathbf{z}_n)) = f'(\eta_1)(m(\mathbf{z}_n) - x^*),$$

for some  $\eta_1$  between  $m(\mathbf{z}_n)$  and  $x^*$ . Thus we get

$$\mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n) = m(\mathbf{z}_n) - \mathbf{H}(\tilde{\mu}_n) \frac{f'(\eta_1)(m(\mathbf{z}_n) - x^*)}{\mathbf{F}'(\mathbf{z}_n)}. \quad (16)$$

Therefore, from (16) and Lemma 1, we obtain

$$w(\mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n)) \leq |\mathbf{H}(\tilde{\mu}_n)| |m(\mathbf{z}_n) - x^*| f'(\eta_1) |w(\frac{1}{\mathbf{F}'(\mathbf{z}_n)})| + w(\mathbf{H}(\tilde{\mu}_n)) |m(\mathbf{z}_n) - x^*| f'(\eta_1) |\frac{1}{\mathbf{F}'(\mathbf{z}_n)}|. \quad (17)$$

Because  $x^* \in \mathbf{z}_n$ , it is obvious that

$$|m(\mathbf{z}_n) - x^*| \leq w(\mathbf{z}_n). \quad (18)$$

On the other hand, we can write

$$\mathbf{z}_n \subseteq m(\mathbf{y}_n) - \frac{2f(m(\mathbf{x}_n)) - f(m(\mathbf{y}_n))}{2f(m(\mathbf{x}_n)) - 5f(m(\mathbf{y}_n))} \frac{f(m(\mathbf{y}_n))}{\mathbf{F}'(\mathbf{x}_n)}. \quad (19)$$

Using the mean value theorem, we have

$$f(m(\mathbf{y}_n)) = f'(\eta_2)(m(\mathbf{y}_n) - x^*), \quad \text{and} \quad f(m(\mathbf{x}_n)) = f'(\eta_3)(m(\mathbf{x}_n) - x^*), \quad (20)$$

for some  $\eta_2$  between  $m(\mathbf{y}_n)$  and  $x^*$  and some  $\eta_3$  between  $m(\mathbf{x}_n)$  and  $x^*$ . Because

$$|m(\mathbf{y}_n) - x^*| \leq w(\mathbf{y}_n) \leq w(\mathbf{x}_n) \quad \text{and} \quad |m(\mathbf{x}_n) - x^*| \leq w(\mathbf{x}_n), \quad (21)$$

so using (20) and (21), we obtain

$$|f(m(y_n))| \leq |f'(\eta_2)|w(y_n), \quad (22)$$

$$\begin{aligned} |2f(m(x_n)) - f(m(y_n))| &\leq 2|f(m(x_n))| + |f(m(y_n))| \\ &= 2|f'(\eta_3)||m(x_n) - x^*| + |f'(\eta_2)||m(y_n) - x^*| \\ &\leq 2|f'(\eta_3)|w(x_n) + |f'(\eta_2)|w(x_n) \leq C_1 w(x_n), \end{aligned} \quad (23)$$

where  $C_1$  is an upper bound for  $2|f'(\eta_3)| + |f'(\eta_2)|$ . On the other hand, Theorem 1 yields

$$w(y_n) \leq C_2(w(x_n))^2, \quad (24)$$

for a positive constant  $C_2$ . So by (22) and (24), we obtain

$$|f(m(y_n))| \leq C_3(w(x_n))^2, \quad (25)$$

in which  $C_3$  is an upper bound for  $C_2|f'(\eta_2)|$ . Using Lemma 3, we have

$$w\left(\frac{1}{F'(x_n)}\right) \leq C_4 w(x_n). \quad (26)$$

Now from (19) and Lemma 1, we can write

$$w(z_n) \leq \frac{|2f(m(x_n)) - f(m(y_n))|}{|2f(m(x_n)) - 5f(m(y_n))|} |f(m(y_n))| w\left(\frac{1}{F'(x_n)}\right),$$

Moreover, using (23), (25), and (26) yields

$$w(z_n) \leq C_5(w(x_n))^4, \quad (27)$$

where  $C_5$  is an upper bound for  $\frac{C_1 C_3 C_4}{|2f(m(x_n)) - 5f(m(y_n))|}$ . By Lemma 3 and (27), we get

$$w\left(\frac{1}{F'(z_n)}\right) \leq w(z_n) \leq C_5(w(x_n))^4. \quad (28)$$

Therefore, from (18), (27) and (28), we have

$$|H(\tilde{\mu}_n)||m(z_n) - x^*| |f'(\eta_1)| w\left(\frac{1}{F'(z_n)}\right) \leq C_6(w(x_n))^8, \quad (29)$$

in which  $C_6$  is an upper bound for  $C_5^2 |f'(\eta_1)| |H(\tilde{\mu}_n)|$ . Now by Lemma 3, there exists a positive constant  $C_7$  such that

$$w(H(\tilde{\mu}_n)) \leq C_7 w(\tilde{\mu}_n). \quad (30)$$

Using Lemmas 1 and 3 and (27), we obtain

$$w(\tilde{\mu}_n) = w\left(\frac{\mathbf{F}(\mathbf{z}_n)}{f(\mathbf{m}(\mathbf{x}_n))}\right) = \frac{w(\mathbf{F}(\mathbf{z}_n))}{|f(\mathbf{m}(\mathbf{x}_n))|} \leq \frac{C_8 w(\mathbf{z}_n)}{|f(\mathbf{m}(\mathbf{x}_n))|} \leq C_9 (w(\mathbf{x}_n))^4, \quad (31)$$

where  $C_8$  is a positive constant and  $C_9$  is an upper bound for  $\frac{C_5 C_8}{|f(\mathbf{m}(\mathbf{x}_n))|}$ . From (30) and (31), we can write

$$w(\mathbf{H}(\tilde{\mu}_n)) \leq C_{10} (w(\mathbf{x}_n))^4, \quad (32)$$

in which  $C_{10} = C_7 C_9$ . Using (18), (27), and (32), we obtain

$$w(\mathbf{H}(\tilde{\mu}_n)) |\mathbf{m}(\mathbf{z}_n) - x^*| |f'(\eta_1)| \left| \frac{1}{\mathbf{F}'(\mathbf{z}_n)} \right| \leq C_{11} (w(\mathbf{x}_n))^8, \quad (33)$$

where  $C_{11}$  is an upper bound for  $C_5 C_{10} |f'(\eta_1)| \left| \frac{1}{\mathbf{F}'(\mathbf{z}_n)} \right|$ . Finally, since  $w(\mathbf{x}_{n+1}) \leq w(\mathbf{S}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n))$ , by (17), (29), and (33), we conclude that  $w(\mathbf{x}_{n+1}) \leq C (w(\mathbf{x}_n))^8$ , where  $C = C_6 + C_{11}$ .  $\square$

As one can see, the new interval method (4) with three-step has an eight-order of convergence, while some other interval methods with the same number of steps have a lower order of convergence; for some of them, see [1, 21].

### 3 Test problems

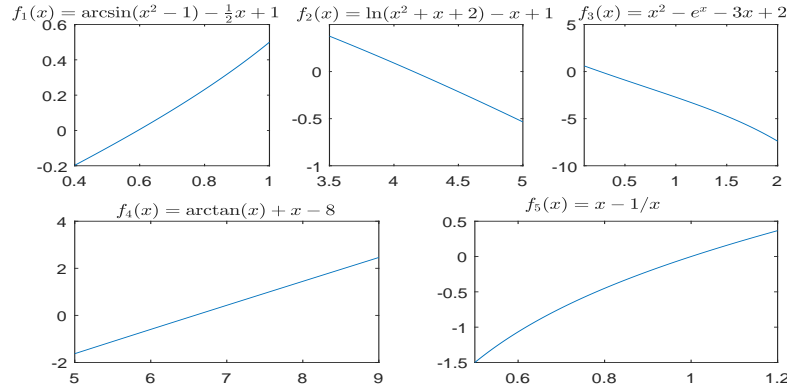
In this section, we give some numerical examples to illustrate the performance of the new approach proposed in Section 2. The new method is compared with the interval Newton method, interval Ostrowski method, and interval modified Ostrowski method. In all examples, the procedures are stopped when  $w(\mathbf{x}_k) < 10^{-16}$ . We utilize INTLAB [22] to compute the verified results on the computer. We study the following examples:

$$\begin{aligned} f_1(x) &= \arcsin(x^2 - 1) - \frac{1}{2}x + 1, & x_1^* &\approx 0.5948109683983692, \\ f_2(x) &= \ln(x^2 + x + 2) - x + 1, & x_2^* &\approx 4.1525907367571583, \\ f_3(x) &= x^2 - e^x - 3x + 2, & x_3^* &\approx 0.25753028543986079, \\ f_4(x) &= \arctan(x) + x - 8, & x_4^* &\approx 6.58002470991429699, \\ f_5(x) &= x - 1/x, & x_5^* &= 1. \end{aligned}$$

The first two examples are taken from [2] and the latest is taken from [15]. For all examples, we use rational function  $h$  as follows:

$$h(t) = 1 + \frac{2t}{1+t}.$$

In Figure 1, one can see the graphs of five functions  $f_1, f_2, f_3, f_4$ , and  $f_5$ , respectively, over the initial intervals  $\mathbf{x}_0^1 = [0.4, 1]$ ,  $\mathbf{x}_0^2 = [3.5, 5]$ ,  $\mathbf{x}_0^3 = [0.1, 2]$ ,

Figure 1: Graphs of functions  $f_1, f_2, f_3, f_4, f_5$ 

$\mathbf{x}_0^4 = [5, 9]$ , and  $\mathbf{x}_0^5 = [0.5, 1.2]$ . By this figure, in addition to obtaining an intuitive view of the functions, we can understand the behavior of the functions for computing the following parameter:

$$\rho_k = \max_{x \in \mathbf{x}_k} |f(x)|. \quad (34)$$

In the tables below, one can see the results obtained by implementing the interval Newton method, the interval Ostrowski method, the interval modified Ostrowski method, and the new method (4) introduced in this paper. The third and fourth columns of the tables show, respectively, the tolerance parameters  $\delta_k = \frac{w(\mathbf{x}_k)}{\max\{|\mathbf{x}_k|, 1\}}$  and  $\rho_k$  introduced by (34). Note that in some tables, mark "—" in the last step shows that the method fails in solving the problem.

As the first example for  $f_1(x) = \arcsin(x^2 - 1) - \frac{1}{2}x + 1$ , we present the obtained results by the mentioned methods in Tables 1–4. The presented results in these tables show that the new method (4) achieves the desired result with less number of iterations and higher accuracy. Also, the interval Ostrowski method fails in solving the problem.

Tables 5–8 show the results obtained by executing different methods for enclosing the root of  $f_2(x) = \ln(x^2 + x + 2) - x + 1$ . It can be seen that the new method (4) successes in the least number of iterations. Also, the interval Ostrowski method fails in solving the problem.

For the third function  $f_3(x) = x^2 - e^x - 3x + 2$ , the reported values in Tables 9–12 show that only the new method (4) successes in getting the result and the other methods fail.

Tables 13–16 display the results obtained by executing four methods for enclosing the root of  $f_4(x) = \arctan(x) + x - 8$ . As one can see, the interval Ostrowski method has failed to obtain a result, and the new approach gives better results than the other methods.

The results of different methods for obtaining appropriate enclosures for the positive root of  $f_5(x) = x - 1/x$  have been displayed in Tables 17–20. The interval Newton method does not yield any result. Whereas the new approach yields the exact root of the function.

Table 1: Results of the interval Newton method for  $f_1(x) = \arcsin(x^2 - 1) - \frac{1}{2}x + 1$

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [0.40000000000000002, 0.66396313641487115] | $2.64 \times 10^{-1}$  | $7.47 \times 10^{-2}$  |
| 2   | [0.56560254826011236, 0.66396313641487115] | $9.84 \times 10^{-2}$  | $7.47 \times 10^{-2}$  |
| 3   | [0.59018815218397114, 0.59856980551945871] | $8.38 \times 10^{-3}$  | $3.98 \times 10^{-3}$  |
| 4   | [0.59480310218157917, 0.59481912020532601] | $1.60 \times 10^{-5}$  | $8.63 \times 10^{-6}$  |
| 5   | [0.59481096839332148, 0.59481096840342751] | $1.01 \times 10^{-11}$ | $5.36 \times 10^{-12}$ |
| 6   | [0.59481096839836900, 0.59481096839836911] | $2.22 \times 10^{-16}$ | 0                      |
| 7   | [0.59481096839836911, 0.59481096839836911] | 0                      | 0                      |

Table 2: Results of the interval Ostrowski method for  $f_1(x) = \arcsin(x^2 - 1) - \frac{1}{2}x + 1$

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [0.54158214865149934, 0.63394129754193074] | $9.24 \times 10^{-2}$  | $4.19 \times 10^{-2}$  |
| 2   | [0.59477478728793232, 0.59485799844400755] | $8.32 \times 10^{-5}$  | $4.98 \times 10^{-5}$  |
| 3   | [0.59481096839836756, 0.59481096839837055] | $3.11 \times 10^{-15}$ | $1.33 \times 10^{-15}$ |
| 4   | —                                          | —                      | —                      |

Table 3: Results of the interval modified Ostrowski method for  $f_1(x) = \arcsin(x^2 - 1) - \frac{1}{2}x + 1$

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [0.58885410911304559, 0.59936304066316770] | $1.05 \times 10^{-2}$  | $4.83 \times 10^{-3}$  |
| 2   | [0.59481096839549719, 0.59481096840132608] | $5.83 \times 10^{-12}$ | $3.13 \times 10^{-12}$ |
| 3   | [0.59481096839836922, 0.59481096839836933] | $2.22 \times 10^{-16}$ | $1.11 \times 10^{-16}$ |
| 4   | [0.59481096839836933, 0.59481096839836933] | 0                      | $1.11 \times 10^{-16}$ |

Table 4: Results of the new method (4) for  $f_1(x) = \arcsin(x^2 - 1) - \frac{1}{2}x + 1$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [0.58015286826057066, 0.60890961953980971] | $2.88 \times 10^{-2}$  | $1.50 \times 10^{-2}$  |
| 2   | [0.59481096839720404, 0.59481096839958292] | $2.38 \times 10^{-12}$ | $1.29 \times 10^{-12}$ |
| 3   | [0.59481096839836911, 0.59481096839836911] | 0                      | 0                      |

Table 5: Results of the interval Newton method for  $f_2(x) = \ln(x^2 + x + 2) - x + 1$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [4.09482718955130400, 4.17132082850488750] | $1.83 \times 10^{-2}$  | $3.47 \times 10^{-2}$  |
| 2   | [4.15231696283340760, 4.15292802943720400] | $1.47 \times 10^{-4}$  | $2.03 \times 10^{-4}$  |
| 3   | [4.15259073289156170, 4.15259074074274000] | $1.90 \times 10^{-9}$  | $2.40 \times 10^{-9}$  |
| 4   | [4.15259073675715750, 4.15259073675715840] | $4.28 \times 10^{-16}$ | $4.44 \times 10^{-16}$ |
| 5   | [4.15259073675715840, 4.15259073675715840] | 0                      | 0                      |

Table 6: Results of the interval Ostrowski method for  $f_2(x) = \ln(x^2 + x + 2) - x + 1$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [4.14427225093898070, 4.15515943057456380] | $2.62 \times 10^{-3}$  | $5.01 \times 10^{-3}$  |
| 2   | [4.15259073560489430, 4.15259073791874480] | $5.57 \times 10^{-10}$ | $6.10 \times 10^{-10}$ |
| 3   | [4.15259073675715840, 4.15259073675715840] | 0                      | 0                      |

Table 7: Results of the interval modified Ostrowski method for  $f_2(x) = \ln(x^2 + x + 2) - x + 1$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [4.15136705154255560, 4.15297239536206850] | $3.87 \times 10^{-4}$  | $7.37 \times 10^{-4}$  |
| 2   | [4.15259073675715750, 4.15259073675715840] | $4.28 \times 10^{-16}$ | $4.44 \times 10^{-16}$ |
| 3   | —                                          | —                      | —                      |

Table 8: Results of the new method (4) for  $f_2(x) = \ln(x^2 + x + 2) - x + 1$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$            | $\rho_k$              |
|-----|--------------------------------------------|-----------------------|-----------------------|
| 1   | [4.15167922809522590, 4.15321948581378480] | $3.71 \times 10^{-4}$ | $5.49 \times 10^{-4}$ |
| 2   | [4.15259073675715840, 4.15259073675715840] | 0                     | 0                     |

Table 9: Results of the interval Newton method for  $f_3(x) = x^2 - e^x - 3x + 2$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [0.1000000000000001, 0.76487534371627797]  | $6.65 \times 10^{-1}$  | 1.86                   |
| 2   | [0.17953909948997981, 0.30082399330312792] | $1.21 \times 10^{-1}$  | $2.97 \times 10^{-1}$  |
| 3   | [0.25663052647850410, 0.25844642836458781] | $1.82 \times 10^{-3}$  | $3.46 \times 10^{-3}$  |
| 4   | [0.25753027894621072, 0.25753029191301735] | $1.30 \times 10^{-8}$  | $2.45 \times 10^{-8}$  |
| 5   | [0.25753028543986067, 0.25753028543986073] | $1.11 \times 10^{-16}$ | $4.44 \times 10^{-16}$ |
| 6   | —                                          | —                      | —                      |

Table 10: Results of the interval Ostrowski method for  $f_3(x) = x^2 - e^x - 3x + 2$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$              |
|-----|--------------------------------------------|------------------------|-----------------------|
| 1   | [0.1000000000000001, 0.31655239623745746]  | $2.17 \times 10^{-1}$  | $6.05 \times 10^{-1}$ |
| 2   | [0.25752321108442017, 0.25753842849505237] | $1.52 \times 10^{-5}$  | $3.08 \times 10^{-5}$ |
| 3   | [0.25753028543986073, 0.25753028543986078] | $1.11 \times 10^{-16}$ | 0                     |
| 4   | —                                          | —                      | —                     |

Table 11: Results of the interval modified Ostrowski method for  $f_3(x) = x^2 - e^x - 3x + 2$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$            | $\rho_k$              |
|-----|--------------------------------------------|-----------------------|-----------------------|
| 1   | [0.24154741311026207, 2.0000000000000000]  | $8.79 \times 10^{-1}$ | 7.39                  |
| 2   | [0.25749104640972659, 0.39675078835778121] | $1.39 \times 10^{-1}$ | $5.20 \times 10^{-1}$ |
| 3   | [0.25753043640242368, 0.25753384076872499] | $3.40 \times 10^{-6}$ | $1.34 \times 10^{-5}$ |
| 4   | —                                          | —                     | —                     |

Table 12: Results of the new method (4) for  $f_3(x) = x^2 - e^x - 3x + 2$ 

| $k$ | $\mathbf{x}_k$                             | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------|------------------------|------------------------|
| 1   | [0.22110828457567316, 0.27623770073133980] | $5.51 \times 10^{-2}$  | $1.38 \times 10^{-1}$  |
| 2   | [0.25753028543982470, 0.25753028543989787] | $7.32 \times 10^{-14}$ | $1.40 \times 10^{-13}$ |
| 3   | [0.25753028543986078, 0.25753028543986078] | 0                      | 0                      |

Table 13: Results of the interval Newton method for  $f_4(x) = \arctan(x) + x - 8$ 

| $k$ | $\mathbf{x}_k$                                   | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------------|------------------------|------------------------|
| 1   | $[6.5762681889199976482, 6.5869858860385530619]$ | $1.62 \times 10^{-3}$  | $7.11 \times 10^{-3}$  |
| 2   | $[6.5800246452848929479, 6.5800247578416417582]$ | $1.71 \times 10^{-8}$  | $6.60 \times 10^{-8}$  |
| 3   | $[6.5800247099142961105, 6.5800247099142978868]$ | $2.69 \times 10^{-16}$ | $1.77 \times 10^{-15}$ |
| 4   | $[6.5800247099142969986, 6.5800247099142969986]$ | 0                      | 0                      |

Table 14: Results of the interval Ostrowski method for  $f_4(x) = \arctan(x) + x - 8$ 

| $k$ | $\mathbf{x}_k$                                   | $\delta_k$             | $\rho_k$               |
|-----|--------------------------------------------------|------------------------|------------------------|
| 1   | $[6.5799958235806119689, 6.5800370828300822623]$ | $6.27 \times 10^{-6}$  | $2.95 \times 10^{-5}$  |
| 2   | $[6.5800247099142961105, 6.5800247099142969986]$ | $1.34 \times 10^{-16}$ | $8.88 \times 10^{-16}$ |
| 3   | —                                                | —                      | —                      |

Table 15: Results of the interval modified Ostrowski method for  $f_4(x) = \arctan(x) + x - 8$ 

| $k$ | $\mathbf{x}_k$                                   | $\delta_k$            | $\rho_k$              |
|-----|--------------------------------------------------|-----------------------|-----------------------|
| 1   | $[6.5800246462005800296, 6.5800248588084278012]$ | $3.23 \times 10^{-8}$ | $1.52 \times 10^{-7}$ |
| 2   | $[6.5800247099142969986, 6.5800247099142969986]$ | 0                     | 0                     |

Table 16: Results of the new method (4) for  $f_4(x) = \arctan(x) + x - 8$ 

| $k$ | $\mathbf{x}_k$                                   | $\delta_k$             | $\rho_k$              |
|-----|--------------------------------------------------|------------------------|-----------------------|
| 1   | $[6.5800247087713694683, 6.5800247104028359857]$ | $2.47 \times 10^{-10}$ | $1.16 \times 10^{-9}$ |
| 2   | $[6.5800247099142969986, 6.5800247099142969986]$ | 0                      | 0                     |

Table 17: Results of the interval Newton method for  $f_5(x) = x - 1/x$ 

| $k$ | $\mathbf{x}_k$ | $\delta_k$ | $\rho_k$ |
|-----|----------------|------------|----------|
| 1   | —              | —          | —        |



Table 18: Results of the interval Ostrowski method for  $f_5(x) = x - 1/x$ 

| $k$ | $\mathbf{x}_k$                                  | $\delta_k$             | $\rho_k$               |
|-----|-------------------------------------------------|------------------------|------------------------|
| 1   | [0.99046958119024919309, 1.0128785276723828446] | $2.21 \times 10^{-2}$  | $2.55 \times 10^{-2}$  |
| 2   | [0.9999999856310709534, 1.000000014532778092]   | $2.89 \times 10^{-8}$  | $2.90 \times 10^{-8}$  |
| 3   | [1.000000000000000000, 1.00000000000000222]     | $2.22 \times 10^{-16}$ | $3.33 \times 10^{-16}$ |
| 4   | [1.00000000000000222, 1.00000000000000222]      | 0                      | $3.33 \times 10^{-16}$ |

Table 19: Results of the interval modified Ostrowski method for  $f_5(x) = x - 1/x$ 

| $k$ | $\mathbf{x}_k$                                  | $\delta_k$             | $\rho_k$               |
|-----|-------------------------------------------------|------------------------|------------------------|
| 1   | [0.99900511706023975567, 1.0007695812111181421] | $1.76 \times 10^{-3}$  | $1.99 \times 10^{-3}$  |
| 2   | [1.000000000000000000, 1.00000000000000222]     | $2.22 \times 10^{-16}$ | $3.33 \times 10^{-16}$ |
| 3   | [1.00000000000000222, 1.00000000000000222]      | 0                      | $3.33 \times 10^{-16}$ |

Table 20: Results of the new method (4) for  $f_5(x) = x - 1/x$ 

| $k$ | $\mathbf{x}_k$                                  | $\delta_k$             | $\rho_k$               |
|-----|-------------------------------------------------|------------------------|------------------------|
| 1   | [0.99968995513425429333, 1.0004281041560696419] | $7.37 \times 10^{-4}$  | $8.56 \times 10^{-4}$  |
| 2   | [1.000000000000000000, 1.00000000000000222]     | $2.22 \times 10^{-16}$ | $3.33 \times 10^{-16}$ |
| 3   | 1                                               | 0                      | 0                      |

## 4 Enclosing the roots using extended interval arithmetic

In this section, we want to introduce a technique that can find all the roots of a nonlinear equation  $f(x) = 0$  located in a wide initial interval. Many root-finding methods in floating-point arithmetic can only find one root of the function in a given interval. Our technique is based on combining the new method (4) introduced in this paper and the extended interval arithmetic [12, 17].

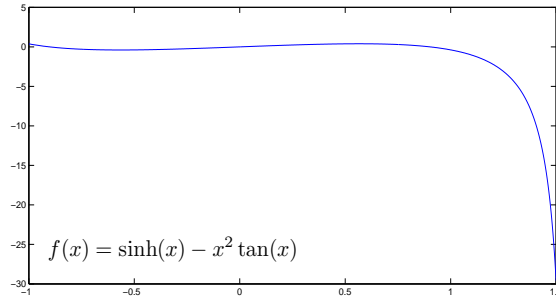
For a continuously differentiable function  $f(x)$ , if  $\mathbf{x}_0$  contains more than one zero of  $f(x)$ , then  $o \in \mathbf{F}'(\mathbf{x}_0)$ , and the discussed theorems in Section 2 will not be applicable. Using the extended interval arithmetic, this problem can be handled. As said in [17], the definition of interval division can be extended as follows:

$$[a, b]/[c, d] = [a, b](1/[c, d]),$$

where

$$1/[c, d] = \{1/y : y \in [c, d]\}.$$

If  $0 \notin [c, d]$ , then we are using the ordinary interval arithmetic. If  $0 \in [c, d]$ , leaving aside the case  $c = d = 0$ , then the extended interval arithmetic

Figure 2: Graph of function  $f(x) = \sinh(x) - x^2 \tan(x)$ 

specifies the following cases:

$$1/[c, d] = \begin{cases} [1/d, +\infty) & \text{if } c = 0 < d, \\ (-\infty, 1/c] \cup [1/d, +\infty) & \text{if } c < 0 < d, \\ (-\infty, 1/c] & \text{if } c < d = 0. \end{cases}$$

Now if the initial interval  $\mathbf{x}_0$  is such that  $0 \in \mathbf{F}'(\mathbf{x}_0)$ , then the quotient  $\frac{f(\text{mid}(\mathbf{x}_0))}{\mathbf{F}'(\mathbf{x}_0)}$  in (5) splits into two unbounded intervals. Thereafter intersecting  $\mathbf{N}(\mathbf{x}_0)$  with the finite interval  $\mathbf{x}_0$  yields two disjoint intervals  $\mathbf{y}_{11}$  and  $\mathbf{y}_{12}$ . First, for  $\mathbf{y}_{11}$ , if  $0 \notin \mathbf{F}'(\mathbf{y}_{11})$ , then we take  $\mathbf{y}_{11}$  as the initial point for the new method (4), otherwise again by computing  $\mathbf{N}(\mathbf{y}_{11})$  and then intersecting it with  $\mathbf{y}_{11}$ , we obtain two other intervals. By repeating this process, we find some intervals that contain a simple zero of  $f(x)$  and  $\mathbf{F}'$  over them does not contain zero. The process for  $\mathbf{y}_{12}$  is similar. Considering these intervals as initial points for the new method (4), we find all roots of  $f(x)$  on the initial interval  $\mathbf{x}_0$ . A similar idea previously has been used for the interval Newton method; see [17].

For an example, we consider  $f(x) = \sinh(x) - x^2 \tan(x)$  on the initial interval  $\mathbf{x}_0 = [-1, 1.5]$ . The graph of this function on  $\mathbf{x}_0 = [-1, 1.5]$  is shown in Figure 2. We have  $0 \in \mathbf{F}'(\mathbf{x}_0) = 10^2[-4.9097, 0.3056]$ . Using the extended interval arithmetic, we obtain

$$\begin{aligned} \mathbf{N}(\mathbf{x}_0) &= m(\mathbf{x}_0) - \frac{f(\text{mid}(\mathbf{x}_0))}{\mathbf{F}'(\mathbf{x}_0)} \\ &= (-\infty, 0.24225490053166] \cup [0.25048201511344, +\infty). \end{aligned}$$

Intersecting  $\mathbf{N}(\mathbf{x}_0)$  with  $\mathbf{x}_0$ , we get

$$\mathbf{y}_1 = [-1, 0.24225490053166] \cup [0.25048201511344, 1.5].$$

Indeed  $\mathbf{F}'([-1, 0.24225490053166])$  and  $\mathbf{F}'([0.25048201511344, 1.5])$  contain zero, too. We repeat the above process by putting  $\mathbf{x}_0 = [-1, 0.24225490053166]$  and  $\mathbf{x}_0 = [0.25048201511344, 1.5]$ , separately. Doing this work several times, we obtain three appropriate intervals, and then we apply the new method (4) on these intervals. The obtained results are shown in Table 21.

As one can see, in a few iterations, all three roots of  $f(x) = \sinh(x) - x^2 \tan(x)$  in  $\mathbf{x}_0 = [-1, 1.5]$  have been enclosed with sharp bounds and high accuracy.

Table 21: Results of the new technique in Section 4 for  $f(x) = \sinh(x) - x^2 \tan(x)$

| $k$ | $\mathbf{x}_k$                                                  | $\delta_k$             | $\rho_k$               |
|-----|-----------------------------------------------------------------|------------------------|------------------------|
| 1   | $\mathbf{x}_{11}=[0.87539095698495750, 0.93264721412667118]$    | $5.73 \times 10^{-2}$  | $9.89 \times 10^{-2}$  |
| 2   | $\mathbf{x}_{12}=[0.90196399818943263, 0.90196401144268923]$    | $1.33 \times 10^{-8}$  | $2.08 \times 10^{-8}$  |
| 3   | $\mathbf{x}_{13}=[0.90196400520858955, 0.90196400520858966]$    | $2.22 \times 10^{-16}$ | $4.44 \times 10^{-16}$ |
| 1   | $\mathbf{x}_{21}=[-0.00000014204496225, 0.00000008340020063]$   | $2.25 \times 10^{-7}$  | $1.42 \times 10^{-7}$  |
| 2   | $\mathbf{x}_{22}=10^{-50}[-0.20045735325692, 0.46773382426614]$ | $6.68 \times 10^{-51}$ | $4.68 \times 10^{-51}$ |
| 1   | $\mathbf{x}_{31}=[-0.90414914681585001, -0.90027520645356984]$  | $3.87 \times 10^{-3}$  | $6.51 \times 10^{-3}$  |
| 2   | $\mathbf{x}_{32}=[-0.90196400520858988, -0.90196400520858899]$  | $8.88 \times 10^{-16}$ | $1.33 \times 10^{-15}$ |

## 5 Concluding remarks

In this work, a new family of numerical methods for enclosing the simple roots of the nonlinear equations was proposed. We showed that the new methods have an eight-order of convergence and also that the convergence analysis of the methods was studied. Some numerical examples were presented to show the feasibility and effectiveness of the new method proposed in Section 2. Also, we proposed a technique based on combining the new method (4) with the extended interval arithmetic to find all the roots of a nonlinear equation located in an initial interval. Finally, a numerical example for testing this technique was presented.

## Acknowledgment

The author would like to thank the Shahid Chamran University of Ahvaz for financial support under the grant number SCU.MM1400.33518.

## References

- [1] Bakhtiari, P., Lotfi, T., Mahdiani, K. and Soleymani, F. *Interval Ostrowski-type methods with guaranteed convergence*, Ann. Univ. Fer-

- rara 59 (2013) 221–234.
- [2] Bi, W., Ren, H. and Wu, Q. *Three-step iterative methods with eighth-order convergence for solving nonlinear equations*, J. Comput. Appl. Math. 255 (2009) 105–112.
  - [3] Chun, C. and Neta, B. *A new sixth-order scheme for nonlinear equations*, Appl. Math. Lett. 25 (2012) 185–189.
  - [4] Chun, C. and Neta, B. *Comparative study of methods of various orders for finding repeated roots of nonlinear equations*, J. Comput. Appl. Math. 340 (2018) 11–42.
  - [5] Cordero, A., Hueso, J.L., Martínez, E. and Torregrosa, J.R. *New modifications of Potra-Pták's method with optimal fourth and eighth orders of convergence*, J. Comput. Appl. Math. 234(10) (2010) 2969–2976.
  - [6] Cordero, A., Jordan, C. and Torregrosa, J.R. *One-point Newton-type iterative methods: A unified point of view*, J. Comput. Appl. Math. 275 (2015) 366–374.
  - [7] Cordero, A. and Torregrosa, J.R. *A class of multi-point iterative methods for nonlinear equations*, Appl. Math. Comput. 197 (2008) 337–344.
  - [8] Cordero, A., Torregrosa, J.R. and Vassileva, M.P. *Increasing the order of convergence of iterative schemes for solving nonlinear systems*, J. Comput. Appl. Math. 252 (2013) 86–94.
  - [9] Dehghan, M. and Hajarian, M. *Some derivative free quadratic and cubic convergence iterative formulas for solving nonlinear equations*, Comput. Appl. Math. 29 (2010) 19–30.
  - [10] Dehghan, M. and Hajarian, M. *New iterative method for solving nonlinear equations with fourth-order convergence*, Int. J. Comput. Math. 87 (2010) 834–83.
  - [11] Eftekhari, T. *A new proof of interval extension of the classic Ostrowskis method and its modified method for computing the enclosure solutions of nonlinear equations*, Numer. Algorithms 69(1) (2015) 157–165.
  - [12] Kahan, W.M. *A more complete interval arithmetic* Lecture notes for an engineering summer course in numerical Analysis at the University of Michigan. Technical report, University of Michigan, 1968.
  - [13] Kearfott, R.B. *Interval computations: Introduction, uses, and resources*, Euromath Bull. 2(1) (1996) 95–112.
  - [14] King, R. *A family of fourth order methods for nonlinear equations*, SIAM J. Numer. Anal. 10 (1973) 876–879.

- [15] Lotfi, T. and Eftekhari, T. *A new optimal eighth-order Ostrowski-type family of iterative methods for solving nonlinear equations*, Chin. J. Math. (N.Y.) 2014, Art. ID 369713, 7 pp.
- [16] Moore, R.E. *Interval analysis*, Englewood Cliffs: Prentice-Hall, 1966.
- [17] Moore, R.E., Kearfott, R.B. and Cloud, M.J. *Introduction to interval analysis*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009.
- [18] Noor, M.A. and Khan, W.A. *New iterative methods for solving nonlinear equation by using homotopy perturbation method*, Appl. Math. Comput. 219 (2012) 3565–3574.
- [19] Noor, M.A., Waseem, M. and Noor, K.I. *New iterative technique for solving a system of nonlinear equations*, Appl. Math. Comput. 271 (2015) 446–466.
- [20] Ostrowski, A.M. *Solution of equations in Euclidean and Banach spaces*, Third edition of Solution of equations and systems of equations. Pure and Applied Mathematics, Vol. 9. Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1973.
- [21] Petković, M.S. *Multi-step root solvers of Traub's type in real interval arithmetic*, Appl. Math. Comput. 248 (2014) 430–440.
- [22] Rump, S.M. *INTLAB—interval laboratory*, Developments in reliable computing, pp. 77–104. Springer, Dordrecht, 1999.
- [23] Sharma, J.R. and Sharma, R. *A new family of modified Ostrowski's methods with accelerated eighth order convergence*, Numer. Algorithms 54 (2010) 445–458.
- [24] Sunaga, T. *Theory of an interval algebra and its applications to numerical analysis*, [Reprint of Res. Assoc. Appl. Geom. Mem. 2 (1958), 29–46]. Japan J. Indust. Appl. Math. 26(2-3) (2009) 125–143.

#### How to cite this article

Dehghani-Madiseh, M., A family of eight-order interval methods for computing rigorous bounds to the solution to nonlinear equations. *Iran. j. numer. anal. optim.*, 2023; 13(1): [102-120](#).  
<https://doi.org/10.22067/ijnao.2022.74632.1092>.



## Numerical method for solving fractional Sturm–Liouville eigenvalue problems of order two using Genocchi polynomials

A. Aghazadeh, Y. Mahmoudi\*<sup></sup> and F. Dastmalchi Saei<sup></sup>

### Abstract

A new numerical scheme based on Genocchi polynomials is constructed to solve fractional Sturm–Liouville problems of order two in which the fractional derivative is considered in the Caputo sense. First, the differential equation with boundary conditions is converted into the corresponding integral equation form. Next, the fractional integration and derivation operational matrices for Genocchi polynomials, are introduced and applied for approximating the eigenvalues of the problem. Then, the proposed polynomials are applied to approximate the corresponding eigenfunctions. Finally, some examples are presented to illustrate the efficiency and accuracy of the numerical method. The results show that the proposed method is better than some other approximations involving orthogonal bases.

**AMS subject classifications (2020):** Primary 45D05; Secondary 65D99.

**Keywords:** Sturm–Liouville problem; Caputo fractional derivative; Eigenvalue; Eigenfunction; Genocchi polynomials.

---

\*Corresponding author

Received 10 March 2022; revised 1 June 2022; accepted 17 June 2022

Yaghoub Mahmoudi

Department of Mathematics, Tabriz Branch, Islamic Azad University, Tabriz, Iran.  
e-mail: mahmoudi@iaut.ac.ir

Arezu Aghazadeh

Department of Mathematics, Tabriz Branch, Islamic Azad University, Tabriz, Iran.  
e-mail: agharezu@gmail.com

Farhad Dastmalchi Saei

Department of Mathematics, Tabriz Branch, Islamic Azad University, Tabriz, Iran.  
e-mail: dastmalchi@iaut.ac.ir

## 1 Introduction

The purpose of solving the Sturm–Liouville problem is to find the eigenvalues and eigenfunctions of a set of equations, which have many applications in mathematics and physics. The description of the vibrations of a string or a quantum mechanical oscillator is modeled as Sturm–Liouville equations [2, 3, 9]. For more information about the application of Sturm–Liouville problems, we refer to [4, 5, 9, 10, 14] and references therein. With the advent of fractional calculus [8, 20, 24], the use of fractional derivatives in Sturm–Liouville equations led to a new class of equations, which are known as fractional Sturm–Liouville problems. Various types of fractional order derivatives are defined [11, 12, 18, 21, 33], but most researchers use Caputo’s concept for Sturm–Liouville equations because Caputo fractional derivative is more compatible with practical application in physics and engineering [22, 23, 25, 26, 30]. The analytical solution to Sturm–Liouville problems is usually not computable. This problem has limited the application of these equations in various fields. In simpler cases, the analytical solutions to this equation can be expressed in terms of specific functions, such as Mittag-Leffler functions, which have their own complexities in terms of calculations. So, in most cases, scientists seek to find numerical methods to solve Sturm–Liouville problems.

In this study, we consider the following fractional Sturm–Liouville problem:

$${}_0^C D_x^\alpha y(x) + (\lambda r(x) - q(x))y(x) = 0, \quad 0 \leq x \leq 1, \quad (1)$$

where  $q(x)$  and  $r(x)$  are real value continuous functions and  $r(x) \neq 0$  for  $x \in [0, 1]$ ,  $1 < \alpha \leq 2$ , and  ${}_0^C D_x^\alpha$  denotes the fractional Caputo derivative and the boundary conditions are as follows:

$$\begin{cases} ay(0) + by'(0) = 0, \\ cy(1) + dy'(1) = 0, \end{cases} \quad (2)$$

where  $a$ ,  $b$ ,  $c$  and  $d$  are real constants and  $a^2 + b^2 \neq 0$ ,  $c^2 + d^2 \neq 0$ .

The authors in [2] established sufficient conditions for the existence and uniqueness of solutions for various classes of initial and boundary value problems for fractional differential equations involving the Caputo fractional derivative. Also, the existence and uniqueness of the solution for a fractional Sturm–Liouville boundary value problems based on the Banach fixed point theorem were proved in [15]. We also refer to [6, 13, 34] for more studies on the existence and uniqueness of the solution for boundary value problems of types (1) and (2).

A wide range of numerical methods has been used to solve problems (1) and (2) [1, 4, 5, 7, 18, 19]. The Laplace transform method is applied to convert (1)–(2) to the equivalent integral equation with a weakly singular kernel in [31]. Then, the authors applied a piecewise Lagrange integration method to solve the corresponding integral equation numerically. Inspired

by their work, we apply the Genocchi polynomials approximation method to the problem (1)–(2).

Genocchi polynomials, which were introduced in [32], are very important and useful polynomials. These polynomials share some great advantages with Bernoulli and Euler polynomials for approximating an arbitrary smooth function [32]. Genocchi polynomials were applied for solving integer-order delay differential equations [16], fractional optimal control problems [28], and fractional pantograph equation [16]. The numerical solutions obtained by Genocchi polynomials are comparable or even more accurate compared to some well-known existing methods. In this study, motivated by these advantages, we define and successfully apply the operational matrices of Genocchi polynomials to approximate the eigenvalues and the corresponding eigenfunctions of the Sturm–Liouville problem (1) and (2). First, a new approach for calculating the fractional derivative, integration, and product operational matrices is introduced, and then the operational matrices are applied for problems (1) and (2). The structure of the paper is as follows: In Section 2, we recall some definitions and results related to fractional calculus. In Section 3, we construct the new numerical method. Some illustrative examples are provided in Section 4. Finally, the conclusion is given in Section 5.

## 2 Preliminaries

In this part, the definition of fractional calculus, Genocchi polynomials, and their attributes are explained.

### 2.1 Fractional calculus

The Riemann–Liouville fractional integral  ${}_0I_x^\alpha$  of order  $0 \leq \alpha < 1$  is presented with (see [29])

$${}_0I_x^\alpha u(x) = \begin{cases} \frac{1}{\Gamma(\alpha)} \int_0^x (x-s)^{\alpha-1} u(s) ds, & \alpha > 0, \\ u(x), & \alpha = 0. \end{cases} \quad (3)$$

One of the fundamental attributes of the operator  ${}_0I_x^\alpha$  is

$${}_0I_x^\alpha x^\beta = \frac{\Gamma(\beta+1)}{\Gamma(\beta+1+\alpha)} x^{\beta+\alpha}. \quad (4)$$

The Caputo fractional derivative of order  $\alpha > 0$  is determined as [29]



$${}_0^C D_x^\alpha u(x) = {}_0 I_x^{n-\alpha} \frac{d^n}{dx^n} u(x) = \frac{1}{\Gamma(n-\alpha)} \int_0^x (x-s)^{n-\alpha-1} u^{(n)}(s) ds, \quad (5)$$

where  $n$  is an integer ( $n-1 < \alpha \leq n$ ) and  $u^{(n)} \in L^1[0, 1]$ .

The main relationship between the Riemann–Liouville integral operator and Caputo fractional derivative is as follows:

$$\begin{aligned} {}_0^C D_x^\alpha {}_0 I_x^\alpha u(x) &= u(x), \\ {}_0 I_x^\alpha {}_0^C D_x^\alpha u(x) &= u(x) - \sum_{r=0}^{n-1} u^{(r)}(0^+) \frac{x^r}{r!} \quad (n-1 < \alpha \leq n), \end{aligned} \quad (6)$$

where  $u^{(r)} \in L^1[0, 1]$ ,  $r = 0, 1, \dots, n-1$ .

## 2.2 Genocchi polynomials and properties

The Genocchi numbers  $g_n$  and the Genocchi polynomials  $G_n(x)$  are defined as the coefficients of the exponential generating functions as follows:

$$\frac{2t}{e^t + 1} = \sum_{n=0}^{\infty} g_n \frac{t^n}{n!}, \quad |t| < \pi, \quad (7)$$

$$\frac{2te^{xt}}{e^t + 1} = \sum_{n=0}^{\infty} G_n(x) \frac{t^n}{n!}, \quad |t| < \pi. \quad (8)$$

The Genocchi polynomial  $G_n(x)$  is a polynomial given by

$$G_n(x) = \sum_{k=0}^n \gamma_k^n x^k, \quad \gamma_k^n = \binom{n}{k} g_{n-k}, \quad n = 1, 2, \dots \quad (9)$$

The Genocchi polynomials have interesting properties, some of which are as follows:

$$\int_0^1 G_n(x) G_m(x) dx = \frac{2(-1)^n n! m!}{(m+n)!} g_{m+n}, \quad n, m \geq 1, \quad (10)$$

$$\frac{d}{dx} G_n(x) = n G_{n-1}(x), \quad n \geq 1, \quad (11)$$

$$G_n(1) + G_n(0) = 0, \quad n > 1. \quad (12)$$

The set of  $Y = \{G_1(x), G_2(x), \dots, G_N(x)\} \subset L^2[0, 1]$  is a linearly independent set (see [16]). Any  $f(x) \in L^2[0, 1]$  has a unique best approximation in  $Span(Y)$ , as  $f_N(x)$ , which can be represented by Genocchi polynomials as follows:

$$f(x) \simeq f_N(x) = \sum_{n=1}^N c_n G_n(x) = C^T G(x), \quad (13)$$

where  $C = [c_1, c_2, \dots, c_N]^T$  is the coefficient vector and  $G = [G_1(x), G_2(x), \dots, G_N(x)]^T$  is the Genocchi polynomials vector. The property of the best approximation requires

$$\langle f(x), G(x) \rangle^T = C^T \langle G(x), G(x) \rangle,$$

and then

$$C^T = \langle f(x), G(x) \rangle^T \mathbf{W}^{-1}, \quad (14)$$

where

$$\langle f(x), G(x) \rangle = [\langle f(x), G_1(x) \rangle, \langle f(x), G_2(x) \rangle, \dots, \langle f(x), G_N(x) \rangle]^T,$$

and  $\mathbf{W} = \langle G(x), G(x) \rangle$  is an  $N \times N$  symmetric matrix, and by (10) its entries are calculated as follows:

$$\mathbf{W}_{nm} = \langle G_n(x), G_m(x) \rangle = \int_0^1 G_n(x) G_m(x) dx = \frac{2(-1)^n n! m!}{(m+n)!} g_{m+n}. \quad (15)$$

The following lemma provides the upper bound for the error of function approximation by Genocchi polynomials.

**Lemma 1.** Suppose that  $f(x) \in C^{N+1}[0, 1]$  is approximated by truncated Genocchi polynomials  $f_N(x)$  in (13). Then

$$\|f(x) - f_N(x)\|_2 \leq \frac{R}{(N+1)! \sqrt{2N+3}}, \quad (16)$$

where  $R = \max_{x \in [0, 1]} |f^{(N+1)}(x)|$ .

*Proof.* We refer the reader to [17]. □

### 2.3 Genocchi operational matrices

By (11), the derivative of  $G(x)$  can be expressed as follows:

$$\frac{d}{dx} G(x) = \mathbf{D} G(x), \quad (17)$$

where

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 3 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & N-1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & N & 0 \end{bmatrix}, \quad (18)$$

is called the operational matrix of integration for Genocchi polynomials. It is obvious that

$$\frac{d^n}{dx^n} G(x) = \mathbf{D}^n G(x). \quad (19)$$

The operational matrix for fractional order derivative of order  $\alpha$  in Caputo sense is defined as follows:

$${}_0^C D_x^\alpha G(x) = \mathbf{D}^\alpha G(x), \quad (20)$$

where  $\mathbf{D}^\alpha$  is an  $N \times N$  matrix. By using (14), we can write

$$\mathbf{D}^\alpha = \mathbf{Q}^\alpha \mathbf{W}^{-1}, \quad (21)$$

where  $\mathbf{Q}^\alpha = \langle D^\alpha G(x), G(x) \rangle$  is an  $N \times N$  matrix whose entries are defined by

$$\begin{aligned} \mathbf{Q}_{ij}^\alpha &= \langle D^\alpha G_i(x), G_j(x) \rangle \\ &= \left\langle \sum_{k=n}^i \frac{\gamma_k^i k!}{\Gamma(k+1-\alpha)} x^{k-\alpha}, \sum_{r=0}^j \gamma_r^j x^r \right\rangle \\ &= \sum_{k=n}^i \sum_{r=0}^j \frac{\gamma_k^i \gamma_r^j k!}{\Gamma(k+1-\alpha)} \int_0^1 x^{k+r-\alpha} dx \\ &= \sum_{k=n}^i \sum_{r=0}^j \frac{\gamma_k^i \gamma_r^j k!}{\Gamma(k+1-\alpha)(k+r+1-\alpha)}, \quad i \geq n. \end{aligned} \quad (22)$$

It is obvious that the first  $n-1$  rows for  $\mathbf{Q}^\alpha$  are equal to zero, where  $n-1 < \alpha \leq n$ .

The operational matrix for fractional order Riemann–Liouville integration of order  $\alpha$ , ( $0 < \alpha \leq 1$ ) is defined as follows:

$$I^\alpha G(x) = \mathbf{P}^\alpha G(x), \quad (23)$$

where  $\mathbf{P}^\alpha$  is an  $N \times N$  matrix. By using (14), we can write

$$\mathbf{P}^\alpha = \mathbf{V}^\alpha \mathbf{W}^{-1}, \quad (24)$$

where  $\mathbf{V}^\alpha = \langle I^\alpha G(x), G(x) \rangle$  is an  $N \times N$  matrix whose entries are defined by

$$\begin{aligned}
\mathbf{V}_{ij}^\alpha &= \langle I^\alpha G_i(x), G_j(x) \rangle \\
&= \left\langle \sum_{k=0}^i \frac{\gamma_k^i k!}{\Gamma(k+1+\alpha)} x^{k+\alpha}, \sum_{r=0}^j \gamma_r^j x^r \right\rangle \\
&= \sum_{k=0}^i \sum_{r=0}^j \frac{\gamma_k^i \gamma_r^j k!}{\Gamma(k+1+\alpha)} \int_0^1 x^{k+r+\alpha} dx \\
&= \sum_{k=0}^i \sum_{r=0}^j \frac{\gamma_k^i \gamma_r^j k!}{\Gamma(k+1+\alpha)(k+r+1+\alpha)}.
\end{aligned} \tag{25}$$

The product of two Genocchi vectors can be expressed by Genocchi vector as follows:

$$G(x)G^T(x)C = \tilde{\mathbf{C}}G(x), \tag{26}$$

where  $\tilde{\mathbf{C}}$  is called the operational matrix for the product. By using (14), the product operational matrix  $\tilde{\mathbf{C}}$  can be defined as follows:

$$\tilde{\mathbf{C}} = \mathbf{S}\mathbf{W}^{-1}, \tag{27}$$

where  $\mathbf{S} = \langle G(x)G^T(x)C, G(x) \rangle$ . The entries of  $\mathbf{S}$  are determined as

$$\begin{aligned}
\mathbf{S}_{ij} &= \langle (G(x)G^T(x)C)_i, G_j(x) \rangle \\
&= \left\langle G_i(x) \sum_{k=0}^N c_k G_k(x), G_j(x) \right\rangle = \sum_{k=1}^N c_k q_{ijk},
\end{aligned} \tag{28}$$

where

$$\begin{aligned}
q_{ijk} &= \int_0^1 G_i(x)G_k(x)G_j(x)dx \\
&= \sum_{r=0}^i \sum_{s=0}^j \sum_{t=0}^k \frac{\gamma_r^i \gamma_s^j \gamma_t^k}{r+s+t+1}.
\end{aligned} \tag{29}$$

### 3 Method of solution

In this section, we apply Genocchi polynomials to solve problem (1). Consider the fractional Sturm–Liouville problem (1) with the boundary conditions (2). The following Lemma converts this problem to the equivalent integral equation.

**Lemma 2.** The function  $y(x)$  is a solution for the fractional Sturm–Liouville problem (1)–(2) if and only if  $y(x)$  satisfies the following fractional integral equation:

$$y(x) = g(x) [h_1 {}_0I_x^\alpha f(x)|_{x=1} + h_2 D {}_0I_x^\alpha f(x)|_{x=1}] - {}_0I_x^\alpha f(x), \quad (30)$$

where  $f(x) = (\lambda r(x) - q(x))y(x)$  and

$${}_0I_x^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-t)^{\alpha-1} f(t) dt.$$

The operator  $D$  in (30) is the classical derivative operator and  $g(x)$ ,  $h_1$ , and  $h_2$  are defined considering the values of  $a$ ,  $b$ ,  $c$ , and  $d$  as follows:

$$\begin{aligned} g(x) &= \frac{acx - bc}{ad - bc + ac}, \quad h_1 = 1, \quad h_2 = \frac{d}{c}, & \text{if } a \neq 0, \quad c \neq 0, \\ g(x) &= \frac{adx - bd}{ad - bc + ac}, \quad h_1 = \frac{c}{d}, \quad h_2 = 1, & \text{if } a \neq 0, \quad d \neq 0, \\ g(x) &= \frac{-adx + bd}{-ad + bc - ac}, \quad h_1 = \frac{c}{d}, \quad h_1 = 1, & \text{if } b \neq 0, \quad d \neq 0, \\ g(x) &= \frac{-acx + bc}{-ad + bc - ac}, \quad h_1 = 1, \quad h_2 = \frac{d}{c}, & \text{if } b \neq 0, \quad c \neq 0. \end{aligned} \quad (31)$$

*Proof.* We refer to [31]. □

To solve the Sturm-Liouville problem (1)–(2), we apply the Genocchi polynomials method to the integral equation (30). First, we approximate  $y(x)$ ,  $r(x)$ , and  $q(x)$  in (30) by the truncated Genocchi polynomials (13) as follows:

$$y(x) \simeq C^T G(x), \quad r(x) \simeq R^T G(x), \quad q(x) \simeq Q^T G(x). \quad (32)$$

To apply (32) in (30), we first approximate the elements in (30) as follows:

$$\begin{aligned} f(x) &= y(x)(\lambda r(x) - q(x)) \\ &\simeq C^T G(x)(\lambda G^T(x)R - G^T(x)Q) \\ &= C^T (\lambda \tilde{\mathbf{R}} - \tilde{\mathbf{Q}}) G(x) = C^T \tilde{\mathbf{F}} G(x), \end{aligned} \quad (33)$$

where  $\tilde{\mathbf{R}}$  and  $\tilde{\mathbf{Q}}$  are the product operational matrices corresponding to the vectors  $R$  and  $Q$ , respectively, and  $\tilde{\mathbf{F}} = \lambda \tilde{\mathbf{R}} - \tilde{\mathbf{Q}}$ . By (33), the fractional integration part of (30) can be written as follows:

$${}_0I_x^\alpha f(x) \simeq C^T \tilde{\mathbf{F}} {}_0I_x^\alpha G(x) = C^T \tilde{\mathbf{F}} \mathbf{P}^\alpha G(x), \quad (34)$$

where  $\mathbf{P}^\alpha$  is the operational matrix for fractional integration of order  $\alpha$ . The other part of (30) can be represented as

$$D {}_0I_x^\alpha f(x) \simeq C^T \tilde{\mathbf{F}} \mathbf{P}^\alpha D G(x) = C^T \tilde{\mathbf{F}} \mathbf{P}^\alpha \mathbf{D} G(x), \quad (35)$$

where  $\mathbf{D}$  is the operational matrix for regular derivative of order one.

Substituting (32), (33), (34), and (35) in (30) and neglecting the truncation errors yield

$$C^T G(x) = C^T \left\{ \left[ h_1 \tilde{\mathbf{F}} \mathbf{P}^\alpha + h_2 \tilde{\mathbf{F}} \mathbf{P}^\alpha \mathbf{D} \right] G(1) G^T - \tilde{\mathbf{F}} \mathbf{P}^\alpha \right\} G(x). \quad (36)$$

The property of linear independency for Genocchi polynomials requires

$$(\mathbf{I} - \mathbf{A}(\lambda))C = 0, \quad (37)$$

where

$$[\mathbf{A}(\lambda)]^T = \left[ h_1 \tilde{\mathbf{F}} \mathbf{P}^\alpha + h_2 \tilde{\mathbf{F}} \mathbf{P}^\alpha \mathbf{D} \right] G(1) G^T - \tilde{\mathbf{F}} \mathbf{P}^\alpha, \quad (38)$$

and  $\mathbf{I}$  is the  $N \times N$  identity matrix.

To have nontrivial solutions for (1) and (2) we have to find nonzero solutions for (37). Therefor, we need to solve the following root-finder problem:

$$\det(\mathbf{I} - \mathbf{A}(\lambda)) = 0, \quad (39)$$

which can be done by mathematical softwares such as MATLAB and Maple.

After computing eigenvalues, we approximate the eigenfunction  $y_i(t)$  corresponding to the eigenvalue  $\lambda_i$  by solving the following linear system:

$$(\mathbf{I} - \mathbf{A}(\lambda_i))C = 0. \quad (40)$$

Since  $\det(\mathbf{I} - \mathbf{A}(\lambda_i)) = 0$  and  $\lambda_i$  is the simple root, then we set  $C_N = c$  and solve (40) to find  $C_k$ ,  $k = 1, 2, \dots, N - 1$  with respect to  $c$ . Then the eigenfunction  $y_i(t)$  is obtained by

$$y_i(t) \simeq \sum_{k=1}^N C_{k,c} G_k(t), \quad i = 0, 1, \dots \quad (41)$$

By using an appropriate auxiliary condition (for example,  $y'_i(0) = 1$  or  $y''_i(0) = 1$ ),  $y_i(t)$  can be determined uniquely.

## 4 Numerical results

In this section, we present some illustrative examples and show the efficiency of the proposed method.

**Example 1.** Consider the following fractional eigenvalue problem:

$$\begin{aligned} {}^C_0 D_x^\alpha(x) + \lambda y(x) &= 0, & 0 \leq x \leq 1, \\ y(0) &= y(1) = 0. \end{aligned} \quad (42)$$

Table 1: Eigenvalues for  $\alpha = 2$ , Example 1.

|             | exact value      | proposed method   |          | method of [31] |         |
|-------------|------------------|-------------------|----------|----------------|---------|
|             |                  | $N = 25$          | error    | $N = 800$      | error   |
| $\lambda_1$ | 9.86960440108936 | 9.869604401089359 | 2.58e-28 | 9.86960440     | 1.09e-9 |
| $\lambda_2$ | 39.4784176043574 | 39.47841760435743 | 5.22e-19 | 39.47841760    | 4.36e-9 |
| $\lambda_3$ | 88.8264396098042 | 88.82643960980423 | 1.62e-15 | 88.82643963    | 2.02e-8 |
| $\lambda_4$ | 157.913670417430 | 157.9136704174555 | 2.58e-11 | 157.91367057   | 1.53e-7 |
| $\lambda_5$ | 246.740110027234 | 246.7401100283036 | 1.07e-9  | 246.74011064   | 6.13e-7 |
| $\lambda_6$ | 355.305758439217 | 355.3057578574429 | 5.82e-7  | 355.30575850   | 6.08e-8 |
| $\lambda_7$ | 483.610615653379 | 483.6106104947717 | 5.16e-6  | 483.61061573   | 7.66e-8 |
| $\lambda_8$ | 631.654681669719 | 631.6551696075792 | 4.88e-4  | 631.65468191   | 2.40e-7 |

Table 2: Eigenvalues for different values of  $\alpha$ , Example 1 ( $N = 20$ ).

|             | $\alpha = 1.95$ | $\alpha = 1.9$ | $\alpha = 1.85$ | $\alpha = 1.8$ |
|-------------|-----------------|----------------|-----------------|----------------|
| $\lambda_1$ | 9.66077186263   | 9.51414296571  | 9.44013733938   | 9.45685703307  |
| $\lambda_2$ | 36.4045254812   | 33.5956740224  | 30.9825046888   | 28.4768769642  |
| $\lambda_3$ | 80.3290382609   | 73.0390014094  | 66.9448021792   | 62.2003971381  |
| $\lambda_4$ | 140.076253643   | 124.418589764  | 110.360397400   | 97.0631636666  |
| $\lambda_5$ | 216.597587060   | 191.145884721  | 170.311838185   | 155.450547906  |
| $\lambda_6$ | 308.346500930   | 267.943639675  | 232.015405069   | 196.595011516  |

This corresponds to the case  $a \neq 0$  and  $c \neq 0$  in (31). Then, we set  $a = c = 1$ ,  $b = d = 0$ ,  $h_1 = 1$ ,  $h_2 = 0$ , and  $g(x) = x$ . For  $\alpha = 2$ , (42) is converted to the classical Sturm–Liouville problem corresponding to the eigenvalues  $\lambda_n = (n\pi)^2$ .

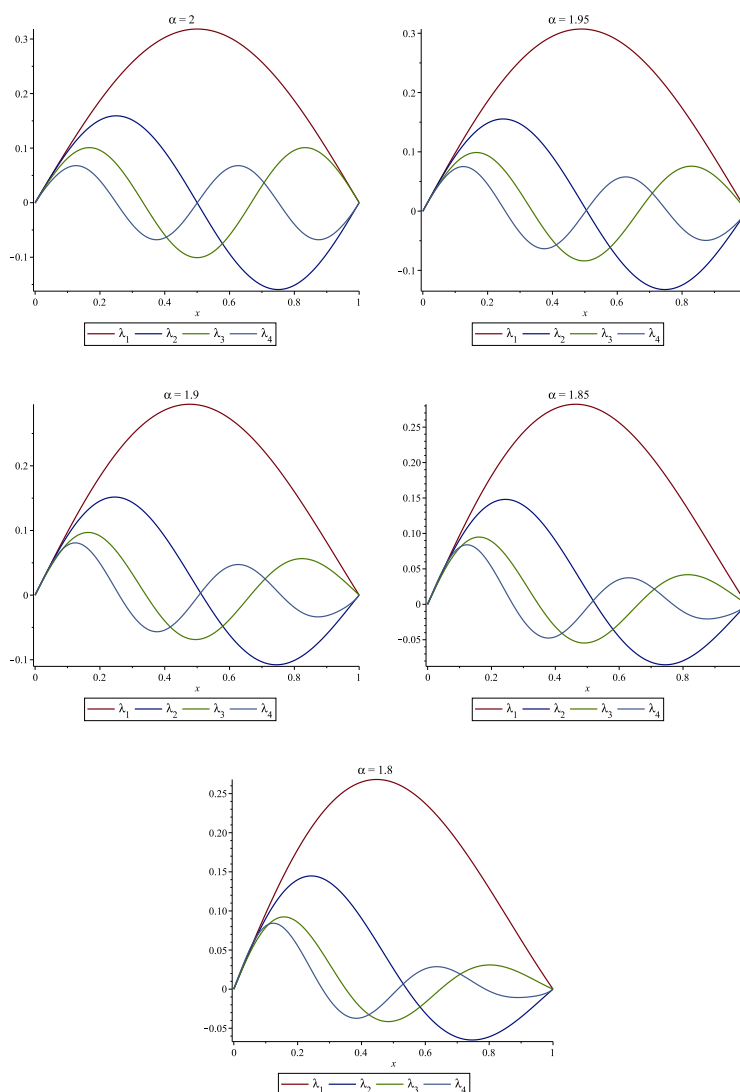
In this example, we use the auxiliary condition  $y'_i(0) = 1$  to approximate the eigenfunctions. In Table 1, the numerical results are compared with the results of [31]. Table 2 represents the approximated eigenvalues obtained by this method. The results in Tables 1 and 2 clearly show that the proposed algorithm is accurate. Figure 1 shows the first four eigenfunctions for  $\alpha = 1.85$ , 1.9, 1.95, 2 and  $N = 15$ . In Figure 2, the graphs of same order eigenvalues are compared for different values of  $\alpha$ .

**Example 2.** For the second example, we solve the following eigenvalue problem:

$${}_0^C D_x^\alpha y(x) + \frac{\lambda}{(1+x)^2} y(x) = 0, \quad 0 \leq x \leq 1, \quad (43)$$

$$y(0) = 0, \quad y(1) = 0.$$

For  $\alpha = 2$ , the eigenvalues of (43) are  $\lambda_n = (n\pi/\ln 2)^2 + 1/4$ , and the corresponding eigenfunctions are  $y_n(x) = c_n \sqrt{1+x} \sin\left(\frac{n\pi \ln(1+x)}{\ln 2}\right)$ , where  $c_n$  is constant coefficient [31]. The entries are chosen  $h_1 = 1$ ,  $h_2 = 0$ , and  $g(x) = x$ . We use the auxiliary condition  $y'_i(0) = 1$  to compute four first eigenfunctions of Example 2, which are plotted in Figure 3. Tables 3 and 4 represent the numerical results obtained for Example 2, which were compared with the results in [31].

Figure 1: First four eigenfunctions for different values of  $\alpha$ , Example 1.

**Example 3.** For the next example, we solve the following eigenvalue problem:

$$\begin{aligned} {}^C_0 D_x^\alpha y(x) + (\lambda + 10 \sin \pi x) y(x) &= 0, & 0 \leq x \leq 1, \\ y(0) = 0, \quad y(1) &= 0. \end{aligned} \quad (44)$$



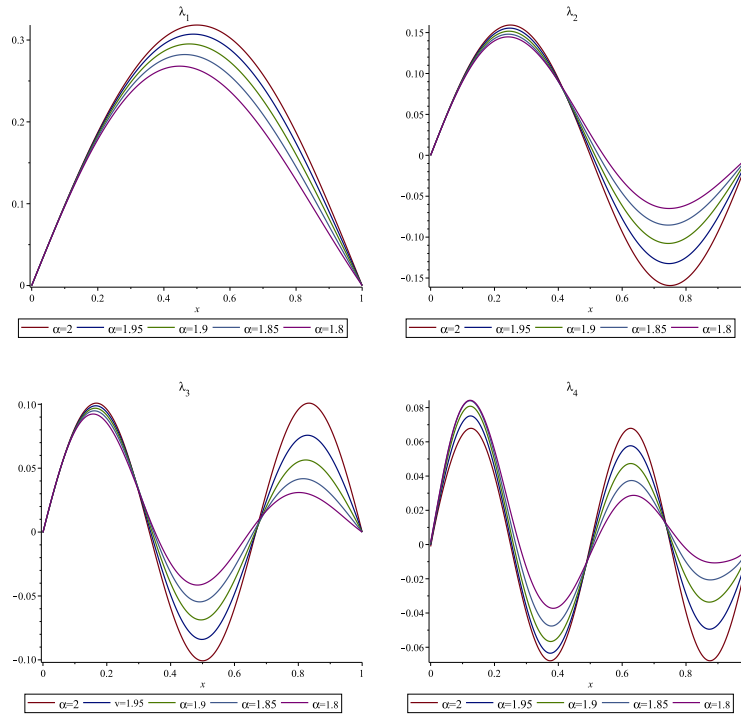


Figure 2: First four eigenfunctions for Example 1.

Table 3: Eigenvalues for  $\alpha = 2$ , Example 2.

|                | exact value      | proposed method  |          | method of [31] |         |
|----------------|------------------|------------------|----------|----------------|---------|
|                |                  | $N = 25$         | error    | $N = 500$      | error   |
| $\lambda_1$    | 20.7922884552238 | 20.7922884552238 | 2.70e-17 | 20.79228809    | 3.65e-7 |
| $\lambda_2$    | 82.4191538208953 | 82.4191538208952 | 6.85e-14 | 82.41914815    | 5.67e-6 |
| $\lambda_3$    | 185.130596097014 | 185.130596097016 | 1.91e-12 | 185.13056780   | 2.83e-5 |
| $\lambda_4$    | 328.926615283581 | 328.926615287143 | 3.56e-9  | 328.92652741   | 8.79e-5 |
| $\lambda_5$    | 513.807211380596 | 513.807211123754 | 2.57e-7  | 513.80700157   | 2.10e-4 |
| $\lambda_6$    | 739.772384388058 | 739.772393148381 | 8.76e-6  | 739.77196118   | 4.23e-4 |
| $\lambda_7$    | 1006.82213430597 | 1006.82198782297 | 1.46e-4  | 1006.82137649  | 7.58e-4 |
| $\lambda_8$    | 1314.95646113432 | 1314.95690886405 | 4.48e-4  | 1314.95521911  | 1.24e-3 |
| $\lambda_9$    | 1664.17536487313 | 1664.19294987497 | 1.76e-2  | 1664.17346838  | 1.90e-3 |
| $\lambda_{10}$ | 2054.47884552238 | 2054.28429017518 | 1.95e-1  | 2054.47611287  | 2.73e-3 |

Setting  $h_1 = 1$ ,  $h_2 = 0$ , and  $g(x) = x$ , we use the proposed method to solve (44). With  $N = 20$  and 40 digits for computation in Maple software, some first eigenvalues are listed in Table 5 for different values of  $\alpha$ .

We use the auxiliary condition  $y'_i(0) = 1$  to compute four first eigenfunctions of Example 3, which are plotted in Figure 4 for different values of  $\alpha$ .

**Example 4.** Consider the following eigenvalue problem:

Table 4: Eigenvalues for different values of  $\alpha$ ,  $N = 20$ , Example 2.

|                | $\alpha = 1.95$ | $\alpha = 1.9$ | $\alpha = 1.85$ | $\alpha = 1.8$ |
|----------------|-----------------|----------------|-----------------|----------------|
| $\lambda_1$    | 20.6677971198   | 20.7298859439  | 21.0420113791   | 21.7289820388  |
| $\lambda_2$    | 75.7710019802   | 69.4887851211  | 63.3252982915   | 56.9307959084  |
| $\lambda_3$    | 167.652678491   | 153.001698048  | 141.496620930   | 134.502772594  |
| $\lambda_4$    | 291.242134899   | 257.628271127  | 226.249088945   | 193.872634358  |
| $\lambda_5$    | 451.065004856   | 398.913421123  | 358.715501294   | 345.306193742  |
| $\lambda_6$    | 641.015025258   | 554.923522781  | 474.802195243   | 379.243396666  |
| $\lambda_7$    | 867.315229409   | 752.921297326  | 669.708580955   | 2260.68651346  |
| $\lambda_8$    | 1122.28888722   | 956.431286954  | 798.315596410   | 3571.46938188  |
| $\lambda_9$    | 1414.21783814   | 1212.08349835  | 1084.45731615   | 8833.11202980  |
| $\lambda_{10}$ | 1732.91424589   | 1458.60978678  | 1178.69721277   | 80410.2798175  |

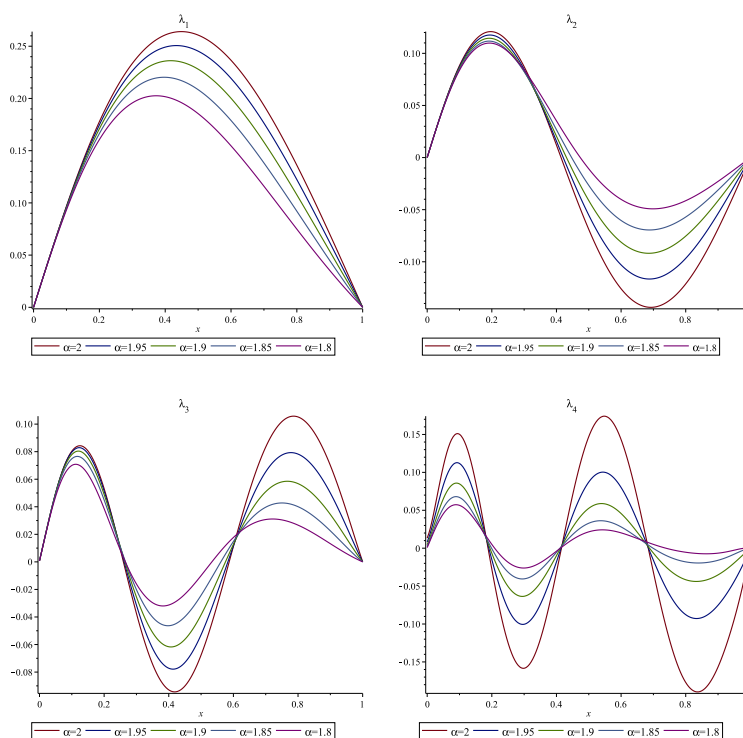


Figure 3: First four eigenfunctions for Example 2.

$$\begin{aligned}
 {}^C D_x^\alpha y(x) + (2\lambda e^x - 5 \sin \pi x) y(x) &= 0, & 0 \leq x \leq 1, \\
 y(0) = y'(0) \neq 0, \quad y(1) &= 0.
 \end{aligned} \tag{45}$$

We set  $a = 1$ ,  $b = -1$ ,  $c = 1$ , and  $d = 0$ . Then  $h_1 = 1$ ,  $h_2 = 0$  and  $g(x) = (x + 1)/2$  (see (33)). With  $N = 25$  and 40 digits for computation, some first eigenvalues are presented and compared in Table 6 for different values of  $\alpha$ .

Table 5: Eigenvalues of Example 3 ( $N = 20$ ).

|             | $\alpha = 2$  | $\alpha = 1.95$ | $\alpha = 1.9$ | $\alpha = 1.8$ |
|-------------|---------------|-----------------|----------------|----------------|
| $\lambda_1$ | 18.3200943208 | 18.1463655269   | 18.0384472376  | 18.0792788380  |
| $\lambda_2$ | 46.2367305574 | 43.1880344976   | 40.4110340394  | 35.3810395900  |
| $\lambda_3$ | 95.3857838965 | 86.8856296038   | 79.5840346405  | 68.6611728543  |
| $\lambda_4$ | 164.391065891 | 146.560791927   | 130.915815214  | 103.643270952  |
| $\lambda_5$ | 253.178225348 | 223.035468745   | 197.577387685  | 161.771332663  |
| $\lambda_6$ | 361.722182212 | 314.767445388   | 274.375010614  | 203.160705578  |
| $\lambda_7$ | 490.014088674 | 423.155049789   | 367.477687375  | 305.476138569  |
| $\lambda_8$ | 637.940873962 | 546.469154163   | 468.674770446  | 315.791118150  |

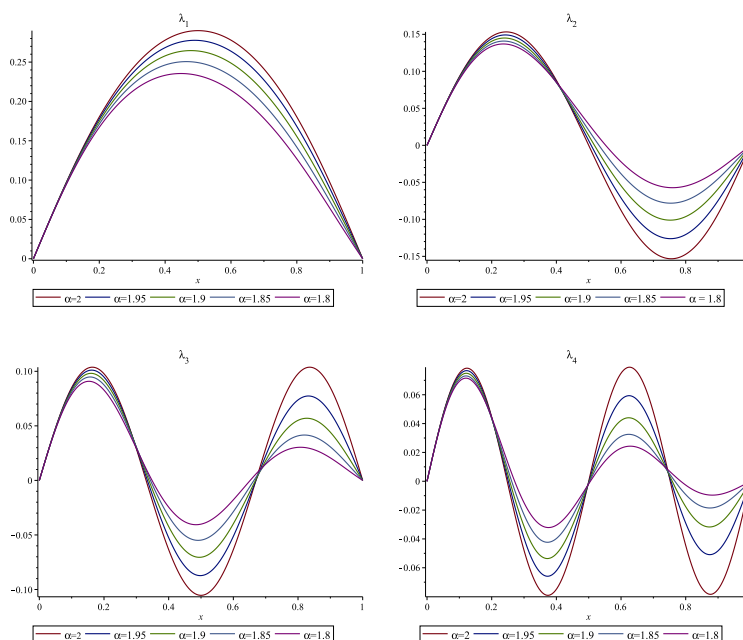


Figure 4: First four eigenfunctions for Example 3.

We use the auxiliary condition  $y_1'(1) = 1$  and  $y_i'(1) = -1$ ,  $i \neq 1$  to compute four first eigenfunctions of Example 4, which are plotted in Figure 5.

**Example 5.** Consider the following forth-order eigenvalue problem: [27]

$$v^{(4)}(t) - (0.02t^2v'(t))' + (0.0001t^4 - 0.02)v(t) = \lambda v(t), \quad 0 \leq t \leq 5, \quad (46)$$

$$v(0) = v''(0) = 0, \quad v(1) = v''(1) = 0.$$

The eigenvalues of this problem are the square of the eigenvalues of the following second order Sturm–Liouville problem (see [27]):

Table 6: Eigenvalues of Example 4.

| proposed method, $N = 25$ |              |                 |                 |                 |                |
|---------------------------|--------------|-----------------|-----------------|-----------------|----------------|
|                           | $\alpha = 2$ | $\alpha = 1.95$ | $\alpha = 1.9$  | $\alpha = 1.85$ | $\alpha = 1.8$ |
| $\lambda_1$               | 2.6213001864 | 2.5785602459    | 2.5408885358    | 2.5083120348    | 2.4809701676   |
| $\lambda_2$               | 8.3495214073 | 7.7826681845    | 7.2770461324    | 6.8263714594    | 6.4250121537   |
| $\lambda_3$               | 20.160277342 | 18.278658822    | 16.631507344    | 15.191185649    | 13.934739577   |
| $\lambda_4$               | 37.779499263 | 33.634640153    | 30.049170175    | 26.945962302    | 24.258034906   |
| $\lambda_5$               | 61.245497320 | 53.803913449    | 47.432736730    | 41.977052803    | 37.309595288   |
| $\lambda_6$               | 90.568134413 | 78.731592840    | 68.678948337    | 60.132525229    | 52.854782758   |
| $\lambda_7$               | 125.75080891 | 108.37362381    | 93.720958762    | 81.357289626    | 70.931030267   |
| $\lambda_8$               | 166.79492842 | 142.69052813    | 122.48797531    | 105.53364534    | 91.272428145   |
| method of [31], $N = 800$ |              |                 |                 |                 |                |
|                           | $\alpha = 2$ | $\alpha = 1.95$ | $\alpha = 1.85$ |                 |                |
| $\lambda_1$               | 2.62130018   | 2.57856033      | 2.50831250      |                 |                |
| $\lambda_2$               | 8.34952147   | 7.78266742      | 6.82636707      |                 |                |
| $\lambda_3$               | 20.16027739  | 18.27866231     | 15.19120752     |                 |                |
| $\lambda_4$               | 37.77949944  | 33.63463032     | 26.94588552     |                 |                |
| $\lambda_5$               | 61.24549782  | 53.80393654     | 41.97727381     |                 |                |

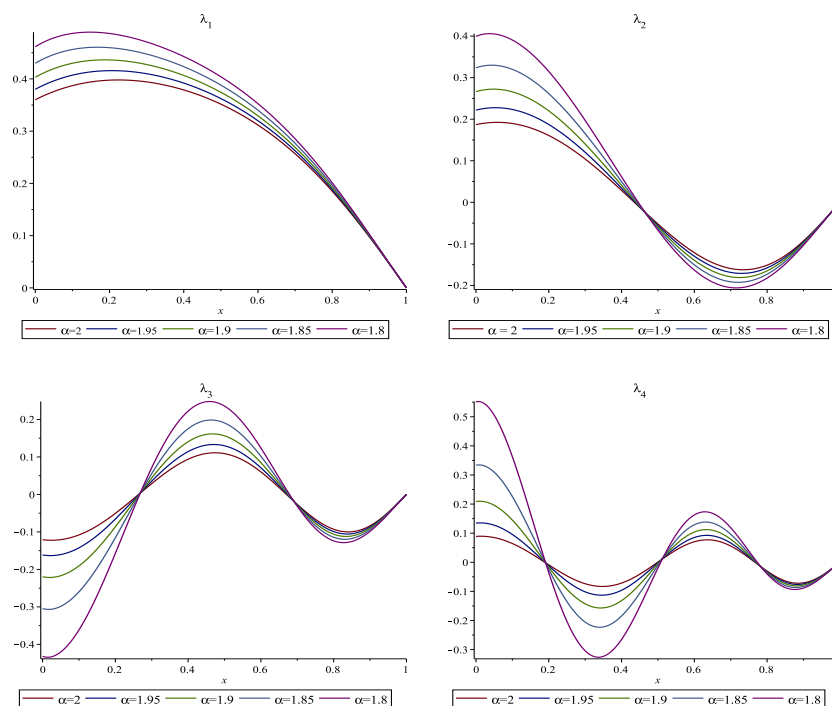


Figure 5: First four eigenfunctions for Example 4.

$$\begin{aligned}
 -u''(t) + 0.01t^2u(t) &= \lambda u(t), & 0 \leq t \leq 5, \\
 u(0) = 0, u(5) &= 0.
 \end{aligned} \tag{47}$$

By taking the variable change  $t = 5x$ , the following problem is achieved:

$$\begin{aligned} y''(x) + (25\lambda - 6.25x^2)y(x) &= 0, & 0 \leq x \leq 1, \\ y(0) &= 0, \quad y(1) = 0. \end{aligned} \quad (48)$$

Taking  $h_1 = 1$ ,  $h_2 = 0$ ,  $g(x) = x$ ,  $r(x) = 25$ , and  $q(x) = 6.25x^2$ , with  $N = 25$  and 40 digits for computation in Maple software, some first eigenvalues are presented and compared with the results of [27] in Table 7. We use the auxiliary condition  $y'_i(0) = 1$  to compute four first eigenfunctions of Example 5, which are plotted in Figure 6.

Table 7: Eigenvalues of Example 5.

|             | proposed method | [27]          |
|-------------|-----------------|---------------|
| $\lambda_1$ | 0.215050864368  | 0.2150508644  |
| $\lambda_2$ | 2.754809934683  | 2.7548099347  |
| $\lambda_3$ | 13.21535154056  | 13.2153515406 |
| $\lambda_4$ | 40.95081975918  | 40.9508197592 |
| $\lambda_5$ | 99.05347806596  | 99.0534780635 |
| $\lambda_6$ | 204.3557315637  | 204.355732268 |
| $\lambda_7$ | 377.4304091791  | 377.430420689 |
| $\lambda_8$ | 642.5918663327  | 642.590868170 |

## 5 Conclusion

In this paper, the eigenvalues of the second-order fractional Sturm–Liouville problem were approximated by using Genocchi polynomials. The operational matrix for fractional integration, fractional derivative, and product was evaluated. Then, the operational matrices were applied to the fractional Sturm–Liouville problem to convert it into a homogeneous system of linear equations. The eigenvalues of the coefficient matrix corresponded to the eigenvalues of the main Sturm–Liouville problem. The presented illustrative examples verified that the proposed method generated more accurate approximations compared to the results, which were reported in other papers. Like other methods in the literature, the results were accurate for lower indices but not very accurate for higher indices. After computing the eigenvalues, the corresponding eigenfunctions were approximated by applying an auxiliary condition. The results were in good agreement with the exact solutions or the results of other papers.

## References

- [1] Abbasbandy, S. and Shirzadi, A. *Homotopy analysis method for multiple solutions of the fractional Sturm–Liouville problems*, Numer. Algorithms 54 (4) (2010) 521–532.
- [2] Agarwal, R., Benchohra, M. and Hamani, S. *A survey on existence results for boundary value problems of nonlinear fractional differential*

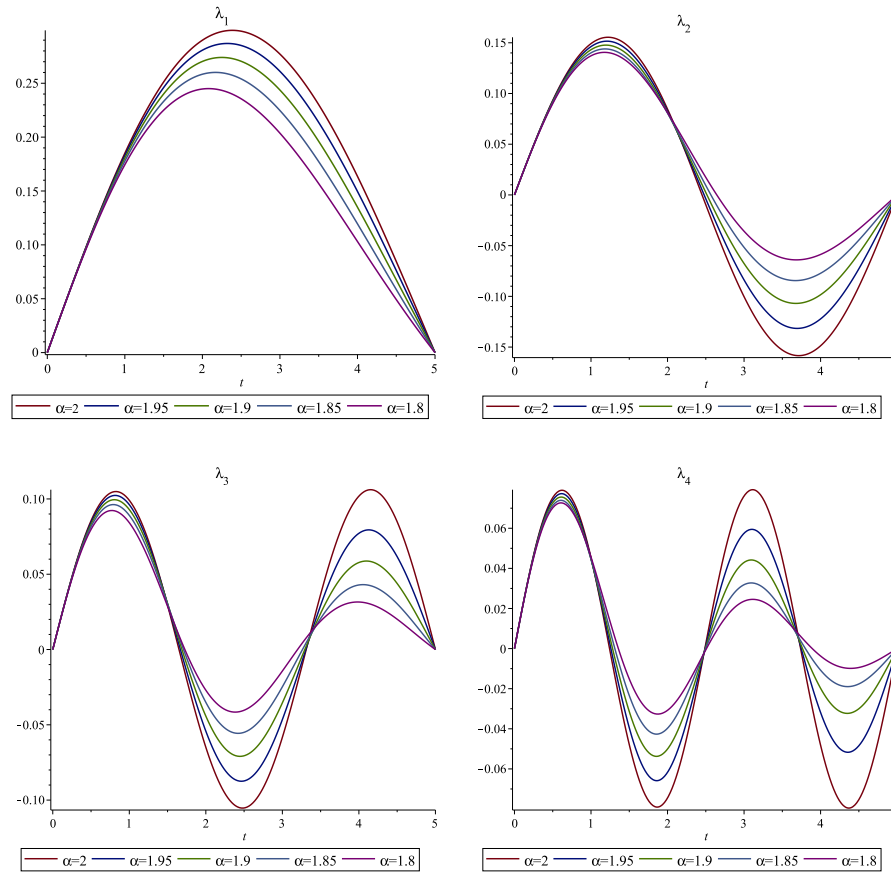


Figure 6: First four eigenfunctions for Example 5.

*equations and inclusions*, Acta Appl. Math. 109(3) (2010) 973–1033.

- [3] Akbarfam, A.J. and Mingarelli, A. *Duality for an indefinite inverse Sturm–Liouville problem*, J. Math. Anal. Appl. 312(2) (2005) 435–463.
- [4] Al-Mdallal, Q. *An efficient method for solving fractional Sturm–Liouville problems*, Chaos Solitons Fractals 40(1) (2009) 183–189.
- [5] Al-Mdallal, Q. *On the numerical solution of fractional Sturm–Liouville problems*, Int. J. Comput. Math. 87(12) (2010) 2837–2845.
- [6] Al-Refai, M. *Basic results on nonlinear eigenvalue problems of fractional order*, Electron. J. Differential Equations 2012, No. 191, 12 pp.

- [7] Antunes, P. and Ferreira, R.A. *An augmented-RBF method for solving fractional Sturm-Liouville eigenvalue problems*, SIAM J. Sci. Comput. 37(1) (2017) A515–A535.
- [8] Asl, M. and Javidi, M. *An improved pc scheme for nonlinear fractional differential equations: Error and stability analysis*, J. Comput. Appl. Math. 324 (2017) 101–117.
- [9] Atkinson, F.V. and Mingarelli, A.B. *Multiparameter eigenvalue problems: Sturm-Liouville theory*, CRC Press, 2010.
- [10] Bas, E. and Metin, F. *Spectral analysis for fractional hydrogen atom equation*, Adv. Pure Appl. Math. 5(13) (2015) 767.
- [11] Blaszczyk, T. and Ciesielski, M. *Numerical solution of fractional Sturm-Liouville equation in integral form*, Fract. Calc. Appl. Anal. 17(2) (2014) 307–320.
- [12] Dehghan, M. and Mingarelli, A. *Fractional Sturm-Liouville eigenvalue problems*, I. Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Mat. RACSAM 114(2) (2020) Paper No. 46, 15 pp.
- [13] El-Sayed, A. and Gaafar, F. M. *Existence and uniqueness of solution for Sturm-Liouville fractional differential equation with multi-point boundary condition via Caputo derivative*, Adv. Difference Equ. 2019(1) (2019) 1–17.
- [14] Fix, G. and Roof, J. *Least squares finite-element solution of a fractional order two-point boundary value problem*, Comput. Math. Appl. 48(7-8) (2004) 1017–1033.
- [15] Hani, R.M. *Existence and uniqueness of the solution for fractional sturm-liouville boundary value problem*, Coll. Basic Educ. Res. J. 11(2) (2011) 698–710.
- [16] Isah, A. and Phang, C. *Operational matrix based on Genocchi polynomials for solution of delay differential equations*, Ain Shams Eng. J. 9(4) (2018) 2123–2128.
- [17] Isah, A., Phang, C. and Phang, P. *Collocation method based on Genocchi operational matrix for solving generalized fractional pantograph equations*, Int. J. Differ. Equ. 2017, Art. ID 2097317, 10 pp.
- [18] Jin, B., Lazarov, R., Pasciak, J. and Rundell, W. *A finite element method for the fractional Sturm-Liouville problem*, arXiv preprint arXiv:1307.5114 (2013).
- [19] Jin, B. and Rundell, W. *An inverse Sturm-Liouville problem with a fractional derivative*, J. Comput. Phys. 231(14) (2012) 4954–4966.

- [20] Kexue, L. and Jigen, P. *Laplace transform and fractional differential equations*, Appl. Math. Lett. 24(12) (2011) 2019–2023.
- [21] Klimek, M. and Blasik, M. *Regular Sturm–Liouville problem with Riemann–Liouville derivatives of order in (1, 2): discrete spectrum, solutions and applications*, in: Advances in Modelling and Control of Non-Integer-Order Systems (2015) 25–36, Springer, Cham.
- [22] Luchko, Y. *Initial–boundary-value problems for the one-dimensional time-fractional diffusion equation*, Fract. Calc. Appl. Anal. 15(1) (2012) 141–160.
- [23] Luo, W., Huang, T., Wu, G. and Gu, X. *Quadratic spline collocation method for the time fractional subdiffusion equation*, Appl. Math. Comput. 276 (2016) 252–265.
- [24] Luo, W., Li, C., Huang, T., Gu, X. and Wu, G. *A high-order accurate numerical scheme for the Caputo derivative with applications to fractional diffusion problems*, Numer. Funct. Anal. Optim. 39(5) (2018) 600–622.
- [25] Metzler, R. and Klafter, J. *Boundary value problems for fractional diffusion equations*, Physica A 278 (1-2) (2000) 107–125.
- [26] Metzler, R. and Nonnenmacher, T. *Space-and time-fractional diffusion and wave equations, fractional Fokker–Planck equations, and physical motivation*, Coll. Basic Educ. Res. J. 284 (1-2) (2002) 67–90.
- [27] Mirzaei, H. *Computing the eigenvalues of fourth order Sturm–Liouville problems with lie group method*, Iranian Journal of Numerical Analysis and Optimization 7(1) (2017) 1–12.
- [28] Phang, C., Ismail, N.F., Isah, A., Loh, J.R. *A new efficient numerical scheme for solving fractional optimal control problems via a Genocchi operational matrix of integration*, J. Vib. Control 24(14) (2018) 3036–3048.
- [29] Podlubny, I. *Fractional differential equations*, Academic Press, San Diego, Calif, USA, 1999.
- [30] Rossikhin, Y. and Shitikova, M. *Application of fractional calculus for dynamic problems of solid mechanics: novel trends and recent results*, Appl. Mech. Rev. 63 (1) (2010) 010801.
- [31] Sadabad, M. K. and Akbarfam, A.J. *An efficient numerical method for estimating eigenvalues and eigenfunctions of fractional Sturm–Liouville problems*, Math. Comput. Simulation 185 (2021) 547–569.



- [32] Tohidi, E., Bhrawy, A. and Erfani, K. *A collocation method based on Bernoulli operational matrix for numerical solution of generalized pantograph equation*, Appl. Math. Model. 37(6) (2013) 4283–4294.
- [33] Zayernouri, M. and Karniadakis, G. *Fractional Sturm–Liouville eigenproblems: theory and numerical approximation*, J. Comput. Phys. (2013) 495–517.
- [34] Zhang, S. *Existence of solution for a boundary value problem of fractional order*, Acta Math. Sci. 26(2) (2006) 220–228.

**How to cite this article**

Aghazadeh, A., Mahmoudi, Y. and Dastmalchi Saei, F., Numerical method for solving fractional Sturm–Liouville eigenvalue problems of order two using Genocchi polynomials. *Iran. j. numer. anal. optim.*, 2023; 13(1): [121-140](#).  
<https://doi.org/10.22067/IJNAO.2022.75635.1115>.



# Impact of inclination angle on thermo-bioconvection of nanofluid containing gyrotactic microorganisms saturated in porous square cavity

J. Bodduna<sup></sup>, C.S. Balla<sup>\*</sup>,<sup></sup> and M.P. Mallesh

## Abstract

This paper focuses on the result of inclined angle on bioconvection of porous media bounded by cavity wall square enclosure filled with both nanofluid and gyrotactic microorganisms passing through the media with pores. The dimensionless velocity, temperature, concentration, and mass transformation equations are solved by using the weighted residual Galerkin's finite element method. The result of the inclination angle from  $\delta = 0^\circ$  to  $\delta = 180^\circ$  in a square cavity is interpreted. The outcomes of inclination on various key parameters, such as Rayleigh number, bioconvective Rayleigh number, Peclet number, Brownian motion, and the ratio of buoyancy, are discussed. Furthermore, the mean Nusselt number, Sherwood number, and density number are discussed at vertical walls.

**AMS subject classifications (2020):** Primary

**Keywords:** Nanofluid, Inclination angle, Buoyancy ratio, Peclet number, Square cavity, Bioconvection.

---

\*Corresponding author

Received 16 May 2022; revised 15 June 2022; accepted 20 June 2022

Jamuna Bodduna

Department of Engineering Mathematics, Research Scholar, Koneru Lakshmaiah Education Foundation, Hyderabad, India. e-mail: jamunabodduna@gmail.com

Chandra Shekar Balla

Department of Mathematics, Chaitanya Bharathi Institute of Technology, Hyderabad, India. e-mail: shekar.balla@gmail.com

M P Mallesh

Department of Engineering Mathematics, Koneru Lakshmaiah Education Foundation, Hyderabad, India. e-mail: malleshmardanpally@gmail.com

## 1 Introduction

The present problem is dealing with the nanofluid flow through the porous square cavity with temperature difference, which has a wide range of applications in recent years, such as geophysics, geothermal energy utilization, and many technologies. The bioconvection of the nanofluid containing gyrotactic microorganisms has a wide range of practical applications, such as chemical catalytic converters, buried electronic cables, pollutant dispersion in aquifers, food industrial forms, and so on. These types of many areas of applications are documented in these references [10, 24, 14, 15, 31].

The properties and utilization of the nanofluid were first introduced by Choi and Eastman [11] at ASME annual meeting. Many people have described the properties of nanofluids, such as [13, 30, 24, 23, 26]. In many electronic devices, like computers, boilers, converters, and so on, the angle of inclination to the surface affects the gravity force on the fluid, temperature gradient, and velocity of the fluid flow. In [32], the author expressed the free convection of the composite wall enclosure. Kuyper et al. [20] studied the effect of inclined angle on different flows in square cavity walls. Kuznetsov [21] explained the microscopic convection motion of the oxytactic microorganisms due to the temperature effect. Shermet and Pop [28] expressed that the result of the thermal movement of microorganisms in the nanofluid having gyrotactic microorganisms is a closed porous square cavity. Aziz, Khan, and Pop [5] presented the flow behavior of the nanofluid with gyrotactic microorganisms on a flat plate. At viscous dissipation, the behavior of oxytactic microorganisms in porous square cavities was discussed [2, 3]. Jamuna and Balla [17] discussed the behavior of the heat source and sink of the gyrotactic microorganisms in the square cavity. The activation energy effect on the gyrotactic microorganisms was discussed in [18]. The influence of Soret and Dufour on free convection of the fluid flow in the inclined four-side closed walls was explained in [8]. The MHD double-diffusion in the porous square enclosure with radiation and chemical reaction and the outcome inclination of the porous square cavity filled with gyrotactic microorganisms with the heat transformation was discussed in [6, 7]. Nanofluid movement in an inclined square cavity with gyrotactic microorganisms at MHD free convection was reported in [27].

The effect of angle movement of the square adiabatic wall on mixed convection of the nanofluid was explored in [16]. Aounallah et al. [4] explained the turbulent flow behavior of the nanofluid in an inclined square cavity on free convection, and Sheremet, Grosan, and Pop [27] investigated the free convective flow of nanofluid in inclined four-sided chamber with gyrotactic microorganisms. Tsai, Li, and Lin [29] discussed the inclination of the plate shield, and Aboueian-Jahromi, Hossein Nezhad, and Behzadmehr [1] studied the steady flow in inclined cylinders.

Rajarathinam and Nithyadevi [25] examined the movement of Cu-water nanofluid in inclined cavity walls with pores. The thermosolutal Maragoni

effects of the bioconvective fluid flow with gyrotactic microorganisms on inclined sheets were explained in [19]. Recently, Varol, Oztop, and Koca [34] explained different fluids' laminar flow in the different inclined enclosures.

Since, from the above literature survey, we note that many authors concentrated on the inclined angle of different geometries with convection of nanofluid with gyrotactic microorganisms. The novelty of this paper contains the square-shaped cavity enclosure with fluid containing nanoparticles and gyrotactic microorganisms. Galerkin's finite-element method is used to solve the nondimensional governing equations.

## 2 Mathematical modeling

We consider the bioconvection flow in an inclined two-dimensional porous four-sided square cavity of dimension  $L$  containing nanofluid with gyrotactic microorganisms. Let us assume that  $\delta$  is the inclination angle of the cavity wall with the horizontal surface. The vertical walls are maintained in various temperatures  $T_C$  and  $T_H$ , respectively ( $T_H > T_C$ ). The remaining walls were kept perfectly insulated. The direction of gravity force  $g$  acts opposite to the vertical axis (Y-axis).

The steady-state Darcy–Boussinesq approximation governing equations are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (1)$$

$$\frac{\mu}{k}u = -\frac{\partial p}{\partial x} - [(\rho_p - \rho_f)(C - C_{\min}) - (1 - C_{\min})\rho_f\beta(T - T_C) + \gamma n\Delta\rho]g \sin \delta, \quad (2)$$

$$\frac{\mu}{k}v = -\frac{\partial p}{\partial y} - [(\rho_p - \rho_f)(C - C_{\min}) - (1 - C_{\min})\rho_f\beta(T - T_C) + \gamma n\Delta\rho]g \cos \delta, \quad (3)$$

$$\begin{aligned} u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = & \alpha_m \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + \tau D_B \left( \frac{\partial C}{\partial x} \frac{\partial T}{\partial x} + \frac{\partial C}{\partial y} \frac{\partial T}{\partial y} \right) \\ & + \frac{\tau D_T}{T_C} \left[ \left( \frac{\partial T}{\partial x} \right)^2 + \left( \frac{\partial T}{\partial y} \right)^2 \right], \end{aligned} \quad (4)$$

$$u \frac{\partial C}{\partial x} + v \frac{\partial C}{\partial y} = D_m \left( \frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) + \frac{D_T}{T_c} \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right), \quad (5)$$

$$\frac{\partial}{\partial x} (un + \tilde{u}n - D_n \frac{\partial n}{\partial x}) + \frac{\partial}{\partial y} (vn + \tilde{v}n - D_n \frac{\partial n}{\partial y}) = 0. \quad (6)$$

Here  $\gamma$  is the mean volume for microorganisms,  $\Delta\rho = \rho_{cell} - \rho_f$  is the density difference of cell,  $T_C$  is the cold wall temperature,  $T_H$  is the hot wall temperature,  $\alpha_m$  is porous medium thermal diffusivity,  $C$  is the concentration of nanoparticles,  $C_0$  is nanoparticles average density,  $C_{\min}$  is minimum concentration of oxygen essential for microorganisms,  $C_p$  is specific heat at

constant pressure,  $D_n$  is microorganisms diffusion coefficient,  $D_B$  is the Brownian diffusion constant,  $D_T$  is the thermophoretic diffusion coefficient,  $n$  is motile density number of microorganisms,  $g$  is the gravity force, chemotaxis constant is  $b$ , and the maximum speed of cell swims is  $w_C$ . The average swimming velocities of microorganisms are  $\tilde{u}$  and  $\tilde{v}$  given as

$$\tilde{u} = \frac{bw_C}{\Delta C} \frac{\partial C}{\partial x}, \quad \tilde{v} = \frac{bw_C}{\Delta C} \frac{\partial C}{\partial y}. \quad (7)$$

Consider dimensional stream function  $\psi$ . Then  $u$  and  $v$  in  $x$  and  $y$  directions are considered as  $u = \frac{\partial \psi}{\partial y}$  and  $v = \frac{\partial \psi}{\partial x}$  by introducing the boundedless variables  $X = \frac{x}{H}$ ,  $Y = \frac{y}{H}$ ,  $\Psi = \frac{\psi}{\alpha_m}$ ,  $\theta = \frac{T-T_C}{T_H-T_C}$ ,  $\phi = \frac{C-C_{\min}}{\Delta C}$ , and  $N = \frac{n}{n_0}$ , where  $n_0$  is the microorganism averaged density.

Substituting above unbounded variables into equation (1)–(7), then we get the following partial differential equations:

$$\begin{aligned} \frac{\partial^2 \Psi}{\partial X^2} + \frac{\partial^2 \Psi}{\partial Y^2} = & RaNr \left( \frac{\partial \phi}{\partial X} \cos \delta - \frac{\partial \phi}{\partial Y} \sin \delta \right) - Ra \left( \frac{\partial \theta}{\partial X} \cos \delta - \frac{\partial \theta}{\partial Y} \sin \delta \right) \\ & + RaRb \left( \frac{\partial N}{\partial X} \cos \delta - \frac{\partial N}{\partial Y} \sin \delta \right), \end{aligned} \quad (8)$$

$$\begin{aligned} \left( \frac{\partial \Psi}{\partial Y} \frac{\partial \theta}{\partial X} - \frac{\partial \Psi}{\partial X} \frac{\partial \theta}{\partial Y} \right) = & \left( \frac{\partial^2 \theta}{\partial X^2} + \frac{\partial^2 \theta}{\partial Y^2} \right) + Nb \left( \frac{\partial \phi}{\partial X} \frac{\partial \theta}{\partial X} + \frac{\partial \phi}{\partial Y} \frac{\partial \theta}{\partial Y} \right) \\ & + Nt \left[ \left( \frac{\partial \theta}{\partial X} \right)^2 + \left( \frac{\partial \theta}{\partial Y} \right)^2 \right], \end{aligned} \quad (9)$$

$$Le \left( \frac{\partial \Psi}{\partial Y} \frac{\partial \phi}{\partial X} - \frac{\partial \Psi}{\partial X} \frac{\partial \phi}{\partial Y} \right) = \frac{\partial^2 \phi}{\partial X^2} + \frac{\partial^2 \phi}{\partial Y^2} + \frac{Nt}{Nb} \left( \frac{\partial^2 \theta}{\partial X^2} + \frac{\partial^2 \theta}{\partial Y^2} \right), \quad (10)$$

$$\frac{\partial \Psi}{\partial X} \frac{\partial N}{\partial Y} - \frac{\partial \Psi}{\partial Y} \frac{\partial N}{\partial X} + \frac{PrPe}{Sc} \left( \frac{\partial^2 \phi}{\partial X^2} + \frac{\partial^2 \phi}{\partial Y^2} \right) = \frac{Pr}{Sc} \left( \frac{\partial^2 N}{\partial X^2} + \frac{\partial^2 N}{\partial Y^2} \right), \quad (11)$$

where  $Ra = \frac{gK\beta(1-C_0)\Delta TL}{v\alpha_m}$ ,  $Rb = \frac{\gamma\Delta\rho n_0}{\rho_f\beta(1-C_0)\Delta T}$ ,  $Le = \frac{\alpha_m}{D_B}$ ,  $Pe = \frac{bw_C}{D_n}$ ,  $Nb = \frac{\tau D_B \Delta C}{\alpha_m}$ ,  $Nt = \frac{\tau D_T (T_H - T_C)}{\alpha_m T_C}$ ,  $Pr = \frac{\mu_f}{\rho_f \alpha_f}$ ,  $Sc = \frac{\mu_f}{\rho_f D_n}$ , and  $Nr = \frac{(\rho_p - \rho_f)C_0}{\rho_f \beta(1-C_0)\Delta T}$ .

The dimensionless form of conditions at boundary is expressed in Figure 1.

We have  $\Psi = 0$  for all sides,

$\phi = 1$ ,  $\theta = 1$ ,  $N = 1$  at  $X = 0$ ,

$\phi = 1$ ,  $\theta = 0$ ,  $N = 1$  at  $X = 1$ ,

$\phi = 1$ ,  $\frac{\partial \theta}{\partial Y} = 0$ ,  $Pe.N \frac{\partial \phi}{\partial Y} = \frac{\partial N}{\partial Y}$  at  $Y = 0$ , and

$$\frac{\partial \phi}{\partial Y} = \frac{\partial \theta}{\partial Y} = \frac{\partial N}{\partial Y} = 0 \quad \text{at } Y = 1.$$

Local solid Nusselt number, Sherwood number of nano particles and Sherwood microorganism are defined as

$$Nu_Y = -\left(\frac{\partial \theta}{\partial X}\right)_{X=0,1}, \quad Sh_Y = -\left(\frac{\partial \phi}{\partial X}\right)_{X=0,1}, \quad \text{and } Nn_Y = -\left(\frac{\partial N}{\partial X}\right)_{X=0,1}.$$

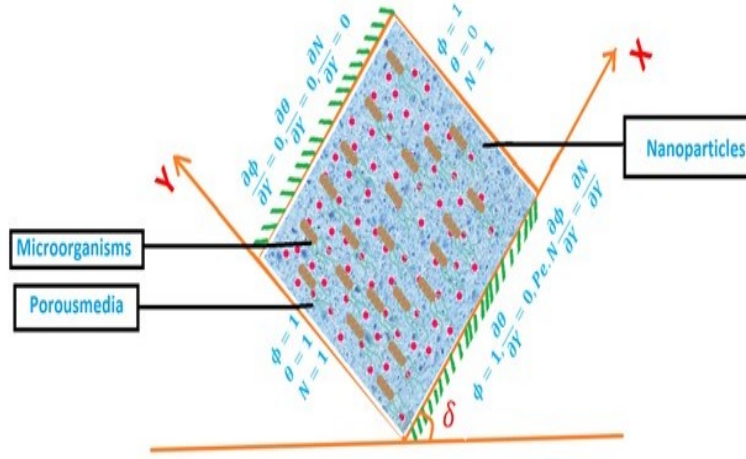


Figure 1: Physical geometry and coordinate system.

The average quantities of Nusselt number, nanoparticle Sherwood number, and microorganism Sherwood number is defined as

$$\begin{aligned} Nu_{avg} &= \int_0^1 Nu_Y dY, \\ Sh_{avg} &= \int_0^1 Sh_Y dY, \\ Nn_{avg} &= \int_0^1 Nn_Y dY. \end{aligned}$$

### 3 Numerical method

To find the numerical solution to (8)–(11), partial differential equations with boundary conditions Galerkin's weighted residuals finite element method are solved with the help of MATLAB [9]. In this method, a two-dimensional field is divided into small triangular parts, in which each part is named an element. Over each element, assume a piecewise trial function.

Let  $\Psi$ ,  $\theta$ ,  $\phi$ , and  $N$  be approximated by  $\Psi = \sum_{i=1}^3 \Psi_i \xi_i$ ,  $\theta = \sum_{i=1}^3 \theta_i \xi_i$ ,  $\phi = \sum_{i=1}^3 \phi_i \xi_i$ , and  $N = \sum_{i=1}^3 N_i \xi_i$ , where  $\xi_i$  is the linear interpolating functions over each triangular element. The FEM model matrix is as follows:

$$\begin{bmatrix} [L^{11}] & [L^{12}] & [L^{13}] & [L^{14}] \\ [L^{21}] & [L^{22}] & [L^{23}] & [L^{24}] \\ [L^{31}] & [L^{32}] & [L^{33}] & [L^{34}] \\ [L^{41}] & [L^{42}] & [L^{43}] & [L^{44}] \end{bmatrix} \begin{bmatrix} \{\Psi\} \\ \{T\} \\ \{C\} \\ \{N\} \end{bmatrix} = \begin{bmatrix} \{M^1\} \\ \{M^2\} \\ \{M^3\} \\ \{M^4\} \end{bmatrix},$$

where

$$L^{11} = \iint_{\Omega_e} \left[ \frac{\partial \xi_j}{\partial X} \frac{\partial \xi_i}{\partial X} + \frac{\partial \xi_j}{\partial Y} \frac{\partial \xi_i}{\partial Y} \right] dx dy,$$

$$\begin{aligned}
L^{12} &= -Ra \iint_{\Omega_e} (\xi_j \frac{\partial \xi_i}{\partial X} \cos \delta - \xi_j \frac{\partial \xi_i}{\partial Y} \sin \delta) dX dY, \\
L^{13} &= RaNr \iint_{\Omega_e} (\xi_j \frac{\partial \xi_i}{\partial X} \cos \delta - \xi_j \frac{\partial \xi_i}{\partial Y} \sin \delta) dX dY, \\
L^{14} &= RaRb \iint_{\Omega_e} (\xi_j \frac{\partial \xi_i}{\partial X} \cos \delta - \xi_j \frac{\partial \xi_i}{\partial Y} \sin \delta) dX dY, \\
M^1 &= 0, \\
L^{21} &= 0, \\
L^{22} &= \iint_{\Omega_e} [\frac{\partial \Psi}{\partial Y} \xi_j \frac{\partial \xi_i}{\partial X} - \frac{\partial \Psi}{\partial X} \xi_j \frac{\partial \xi_i}{\partial Y} + \frac{\partial \xi_i}{\partial X} \frac{\partial \xi_j}{\partial X} + \frac{\partial \xi_i}{\partial Y} \frac{\partial \xi_j}{\partial Y} - Nt(\frac{\partial \theta}{\partial X} \xi_j \frac{\partial \xi_i}{\partial X} + \frac{\partial \theta}{\partial Y} \xi_j \frac{\partial \xi_i}{\partial Y})] dX dY, \\
L^{23} &= -Nb \iint_{\Omega_e} [\frac{\partial \theta}{\partial X} \xi_j \frac{\partial \xi_i}{\partial X} + \frac{\partial \theta}{\partial Y} \xi_j \frac{\partial \xi_i}{\partial Y}] dX dY, \\
L^{24} &= 0, M^2 = 0, \\
L^{31} &= 0, \\
L^{32} &= -\frac{Nt}{Nb} \iint_{\Omega_e} [\frac{\partial \xi_i}{\partial X} \frac{\partial \xi_j}{\partial X} + \frac{\partial \xi_i}{\partial Y} \frac{\partial \xi_j}{\partial Y}] dX dY, \\
L^{33} &= \iint_{\Omega_e} [Le \left( \frac{\partial \Psi}{\partial Y} \frac{\partial \phi}{\partial X} - \frac{\partial \Psi}{\partial X} \frac{\partial \phi}{\partial Y} \right) + \frac{\partial \xi_i}{\partial X} \frac{\partial \xi_j}{\partial X} + \frac{\partial \xi_i}{\partial Y} \frac{\partial \xi_j}{\partial Y}] dX dY, \\
L^{34} &= 0, M^3 = 0, \\
L^{41} &= 0, L^{42} = 0, \\
L^{43} &= \iint_{\Omega_e} \frac{PePr}{Sc} (\frac{\partial \xi_i}{\partial X} \frac{\partial \xi_j}{\partial X} + \frac{\partial \xi_i}{\partial Y} \frac{\partial \xi_j}{\partial Y}) dX dY, \\
L^{44} &= \iint_{\Omega_e} [\frac{\partial \Psi}{\partial Y} \xi_j \frac{\partial \xi_i}{\partial X} - \frac{\partial \Psi}{\partial X} \xi_j \frac{\partial \xi_i}{\partial Y} + \frac{Pr}{Sc} (\frac{\partial \xi_i}{\partial X} \frac{\partial \xi_j}{\partial X} + \frac{\partial \xi_i}{\partial Y} \frac{\partial \xi_j}{\partial Y})] dX dY, \\
M^4 &= 0.
\end{aligned}$$

To linearize the system of equations, the functions are incorporated, which are assumed to be known. After applying the boundary conditions, a matrix of system of linear equations is formed, which is solved by using the Gauss-Seidel iteration method. The convergence of the solution is assumed when the relative error for each variable between two consecutive iterations is observed below the convergence criteria such that  $|\psi^{n+1} - \psi^n| \leq 10^{-5}$ , where  $n$  is the number of iterations and  $\psi$  stands for  $\Psi, \theta, C$ .

To choose the grid size, the grid independence test is performed for  $21 \times 21, 41 \times 41, 61 \times 61, 71 \times 71, 81 \times 81, 91 \times 91$  grid sizes. The grid independence test reveals that the grid size  $81 \times 81$  is sufficient to study in the the bioconvection phenomena.

Table 1: Accuracy test of mean Nusselt numbers with the literature.

| Authors                      | $Ra = 10$ | $Ra = 100$ | $Ra = 1000$ |
|------------------------------|-----------|------------|-------------|
| Varol, Oztop, and Pop [33]   | -         | -          | 13.564      |
| Cross, Bear, and Hickox [12] | -         | -          | 13.470      |
| Manole [22]                  | -         | 3.118      | 13.637      |
| Sheremet and Pop [28]        | 1.079     | 3.115      | 13.667      |
| Present results              | 1.081     | 3.1271     | 13.715      |

## 4 Result and discussion

The present equations (8)–(11) are numerically investigated and analyzed in the porous square cavity filled with nanofluid and gyrotactic microorganisms at different inclination angles. The numerical investigation is carried out with the following parameters considered Rayleigh number ( $Ra = 25$ ), bioconvection Rayleigh number ( $Rb = 15$ ), Lewis number ( $Le = 1$ ), Peclet number, thermophoresis parameter, Brownian parameter, buoyancy ratio parameter, Schmidt number 0.1, Prandtl number 6.9, and inclination angle ( $\delta = 30^\circ$ ) unless when it is mentioned. Streamlines are presented in Figure 2 for various values of inclination angle ( $\delta$ ). When  $\delta = 0^\circ$ , the cells moved in the clockwise direction in the square enclosure. The absolute maximum of the stream function is  $|\Psi_{\max}| = 1.56$ . At  $30^\circ$ , inclination angle the maximum stream function value is  $|\Psi_{\max}| = 2.0435$ , and the rotation of the cell moves in the same direction. The inclination angle increased to  $90^\circ$ , and the velocity of the fluid flow is reduced. At  $120^\circ$ , the intensity of the flow increased to  $|\Psi_{\max}| = 2.0421$ . In this, the cell moves to the center regime with the intensity of the gravitational force. At  $150^\circ$  inclination angle, the flow strength is low and the cell moves near to corners of the left down wall and right top walls. At  $180^\circ$ , the fluid flow velocity is less with the strength of cell  $|\Psi_{\max}| = 1.5629$ .

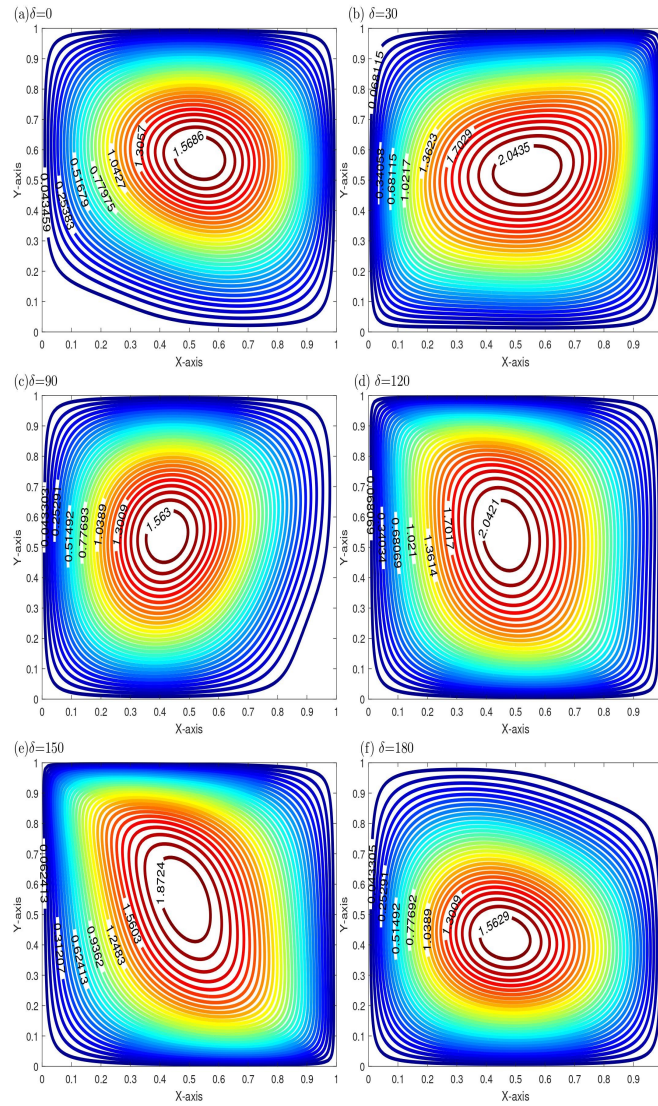
Isotherms are demonstrated in Figure 3 for various inclination angles from  $0^\circ$  to  $180^\circ$ . At inclination  $30^\circ$ , the temperature distribution is indicating the stratified diagonally. At  $180^\circ$ , isotherms are parallel to the vertical walls, which shows the transfer of heat in the mode of heat conduction.

Nanoparticle isoconcentrations of the fluid for various angles are expressed in Figure 4. At  $0^\circ$  to  $30^\circ$  inclination angle, the nanoparticle volume fraction was raised near the bottom cavity wall. When the square enclosure was moved from angle  $90^\circ$  to  $120^\circ$ , the nanoparticle isoconcentration decreased. When the angle is  $120^\circ$  to  $150^\circ$ , again the concentration of nanoparticles increases. When the angle of inclination is inclined from  $150^\circ$  to  $180^\circ$ , the concentration of nanoparticles is decreased. In these all angles, the cell is divided into two different parts: one is formed near the bottom adiabatic wall and the other part is formed as a semi-opened vertex close to the top adiabatic wall.

Figure 5 displays isoconcentrations of microorganisms for the various inclination angles. At  $0^\circ$  to  $30^\circ$  inclination angle, two types of cells are formed: one cell is near the bottom adiabatic wall and the cell moves from the cold wall to the hot wall, the second is an open semi vertex formed at the top cavity wall, and it moves from the hot wall to the cold wall. When the inclination varies from  $0^\circ$  to  $180^\circ$ , the movement of cells in the isoconcentrations of microorganisms and nanoparticle volume fraction is the same. The concentration of microorganisms' maximum value is found at  $\delta = 30^\circ, 120^\circ$ .

In Figure 6, the effect of  $Rb$  and  $Nt$  on average Nusselt number, Sherwood numbers of nanoparticles, and Sherwood number of microorganisms is



Figure 2: Streamlines for the inclination angle  $\delta = 0^\circ - 180^\circ$ 

discussed. Moreover,  $Rb$  increases  $Nu_{avg}$  and  $Sh_{avg}$  from  $0^\circ \leq \delta \leq 90^\circ$ ,  $90^\circ \leq \delta \leq 180^\circ$ , and it reaches high. Indeed, at  $0^\circ$ ,  $90^\circ$ , and  $180^\circ$  the average Nusselt number and average Sherwood number values are low. Also,  $Rb$  increases  $Nn_{avg}$  from  $0^\circ$  to  $180^\circ$ . In addition,  $Nt$  increases  $Nu_{avg}$  from  $0^\circ$  to

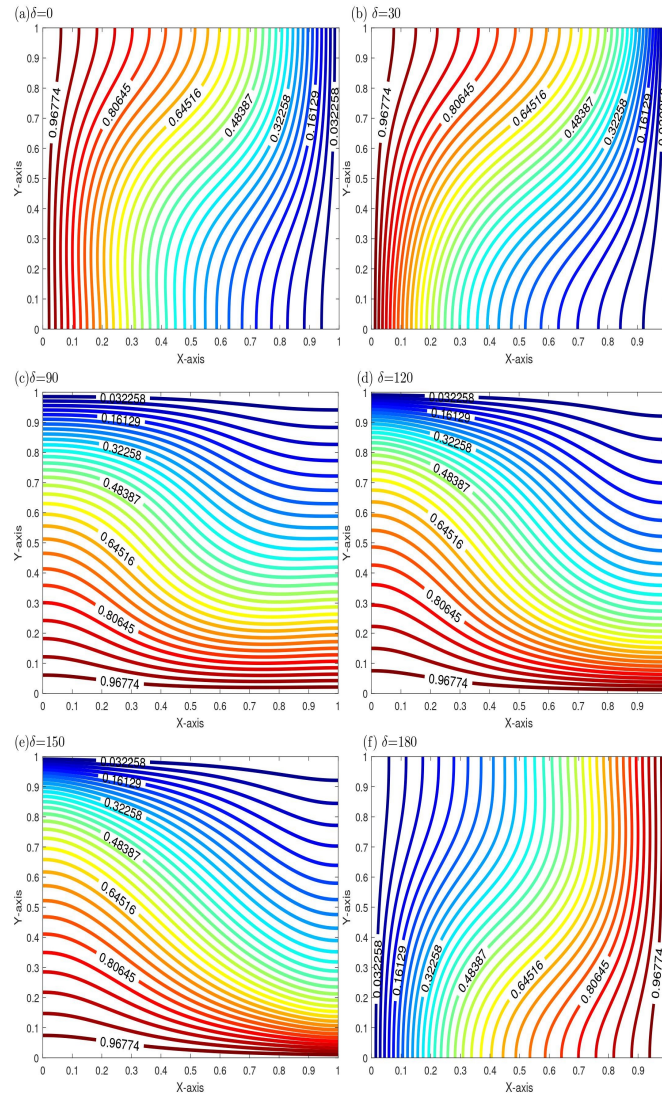
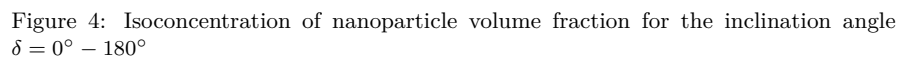


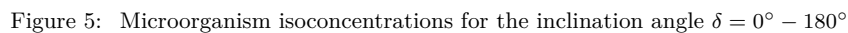
Figure 3: Isotherms for the inclination angle  $\delta = 0^\circ - 180^\circ$

$180^\circ$ , and it shows wavy behavior, and  $Sh_{avg}$  and  $Nn_{avg}$  are also increased with the increase of thermophoresis parameter.



The effect of porous square cavity inclination with the horizontal surface with nanofluid and gyrotactic microorganisms was analyzed with the streamlines,





1. The velocity of the nanofluid flow is high at the angle  $30^\circ$ ,  $120^\circ$  and in the remaining angles, the flow intensity is low.

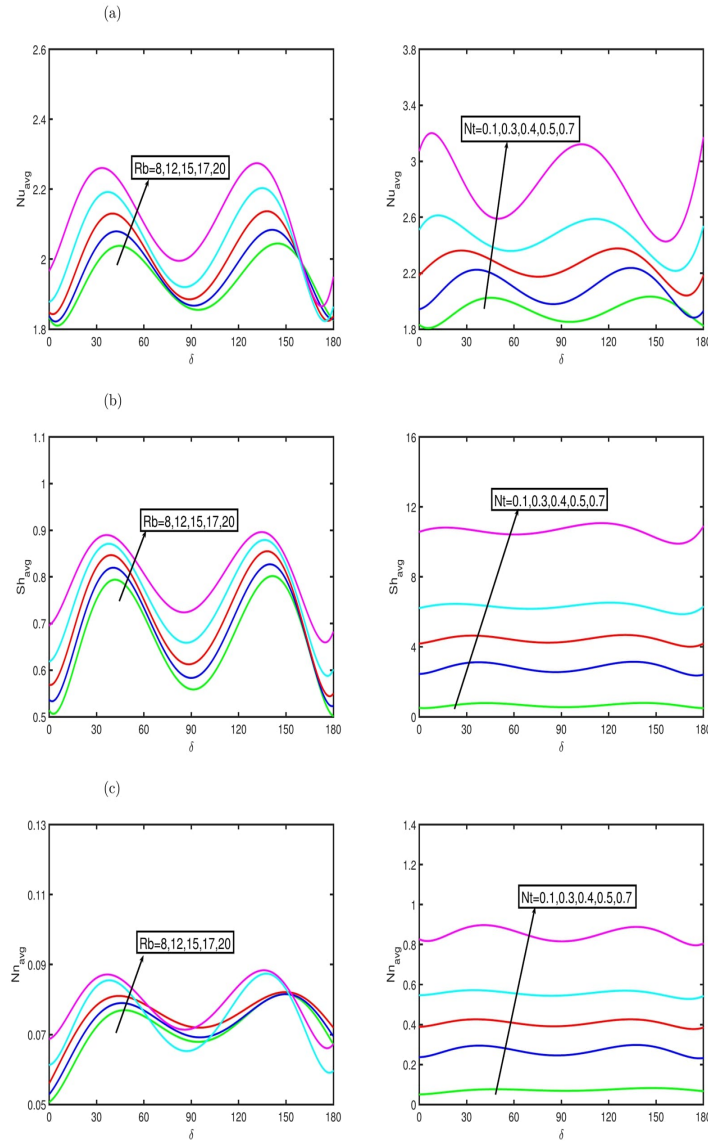


Figure 6: Representation of (a) Average Nusselt number (b) Average nanoparticle Sherwood number (c) Average Microorganism Sherwood number for angle versus  $Rb$  and angle versus  $Nt$

2. The temperature distribution of the nanofluid is affected by the square cavity inclination.

3. Nanoparticle isoconcentration and microorganism isoconcentrations are high at the  $30^\circ$  and  $120^\circ$ , and at the remaining angles, the value is low.

4. Thermophoresis parameter increases  $Nu_{avg}$ ,  $Sh_{avg}$ , and  $Nn_{avg}$  from  $0^\circ \leq \delta \leq 180^\circ$ . Also,  $Nt$  increases  $Nu_{avg}$  from  $0^\circ \leq \delta \leq 180^\circ$ , but at  $0^\circ$  and  $180^\circ$ , the value is low. 5. Bioconvection Rayleigh number increases  $Nu_{avg}$ ,  $Sh_{avg}$ , and  $Nn_{avg}$ .

## Acknowledgements

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

## References

- [1] Aboueian-Jahromi, J., Hossein Nezhad, A. and Behzadmehr, A. *Effects of inclination angle on the steady flow and heat transfer of power-law fluids around a heated inclined square cylinder in a plane channel*, J. Nonnewton Fluid Mech., 166 (23-24) (2011) 1406–1414.
- [2] Alluguvelli, R., Balla, C.S. and Naikoti, K. *Bioconvection in porous square cavity containing oxytactic microorganisms in the presence of viscous dissipation*, Discontinuity, Nonlinearity, and Complexity, 11(02) (2022) 301–313.
- [3] Alluguvelli, R., Balla, C.S., Naikoti, K. and Makinde, O.D. *Nanofluid bioconvection in porous enclosure with viscous dissipation*, Indian J. Pure Appl. Phys. 60(1) (2022) 78–89.
- [4] Aounallah, M., Addad, Y., Benhamadouche, S., Imine, O., Adjloit, L. and Laure, D. *Numerical investigation of turbulent natural convection in an inclined square cavity with a hot wavy wall*, Int. J. Heat Mass Transf. 50(9) (2007) 1683–1693.
- [5] Aziz, A., Khan, W.A. and Pop, I. *Free convection boundary layer flow past a horizontal flat plate embedded in porous medium filled by nanofluid containing gyrotactic microorganisms*, Int. J. Therm. Sci. 56 (2012) 48–57.
- [6] Balla, C.S., Chinthapally Haritha and Kishan, N. *Magnetohydrodynamic double-diffusive convection in fluid saturated inclined porous cavity with thermal radiation and chemical reaction*, J. Chem. Technol. Metall. 53 (2018) 518–536.
- [7] Balla, C.S., Jamuna, B., Krishna Kumari, S.V.H.N. and Rashad, A.M. *Effect of inclination angle on bioconvection in porous square cavity containing gyrotactic microorganisms and nanofluid*, J. Mech. Eng. Sci. 236(9)(2021), 4731–47470.

- [8] Balla, C.S. and Kishan, N. *Soret and Dufour effects on free convective heat and solute transfer in fluid saturated inclined porous cavity*, Int. J. Eng. Sci. Technol. 18(4) (2015) 543–554.
- [9] Balla, C.S., Ramesh, A., Kishan, N. and Rashad, A.M. *Impact of Soret and Dufour on bioconvective flow of nanofluid in porous square cavity*, J. Heat transfer, 50(5) (2021) 5123–5147.
- [10] Bejan, A. *On the boundary layer regime in a vertical enclosure filled with a porous medium*, Lett. Heat Mass Transf. 6(2) (1979) 93–102.
- [11] Choi, S.U.S. and Eastman, J A. *Enhancing thermal conductivity of fluids with nanoparticles*, International mechanical engineering congress and exhibition, San Francisco, CA . United States, 1995.
- [12] Cross, R.J., Bear, M.R. and Hickox, C.E. *The application of flux-corrected transport (FCT) to high Rayleigh number natural convection in a porous medium*. Proceedings of 8th International Heat Transfer Conference, San Francisco (1986).
- [13] Das, S.K., Choi, S.U.S., YU, W., and Pradeep, T. *Nanofluids: Science and Technology*, Mater. Manuf. Process. 24(5) (2009) 600–601.
- [14] Ingham, D.B., Bejan, A., Mamut, E. and Pop, I. *Emerging technologies and techniques in porous media*, 3rd Edition, Kluwer, Dordrecht, 2004, 93–117.
- [15] Ingham, D.B. and Pop, I. *Transport phenomenon in porous media*, Pergamon, Oxford, 1998.
- [16] Izadi, M., Behzadmehr, A. and Shahmardan, M.M. *Effect on inclination angle on mixed convection heat transfer of a nanofluid in a square cavity*, International Journal for Computational Methods in Engineering Science and Mechanics, 16(1) (2015) 11–12.
- [17] Jamuna, B. and Balla, C.S. *Bioconvection in a porous square cavity containing gyrotactic microorganisms under the effects of heat generation/absorption*, Proc. Inst. Mech. Eng. E: J. Process Mech. Eng. 235(5) (2021), 1534–1544.
- [18] Jamuna, B., Mallesh, M.P. and Balla, C.S. *Activation energy process in bioconvection nanofluid flow through porous cavity*, Journal of Porous Media, 25(4) (2022) 37–51.
- [19] Kairi, R.R., Roy, S. and Raut, S. *Thermosolutal Marangoni impact on bioconvection in suspension of Gyrotactic microorganisms over an inclined stretching sheet*, J. Heat Transfer, 143(3) (2021) 031201 (10 pages).

- [20] Kuyper, R.A., Van Der Meer, TH.H., Hoogendoorn, C.J. and Henkes, R.A.W.M. *Numerical study of laminar and turbulent natural convection in an inclined square cavity*, Int. J. Heat Mass Transf. 36 (11)(1993) 2899–2911.
- [21] Kuznetsov, A.V. *Thermo-bioconvection in a suspension of oxytactic bacteria*, Int. Commun. Heat Mass Transf. 32 (8) (2005) 991–999.
- [22] Manole, D.M. *Numerical benchmark results for natural convection in a porous medium cavity*. Heat and Mass Transfer in Porous Media, ASME Conference 1992, vol. 216 (1992) 55–60.
- [23] Minkowycz, W.J., Sparrow, E.M. and Abraham, J.P. *Nanoparticle neat transfer and fluid flow*, CRC Press, New York, 2013.
- [24] Nield, D.A., and Bejan, A. *Convection in porous media*, 4th ed., Springer, New York, 2013.
- [25] Rajarathinam, M. and Nithyadevi ,N. *Heat transfer enhancement of Cu-water nanofluid in an inclined porous square cavity with internal heat generation*, Therm. Sci. Eng. Prog. 4 (2017) 35–44.
- [26] Shenoy, A., Sheremet, M. and Pop, I. *Convective flow and heat transfer from wavy surfaces: Viscous fluids, porous media and nanofluids*, CRC Press, New York, 2016.
- [27] Sheremet, M., Grosan, T. and Pop, I. *MHD free convection flow in an inclined square cavity filled with both nanofluids and gyrotactic microorganisms*, Int. J. Numer. Methods Heat Fluid Flow. 29(12) (2019) 4642–4659.
- [28] Sheremet, M.A. and Pop, I. *Thermobioconvection in a square porous cavity filled by oxy-tactic microorganisms*, Transp. Porous Media, 103 (2014) 191–205.
- [29] Tsai, G.-L., Li, H.-Y. and Lin, C.-C. *Effect of the angle of inclination of a plate shield on the thermal and hydraulic performance of a plate-fin heat sink*, Int. Commun. Heat Mass Transf. 37(4) (2010) 364–371.
- [30] Vadasz, P. *Heat conduction in nanofluid suspensions*, J. Heat Transf. 128(5) (2006) 465–477.
- [31] Vafai, K. *Hand book of Porous media*, 2nd Edition, Taylor and Francis, New York, 2005.
- [32] Varol, Y., Oztop, H. and Koca, A. *Effect of inclination angle on natural convection in composite walled enclosures*, Heat Transf. Eng. 32(1)(2011) 57–68.



- [33] Varol, Y., Oztop, H. and Pop, I. *Influence of inclination angle on buoyancy-driven convection in triangular enclosure filled with a fluid-saturated porous medium*, Heat Mass Transf. 44(5) (2008) 617–624.
- [34] Varol, Y., Oztop, H.F. and Koca, A. *Effects of inclination angle on conduction—natural convection in divided enclosures filled with different fluids*, Int. Commun. Heat Mass Transf. 37(2) (2010) 182–191.

**How to cite this article**

Bodduna, J., Balla, C.S. and Mallesh, M.P., Impact of inclination angle on thermo-bioconvection of nanofluid containing gyrotactic microorganisms saturated in porous square cavity. *Iran. j. numer. anal. optim.*, 2023; 13(1): 141–156. <https://doi.org/10.22067/ijnao.2022.76732.1146>.

## **Aims and scope**

Iranian Journal of Numerical Analysis and Optimization (IJNAO) is published twice a year by the Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad. Papers dealing with different aspects of numerical analysis and optimization, theories and their applications in engineering and industry are considered for publication.

## **Journal Policy**

All submissions to IJNAO are first evaluated by the journal's Editor-in-Chief or one of the journal's Associate Editors for their appropriateness to the scope and objectives of IJNAO. If deemed appropriate, the paper is sent out for review using a single blind process. Manuscripts are reviewed simultaneously by reviewers who are experts in their respective fields. The first review of every manuscript is performed by at least two anonymous referees. Upon the receipt of the referee's reports, the paper is accepted, rejected, or sent back to the author(s) for revision. Revised papers are assigned to an Associate Editor who makes an evaluation of the acceptability of the revision. Based upon the Associate Editor's evaluation, the paper is accepted, rejected, or returned to the author(s) for another revision. The second revision is then evaluated by the Editor-in-Chief, possibly in consultation with the Associate Editor who handled the original paper and the first revision, for a usually final resolution.

The authors can track their submissions and the process of peer review via: <http://ijnao.um.ac.ir>

All manuscripts submitted to IJNAO are tracked by using "iThenticate" for possible plagiarism before acceptance.

## **Instruction for Authors**

The Journal publishes all papers in the fields of numerical analysis and optimization. Articles must be written in English.

All submitted papers will be refereed and the authors may be asked to revise their manuscripts according to the referee's reports. The Editorial Board of the Journal keeps the right to accept or reject the papers for publication.

The papers with more than one authors, should determine the corresponding author. The e-mail address of the corresponding author must appear at the end of the manuscript or as a footnote of the first page.

It is strongly recommended to set up the manuscript by Latex or Tex, using the template provided in the web site of the Journal. Manuscripts should be typed double-spaced with wide margins to provide enough room for editorial remarks.

References should be arranged in alphabetical order by the surname of the first author as examples below:

- [1] Brunner, H. *A survey of recent advances in the numerical treatment of Volterra integral and integro-differential equations*, J. Comput. Appl. Math. 8 (1982), 213-229.
- [2] Stoer, J. and Bulirsch, R. *Introduction to Numerical Analysis*, Springer-verlag, New York, 2002.

# Iranian Journal of Numerical Analysis and Optimization

CONTENTS

Vol. 13, No. 1, pp 1-156, 2023

|                                                                                                                                                          |     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>Applying the meshless Fragile Points method to solve the two-dimensional linear Schrödinger equation on arbitrary domains</b> . . . . .               | 1   |
| D. Haghighi, S. Abbasbandy and E. Shivanian                                                                                                              |     |
| <b>Finding an efficient machine learning predictor for lesser liquid credit default swaps in equity markets</b> . . . . .                                | 19  |
| F. Soleymani                                                                                                                                             |     |
| <b>A modified Liu-Storey scheme for nonlinear systems with an application to image recovery</b> . . . . .                                                | 38  |
| A.I. Kiri, M.Y. Waziri and K. Ahmed                                                                                                                      |     |
| <b>An improvised technique of quintic hermite splines to discretize generalized Burgers–Huxley type equations</b> . . . . .                              | 59  |
| I. Kaur, S. Arora and I. Bala                                                                                                                            |     |
| <b>Generalization of equitable efficiency in multiobjective optimization problems by the direct sum of matrices</b> . . . . .                            | 80  |
| F. Ahmadi, A. R. Salajegheh and D. Foroutannia                                                                                                           |     |
| <b>A family of eight-order interval methods for computing rigorous bounds to the solution to nonlinear equations</b> . . . . .                           | 102 |
| M. Dehghani-Madiseh                                                                                                                                      |     |
| <b>Numerical method for solving fractional Sturm–Liouville eigenvalue problems of order two using Genocchi polynomials</b> . . . . .                     | 121 |
| A. Aghazadeh, Y. Mahmoudi and F. Dastmalchi Saei                                                                                                         |     |
| <b>Impact of inclination angle on thermo-bioconvection of nanofluid containing gyrotactic microorganisms saturated in porous square cavity</b> . . . . . | 141 |
| J. Bodduna, C.S. Balla and M.P. Mallesh                                                                                                                  |     |

web site: <https://ijnao.um.ac.ir>

Email: [ijnao@um.ac.ir](mailto:ijnao@um.ac.ir)

ISSN-Print: 2423-6977

ISSN-Online: 2423-6969