



Iranian Journal of Numerical Analysis and Optimization

Volume 15, Number 4

December 2025

Serial Number: 35

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 15, Number 4, December 2025

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

Publisher: Faculty of Mathematical Sciences, Ferdowsi University of Mashhad

Published by: Ferdowsi University of Mashhad Press

Printing Method: Electronic

Address: Iranian Journal of Numerical Analysis and Optimization

Faculty of Mathematical Sciences, Ferdowsi University of Mashhad

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

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

E-mail: ijnao@um.ac.ir

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

This journal is indexed by:

- [SCOPUS](#)
- [ZbMATH Open](#)
- [ISC](#)
- [DOAJ](#)
- [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 15, Number 4, December 2025

Ferdowsi University of Mashhad - Iran

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.

Ali R. Soheili

Editor-in-Chief

Contents

| | |
|---|-------------|
| A study on efficient chaotic modeling via fixed-memory length fractional Gauss maps | 1310 |
| A. Bellout, R. Bououden, S.E.I. Bouzeraa and M. Berkal | |
| Utilizing the Hybrid approach of the Ramadan group transform and accelerated Adomian method for solving nonlinear integro-differential equations | 1332 |
| M.A. Ramadan, M.M.A. Mansour and H.S. Osheba | |
| Efficient numerical schemes on modified graded mesh for singularly perturbed parabolic convection-diffusion problems | 1361 |
| K.K. Sah | |
| A new exact solution method for bi-level linear fractional problems with multi-valued optimal reaction maps | 1392 |
| F.Y. Feleke and S.M. Kassa | |
| An adaptive scheme for the efficient evaluation of integrals in two-dimensional boundary element method | 1420 |
| R. Si Hadj Mohand, Y. Belkacemi and S. Rechak | |
| Numerical solution of nonlinear diffusion-reaction in porous catalysts using quantum spectral successive linearization method | 1464 |
| S. Abbasbandy | |
| Solving Bratu equations using Bell polynomials and successive differentiation | 1482 |
| N.A. Gezer | |
| Comparative evaluation of large-scale many objective algorithms on complex optimization problems | 1498 |
| R. Chaudhary and A. Prajapati | |
| Combining an interval approach with a heuristic to solve constrained and engineering design problems | 1538 |
| D. Sharma and S.D. Jabeen | |
| A quadrature method for Volterra integral equations of the first kind | 1589 |
| S.A. Hosseini | |
| Nonlinear optimization of revenue per unit of time in discrete Dutch auctions with risk-aware bidders | 1607 |
| R.A. Shamim and M.K. Majahar Ali | |
| On overcoming Dahlquist's second barrier for A-stable linear multistep methods | 1639 |
| G. Hojjati, S. Fazeli and A. Moradi | |

| | |
|--|-------------|
| Portfolio optimization: A mean-variance approach for non-Markovian regime-switching markets | 1658 |
| R. Keykhai | |
| Approximation of functions in Hölder's class and solution of nonlinear Lane–Emden differential equation by orthonormal Euler wavelets | 1688 |
| H.C. Yadav, A. Yadav and S. Lal | |
| Mathematical modeling of an optimal control problem for combined chemotherapy and anti-angiogenic cancer treatment protocols | 1710 |
| Y.A. Mahaman Nouri and S. Bisso | |



A study on efficient chaotic modeling via fixed-memory length fractional Gauss maps

A. Bellout, R. Bououden, S.E.I. Bouzeraa and M. Berkal*, 

Abstract

*Corresponding author

Received 19 May 2025; revised 17 June 2025; accepted 9 July 2025

Aida Bellout

Laboratory of Mathematics and their Interactions, Department of Mathematics, Abdelhafid Boussouf University Center, Algeria. e-mail: a.bellout@centre-univ-mila.dz

Rabah Bououden

Laboratory of Mathematics and their Interactions, Department of Mathematics, Abdelhafid Boussouf University Center, Algeria.

Department of Applied Mathematics, Abdelhafid Boussouf University Center, Mila ,R.P 26, Mila, 43000, Algeria. e-mail: r.bouden@centre-univ-mila.dz

Seyf El Islam Bouzeraa

Department of Applied Mathematics, Abdelhafid Boussouf University Center, Mila ,R.P 26, Mila, 43000, Algeria. e-mail: s.bouzeraa@centre-univ-mila.dz

Messaoud Berkal

Department of Applied Mathematics, Abdelhafid Boussouf University Center, Mila ,R.P 26, Mila, 43000, Algeria. e-mail: m.berkal@centre-univ-mila.dz

How to cite this article

Bellout, A., Bououden, R., Bouzeraa, S.E.I. and Berkal, M., A study on efficient chaotic modeling via fixed-memory length fractional Gauss maps. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1310-1331. <https://doi.org/10.22067/ijnao.2025.93606.1650>

This paper investigates the dynamic behavior of the fractional Gauss map with fixed memory length, highlighting its potential for efficient chaotic modeling. Unlike classical fractional systems that require the full history of states, the proposed approach introduces a memory-limited version, significantly reducing computational cost while preserving complex dynamical features. Through bifurcation analysis, Lyapunov exponents, and the $0 - 1$ test for chaos, the study demonstrates that the system exhibits a rich variety of behaviors, including periodic, quasi-periodic, and chaotic regimes, depending on the fractional order and memory size. A comparative evaluation with the classical Gauss map reveals that the fixed-memory model retains similar chaotic characteristics, but with improved computational efficiency. These findings suggest that fixed-memory fractional maps offer a practical alternative for simulating chaotic systems in real-time applications.

AMS subject classifications (2020): Primary 39A33; Secondary 37D45, 37N30.

Keywords: Chaos; Fractional difference equations; Gauss map; Fixed memory length; Bifurcation; Lyapunov exponent.

1 Introduction

Chaotic systems play a fundamental role in the modeling and analysis of non-linear dynamics across various disciplines, including mathematics, physics, biology, engineering and optimization [3, 4, 5, 7, 6, 8, 11, 10, 12, 13, 14, 15, 24, 26, 27, 25]. During this period, several chaotic discrete systems have been proposed, such as the Logistic map, Tent map, Gauss map, Hénon map, and Lozi map [21, 23, 28, 35, 37, 39].

The Gauss map, in particular, has been widely studied due to its rich and sensitive dependence on initial conditions, making it a valuable tool for exploring chaotic behavior in discrete systems [39]. In parallel, the theory of fractional-order systems has gained increasing attention over the past two decades as an effective framework for modeling systems with memory and hereditary properties [17, 16, 31, 32, 41]. Fractional difference equations provide a generalization of traditional difference equations, allowing the present state to depend on all past states with power-law weighting.

However, a significant limitation of classical fractional models lies in their requirement to store and process the entire history of the system, which can lead to high computational and memory costs. This challenge becomes especially critical in real-time simulations or large-scale systems [16, 30, 33, 41]. To overcome this issue, researchers have introduced fractional systems with fixed memory length, where only a finite number of past states contribute to the current state. This approach reduces computational complexity while preserving key dynamic features of the system [2, 18, 19]. Despite its potential, there is still a lack of comprehensive studies exploring how memory truncation affects the long-term behavior of chaotic fractional maps, particularly in comparison with both classical (nonfractional) and full-memory fractional systems.

To address these limitations, this study employs a fixed memory length approach, which restricts the influence of past states to a finite window. This strategy not only reduces computational complexity but also reflects practical constraints encountered in real-world systems. In many applications, such as real-time control systems, embedded cryptographic protocols, and biological modeling, memory and computational resources are severely limited [9, 22, 36]. For instance, control algorithms deployed on microcontrollers must operate under strict timing and memory constraints, while cryptographic systems benefit from low-latency and lightweight implementations. Similarly, in biological modeling, such as simulating cardiac activity, it is often reasonable to assume that only a limited history influences the current state due to physiological time scales. Fixed-memory fractional models thus provide a realistic and efficient alternative, balancing dynamical richness with feasibility.

This paper addresses this gap by investigating the dynamic behavior of the fractional Gauss map with fixed memory length. We employ several tools—such as bifurcation diagrams, Lyapunov exponents, and the $0 - 1$ test to analyze the system's response under various parameter values. Additionally, we perform a comparative analysis with the classical Gauss map to evaluate the impact of fixed memory on both chaos and computational performance. The results demonstrate that the proposed model maintains the richness of chaotic dynamics while achieving improved computational efficiency, making

it a promising approach for practical applications where memory and speed are critical factors.

2 Fractional discrete-time calculus

Fractional discrete-time calculus extends classical difference equations by allowing the order of differencing to be noninteger (fractional), which makes it suitable for modeling systems with memory and hereditary properties. This is particularly useful in fields where past states influence current behavior in a gradual and persistent manner. One of the key operators used in this context is the Caputo-like delta fractional difference, introduced in [1, Definition 13]. This operator is a discrete analog of the Caputo derivative from continuous fractional calculus, and it is defined in a way that aligns naturally with initial conditions, making it more convenient for modeling real-world systems.

Let $q \in \mathbb{R}$ be fixed and let $\mathbb{N}_q = \{q, q+1, q+2, \dots\}$ denote the isolated time scale. For the function $u(n)$, the delta difference operator Δ is defined as follows:

$$\Delta u(n) = u(n+1) - u(n). \quad (1)$$

Definition 1. [38]

Let $u : \mathbb{N}_q \rightarrow \mathbb{R}$ and $v > 0$. Then the fractional sum of order v is defined by

$$\Delta_q^{-v} u(t) = \frac{1}{\Gamma(v)} \sum_{s=q}^{t-v} (t - \sigma(s))^{(v-1)} u(s), \quad t \in \mathbb{N}_{q+v}, \quad (2)$$

where q is the starting point, $\sigma(s) = s+1$ and t^v is the falling function defined in terms of the Gamma function as

$$t^{(v)} = \frac{\Gamma(t+1)}{\Gamma(t+1-v)}. \quad (3)$$

Definition 2. [1]

For $v > 0$, $v \notin \mathbb{N}$ and $u(t)$ defined on \mathbb{N}_q , the Caputo-like delta difference is defined by

$${}^c \Delta_q^v u(t) = \Delta_q^{-(m-v)} \Delta^m u(t), \quad (4)$$

$$= \frac{1}{\Gamma(m-v)} \sum_{s=q}^{t-(m-v)} (t-\sigma(s))^{(m-v-1)}, \Delta_s^m u(s), \quad (5)$$

where $t \in \mathbb{N}_{q+m-v}$ and $m = [v] + 1$.

Here, $\sigma(s) = s + 1$ is the forward jump operator commonly used on discrete time scales. It defines the next point in the discrete domain and ensures that the summation aligns with the forward-shifted indices. This choice is standard in delta-type fractional calculus and reflects the progression of discrete time by one step. The term $(t-\sigma(s))^{(m-v-1)}$ represents the falling factorial kernel, which weights recent values more heavily than older ones, capturing the memory effect inherent in fractional systems.

Theorem 1 ([20]). For the delta fractional difference equation

$$\begin{cases} [c]c^c \Delta_q^v u(t) = f(t+v-1, u(t+v-1)), \\ \Delta^k u(q) = u_k, m = [v] + 1, \end{cases} \quad k = 0, \dots, m-1,$$

the equivalent discrete integral equation can be obtained as

$$u(t) = u_0(t) + \frac{1}{\Gamma(v)} \sum_{s=q+m-v}^{t-v} (t-\sigma(s))^{(v-1)} \times f(s+v-1, u(s+v-1)), \quad t \in \mathbb{N}_{q+m}, \quad (6)$$

where

$$u_0(t) = \sum_{k=0}^{m-1} \frac{(t-q)^{(q)}}{k!} \Delta^k u(q). \quad (7)$$

This operator calculates a fractional change in $u(t)$, but instead of only using current and previous values (as in classical differences), it uses a weighted sum of past changes, giving more weight to recent values and less to older ones. This model is systems where recent history has a stronger effect, but the influence of earlier states still persists.

The Caputo-like delta fractional difference is chosen in this work due to several theoretical and practical advantages over other discrete fractional operators, such as the Riemann–Liouville type or Grünwald–Letnikov formulations.

First, the Caputo-like formulation allows for the use of initial conditions in the same form as those used in classical integer-order systems. This makes it more suitable for physical and biological modeling, where initial values often have direct physiological interpretations.

Second, the Caputo-like operator naturally accommodates memory effects by incorporating a weighted history of the system states, while still preserving computational tractability due to its structured definition.

Importantly, in the context of biological systems such as cardiac models, memory plays a crucial role in capturing physiological phenomena. The electrical activity of the heart, for example, is influenced not only by the current stimulus but also by a history of past activations and recovery processes. Fractional models have been shown to better replicate such long-range dependence in excitable tissues compared to their integer-order counterparts [36, 40].

The Caputo-like operator is particularly advantageous here because it reflects this hereditary behavior while maintaining a clear relationship with classical dynamics. This balance between interpretability, memory fidelity, and numerical implementation makes the Caputo-like fractional difference a compelling choice for modeling complex, memory-dependent systems like cardiac cells.

3 Fractional Gauss map with fixed memory length

In mathematics, the Gauss map, also referred to as the Gaussian map [29], is a nonlinear iterated mapping that transforms real numbers into a real interval using the Gaussian function defined as follows:

$$x_{n+1} = \exp(-ax_n^2) + b, \quad (8)$$

where a and b are bifurcation parameters.

This map can exhibit chaotic behavior, for example, when $a = 7.5$ and $b = -0.6$. This map is also known as the mouse map due to its bifurcation diagram when $a = 7.5$ and b in the range -1 to 1 resembling a mouse as in

Figure 1.

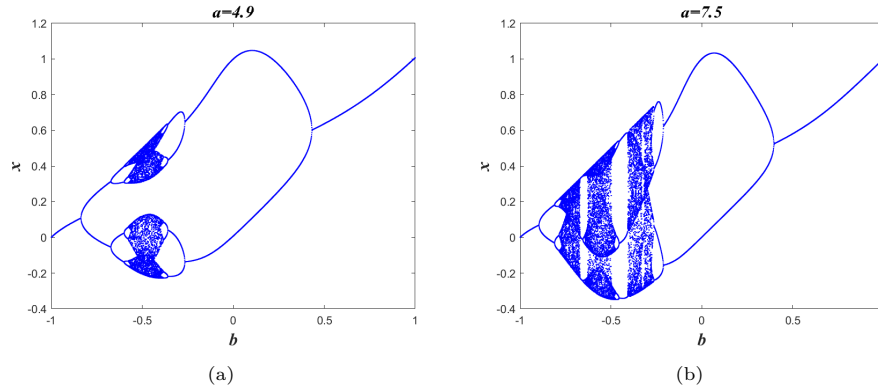
Figure 1: Bifurcation diagram of map (8) with $x_0 = 0$ and b in the range -1 to 1 , (a) $a = 4.9$ and (b) $a = 7.5$

Figure 1 shows the bifurcation diagram of the fractional Gauss map (8) for two different sets of parameters: The first set is $a = 4.9$ and b in the range -1 to 1 ; the second set is $a = 7.5$ and b in the range -1 to 1 .

The first-order difference of Gauss map can be easily expressed as

$$\Delta x_n = \exp(-ax_n^2) + b - x_n. \quad (9)$$

In discrete fractional calculus, the fractional Gauss map can be defined as

$${}^c\Delta_q^v x_n = \exp(-ax(t-1+v)^2) + b - x(t-1+v), \quad (10)$$

where ${}^c\Delta_q^v$ is the fractional difference of Caputo and $0 < v \leq 1$ is the difference order. For the Gauss map (10), an explicit numerical solution can be given by

$$x_n = x_0 + \frac{1}{\Gamma(v)} \sum_{j=1}^n \frac{\Gamma(n-j+v)}{\Gamma(n-j+1)} (\exp(-ax(j-1)^2) + b - x(j-1)), \quad (11)$$

where x_0 is initial condition.

For $v = 1$, the discrete fractional map (11) simplifies to the classical map (8). Unlike the integer order map (8), the fractional map (11) includes

a discrete kernel function that relies on past information x_0, x_1, \dots, x_{n-1} . Consequently, the memory effects in these discrete maps imply that their current state of evolution depends on all previous states. Figure 2 shows

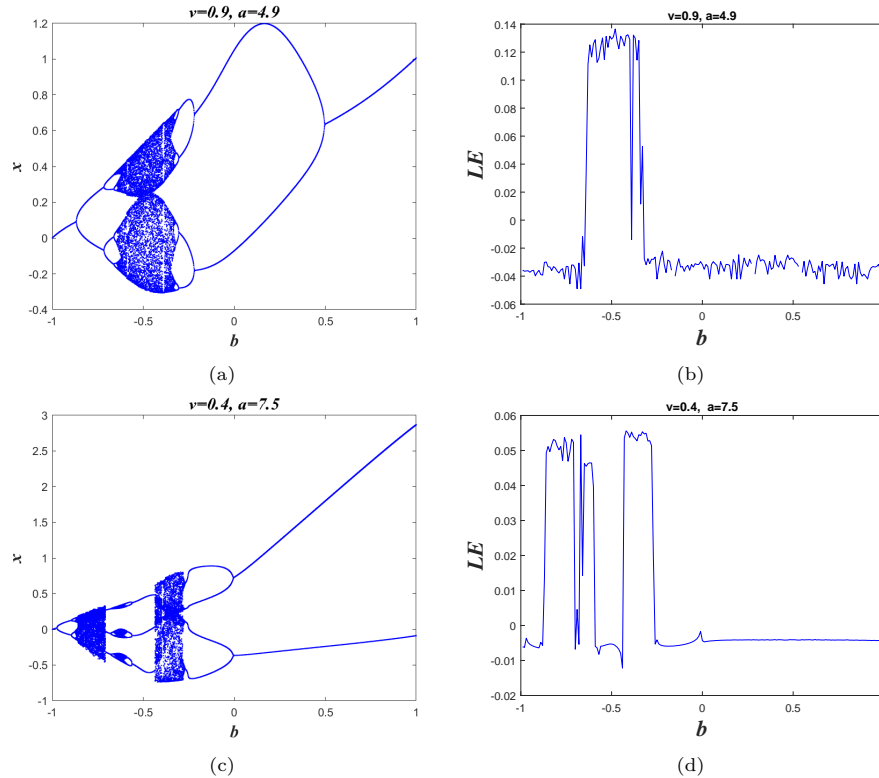


Figure 2: (a) Bifurcation diagram of fractional Gauss map (11) for $v = 0.9$, $x_0 = 0$, b in the range -1 to 1 and $a = 4.9$, (b) the greatest Lyapunov exponent of fractional Gauss map for $v = 0.9$, $x_0 = 0$, b in the range -1 to 1 and $a = 4.9$, (c) Bifurcation diagram of fractional Gauss map (11) for $v = 0.4$, $x_0 = 0$, b in the range -1 to 1 and $a = 7.5$, (d) the greatest Lyapunov exponent of fractional Gauss map for $v = 0.4$, $x_0 = 0$, b in the range -1 to 1 and $a = 7.5$.

the bifurcation diagram and Lyapunov exponent of the fractional Gauss map (11) for two different sets of parameters: The first set is $a = 4.9$, $v = 0.9$, and b in the range -1 to 1 ; the second set is $a = 7.5$, $v = 0.4$, and b in the range -1 to 1 .

For example, for $v = 0.4$ and $a = 7.5$, we note that the map (11) converges to *period* -1 orbit for $-1 \leq b < -0.97$. The first bifurcation occurs when $b = -0.97$, transitioning from a fixed point to a *period* -2 orbit via period-

doubling bifurcation. Figure 2(d) confirms this information because the Lyapunov exponent is negative over the interval $-1 \leq b < -0.97$ and becomes zero when $b = -0.97$. The map maintains the same behavior until $b = -0.89$, where the second bifurcation occurs, transitioning from a *period* – 2 orbit to a *period* – 4 orbit via period-doubling bifurcation. When $b > -0.87$, the Lyapunov exponent becomes positive, indicating that the map has become chaotic over the interval $-0.87 \leq b < -0.7$; this is confirmed by Figures 2(d) and 2(c). When $-0.7 \leq b < -0.67$, the map converges to a *period* – 3 orbit, which further confirms that the map is chaotic for certain values of b [34]. Another bifurcation occurs when $b = -0.67$, transitioning from a *period* – 3 orbit to a *period* – 6 orbit via period-doubling bifurcation. Another return to chaotic behavior of the map (11) from point $b = -0.66$ to point $b = -0.59$ as shown in Figure 2(c). Then, as $-0.59 \leq b < -0.56$, the map (11) converges to *period* – 6 orbit for the second time. After that, the map (11) converges again to *period* – 3 orbit in the interval $-0.56 \leq b < -0.43$. Again, in the interval $-0.43 \leq b < -0.27$, map (11) has chaotic behavior. On the interval $-0.27 \leq b < 0$, map (11) converges to *period* – 4 orbit again. Finally, as $0 \leq b \leq 1$, map (11) converges to *period* – 2 orbit for the second time.

It should be noted that the figure presenting the Lyapunov exponent of the map (11) confirms all previously mentioned results, where the Lyapunov exponent is negative when the orbit of map (11) converges towards a periodic orbit, is zero at the bifurcation point, and positive when the trajectory converges to chaotic behavior.

In classical fractional-order models, the system exhibits infinite memory, where all past states influence the present dynamics with a decaying weight. While this feature captures hereditary effects accurately, it leads to high computational cost and memory storage requirements, especially in long-term simulations.

To address these limitations, this study employs a fixed memory length approach, which restricts the influence of the past to a finite window of previous steps. This simplification not only reduces computational complexity but also reflects a more realistic assumption in many physical and biological systems where distant past events have negligible influence.

Moreover, the use of fixed memory enhances numerical stability and implementation efficiency, making it more suitable for real-time applications or hardware-constrained systems. In contrast, infinite memory schemes may suffer from accumulating numerical errors and impractical memory demands over long simulation horizons.

Therefore, the adoption of a fixed-memory fractional Gauss map strikes a balance between capturing essential memory effects and maintaining tractable, efficient simulations. This feature is particularly advantageous in chaotic modeling, where fast computation and sensitivity to initial conditions are critical. The following equation defines the fractional Gauss map with fixed memory length:

$$\begin{cases} x_n = x_0 + \frac{1}{\Gamma(v)} \sum_{j=1}^n \frac{\Gamma(n-j+v)}{\Gamma(n-j+1)} (\exp(-ax(j-1)^2) + b - x(j-1)) & \text{if } n \leq L, \\ x_n = \frac{1}{\Gamma(v)} \sum_{j=n-L}^n \frac{\Gamma(n-j+v)}{\Gamma(n-j+1)} (\exp(-ax(j-1)^2) + b - x(j-1)) & \text{else,} \end{cases} \quad (12)$$

where L is the length of the memory.

In classical fractional systems, the entire past state history contributes to

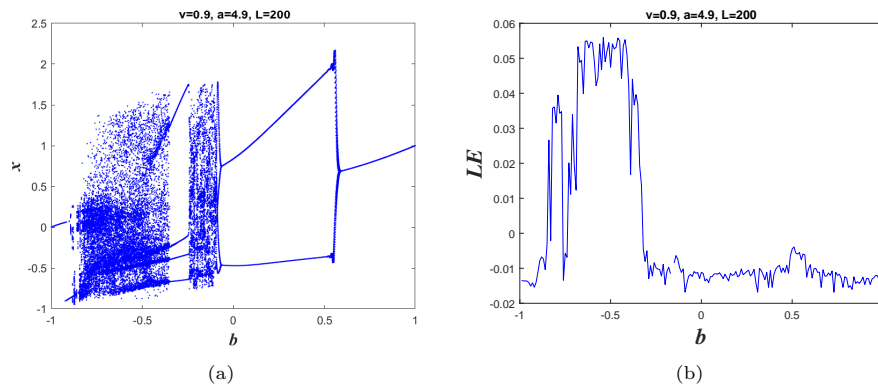


Figure 3: (a) Bifurcation diagram of the fractional Gauss map with fixed memory length for $L = 200$, $v = 0.9$, $x_0 = 0$, $a = 4.9$ and b in the range -1 to 1 , (b) The greatest Lyapunov exponent for $L = 200$, $v = 0.9$, $x_0 = 0$, $a = 4.9$ and b in the range -1 to 1 .

the current state, with memory effects governed by a power-law kernel. This long-term memory is central to capturing hereditary and complex dynamics. However, it comes at the cost of high computational demands and sensitivity

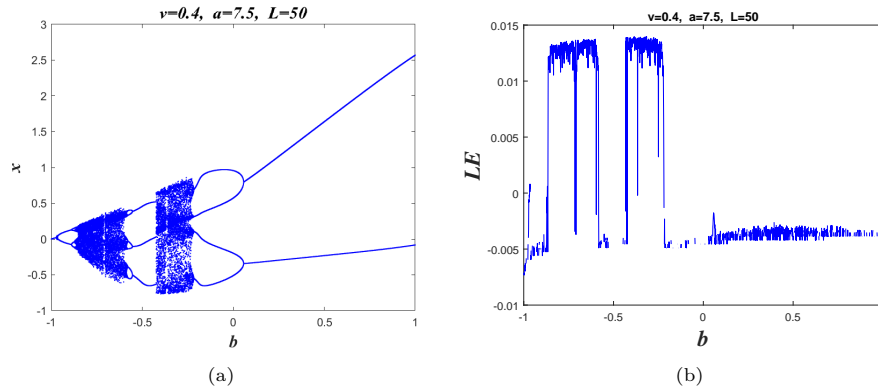


Figure 4: (a) Bifurcation diagram of the fractional Gauss map with fixed memory length for $L = 50$, $v = 0.4$, $x_0 = 0$, $a = 7.5$ and b in the range -1 to 1 , (b) The greatest Lyapunov exponent for $L = 50$, $v = 0.4$, $x_0 = 0$, $a = 7.5$ and b in the range -1 to 1 .

to numerical errors over long simulations. The introduction of *fixed memory length*, denoted by L , truncates the influence of past states to only the most recent L iterations. This simplification raises fundamental questions about how such truncation affects the nature of chaos.

Qualitatively, memory truncation can **dampen long-range correlations**, potentially reducing the depth of chaotic complexity or altering the sequence of bifurcations. However, as demonstrated in Figures 3 and 4, the fixed-memory fractional Gauss map continues to exhibit hallmark features of chaotic systems (including period-doubling routes to chaos, windows of periodicity, and regions of positive Lyapunov exponents) despite having a finite memory. The preservation of these structures suggests that essential nonlinear behavior remains intact even when older history is ignored.

Quantitatively, our simulations show that when comparing the full-memory fractional map (11) with the fixed-memory version (12), the **critical bifurcation points and ranges of chaotic behavior shift only slightly**, and the values of the largest Lyapunov exponents remain within comparable ranges.

Moreover, the numerical gain is substantial. Table 1 illustrates that computation time is reduced by up to 99% when memory is fixed, without sacrificing the model's ability to simulate chaos. This trade-off between **com-**

putational efficiency and memory fidelity is favorable in real-time or embedded systems, where resource constraints are strict.

Table 1: The time required to obtain the path of the fractional Gauss map (FGM) and the fractional gauss map with fixed memory length (FGMFML) when $L = 50$.

| Number of iteration | FGM | FGMFML |
|---------------------|--------------|----------|
| 1000 | 0.920305 | 0.205349 |
| 5000 | 91.580199 | 0.264022 |
| 10000 | 732.813417 | 0.267275 |
| 20000 | 13121.494701 | 0.304672 |

Figures 3 and 4 show the bifurcation diagram and Lyapunov exponent of a fractional Gauss map with fixed memory length and the greatest Lyapunov exponent for two different sets of parameters: The first set is $a = 4.9$, $v = 0.9$, $L = 200$ and b in the range -1 to 1 ; the second set is $a = 7.5$, $v = 0.4$, $L = 50$ and b in the range -1 to 1 .

When comparing Figure 2, which represents the bifurcation diagram of the fractional Gauss map, with Figure 4, which represents the bifurcation diagram of the fractional Gauss map with a fixed memory length, we obtain the following observations and results:

Every feature included in the fractional order bifurcation diagram can additionally be seen in fractional order with fixed memory length bifurcation diagrams.

We can see an increase in the period of the bifurcations, which results in chaos in all bifurcation diagrams.

From the bifurcation diagram shown in Figure 4 for the fractional Gauss map with a fixed memory length $L = 50$ we observe if $-1 \leq b < -0.97$ the map (12) converges to *period* -1 orbit. The first bifurcation occurs when $b = -0.97$, transitioning from a fixed point to a *period* -2 orbit via period-doubling bifurcation. Figure 4(b) confirms this information because the Lyapunov exponent is negative over the interval $-1 \leq b < -0.97$ and becomes zero when $b = -0.97$. The map maintains the same behavior until $b = -0.88$, where the second bifurcation occurs, transitioning from a *period* -2 orbit to a *period* -4 orbit via period-doubling bifurcation. When $-0.88 < b < -0.58$, the Lyapunov exponent becomes positive, indicating that the map has become chaotic over the interval $-0.88 < b < -0.58$; this is

confirmed by Figures 4(b) and 4(a). When $-0.58 \leq b < -0.55$, the map converges to a *period* – 6 orbit, which further confirms that the map is chaotic for certain values of b . Another bifurcation occurs when $b = -0.55$, transitioning from a *period* – 6 orbit to a *period* – 3 orbit. Another return to chaotic behavior of the map (12) from point $b = -0.42$ to point $b = -0.21$ as shown in Figure 4(a). Then, as $-0.21 \leq b < -0.06$ the map (12) converges to *period* – 4 orbit for the second time. After that, the map (12) converges again to *period* – 2 orbit in the interval $0.06 \leq b < 1$.

The figure displaying the Lyapunov exponent for map (12) validates all prior conclusions. It demonstrates that the Lyapunov exponent is negative when the orbit of map (12) approaches a periodic state, zero at the bifurcation point, and positive when the trajectory transitions into chaotic behavior.

The parameter values used in our simulations, such as $a = 4.9$, $a = 7.5$, and $b \in [-1, 1]$, are selected based on their well-known ability to produce rich dynamical behaviors in the classical Gauss map. The fractional orders $v = 0.4$ and $v = 0.9$ were chosen to compare strong memory effects versus near-integer behavior. The memory lengths $L = 50$ and $L = 200$ were used to evaluate the impact of truncation while maintaining low computational cost.

By comparing the bifurcation diagrams of the fractional Gauss map and the fractional Gauss map with fixed memory length, we can conclude that the length of the memory has a big effect on the dynamics of the map.

We mentioned in the first section that the numerical calculation of the discrete fraction system is very time-consuming compared to the numerical calculation of the discrete fraction system with a fixed memory length, and this is what we will prove in this part through the table that summarizes the results of the time taken to obtain the path of the fractional Gauss map and the fractional Gauss map with a fixed memory length, and we used the MATLAB program and ran it on an i5 processor, 2.40GHz with 16G of RAM (random access memory).

Table 1 compares the computation time required by the fractional Gauss map with two memory strategies: Infinite memory and fixed memory length. The results clearly indicate that the fixed-memory approach significantly reduces the computational burden.

This improvement becomes more pronounced as the simulation horizon increases, highlighting the scalability of the proposed method. By limiting the number of past states involved in each iteration, the fixed-memory model avoids redundant calculations while still capturing the essential memory dynamics of the system.

Such a reduction in execution time is crucial for real-time applications and long-term simulations, especially in hardware-constrained environments or large-scale systems. Therefore, the use of fixed memory not only enhances numerical efficiency but also makes the fractional modeling of chaotic systems more practical and accessible.

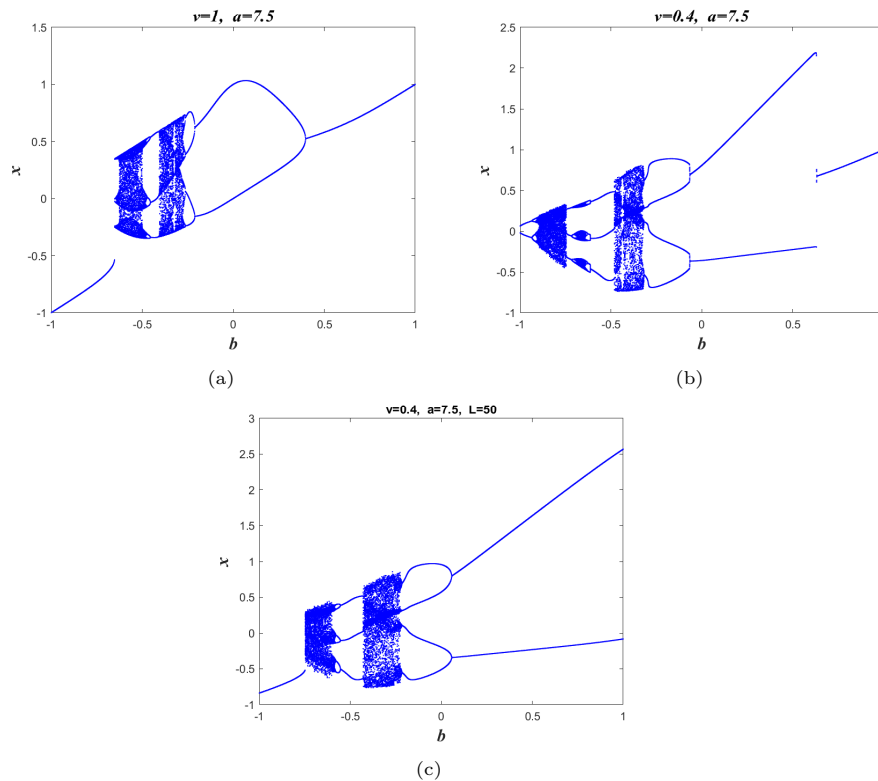


Figure 5: Bifurcation diagram of fractional gauss map with initial condition as $x_0 = 0.9$ and b in the range -1 to 1 where, (a) for $v = 1$, (b) for $v = 0.4$ and (c) for $v = 0.4$ and fixed memory length $L = 50$.

Form Figure 5, we noted that when the initial state was changed, there was a rapid interruption in the bifurcation diagrams for of integer order and

fractional order. This is also what was observed in bifurcation diagram of fractional map with fixed memory length (Figure 5) when changing the initial value from $x_0 = 0$ to $x_0 = 0.9$.

4 Regular and chaotic behavior of a fractional Gauss map with fixed memory length

In 2003, Gatwald and Melbourne introduced the 0 – 1 test to prove the existence of chaotic behavior in nonlinear deterministic systems. The 0 – 1 test is applied to a direct time series and provides a binary-like outcome: A value of $K \approx 0$ indicates regular (nonchaotic) dynamics, while a value of $K \approx 1$ is a strong indicator of chaos. This allows for a clear quantitative distinction between regular and chaotic behavior. The 0 – 1 test is applied to a direct time series. The 1 – 0 test can determine the behavior of a given sequence from the dynamics of trajectories p_c and q_c , where a dynamical system is chaotic if the behavior of the trajectories is Brownian motion (k approaches 1), while a dynamical system is regular if the motion is finite (k approaches from 0). For $c \in [0, \pi]$, We have q_c and p_c determined as follows:

$$p_c(n) = \sum_{j=1}^n x(j) \cos(jc), \quad (13)$$

$$q_c(n) = \sum_{j=1}^n x(j) \sin(jc). \quad (14)$$

The average square displacement $M_c(n)$ of both variables $p_c(n)$ and $q_c(n)$ is calculated from the following relationship:

$$M_c(n) = \frac{1}{N} \sum_{j=1}^N (p_c(j+n) - p_c(j))^2 + (q_c(j+n) - q_c(j))^2. \quad (15)$$

Finally, we calculate the asymptotic growth rate K using the correlation method, where

$$K = \text{median}(k_c). \quad (16)$$

Also, we have k_c defined by the following relation:

$$k_c = \frac{\text{cov}(\xi, \Delta)}{\sqrt{\text{var}(\xi)\text{var}(\Delta)}} \in [-1, 1], \quad (17)$$

where $\xi = (1, 2, \dots, n_{cut})$, $\Delta = (M_n(1), M_n(2), \dots, M_n(cut))$ and $n_{cut} = \text{round}(N/10)$. Moreover, $D_c(n)$ is the adjusted average square displacement. It has been defined as follows:

$$D_c(n) = M_c(n) - (E(\Phi(x_j)))^2 \frac{1 - \cos(nc)}{1 - \cos(c)}, \quad (18)$$

where the average $E(\Phi)$ is given by

$$E(\Phi(x_j)) = \lim_{N \rightarrow +\infty} \frac{1}{N} \sum_{j=1}^N \Phi(x_j). \quad (19)$$

Figure 6 represents the result of the 0 – 1 test for the fractional Gauss map where $v = 0.4$, $L = 50$ with $x_0 = 0$.

We conducted a 0 – 1 test at two different values of b . When $b = 0$ the trajectories of p_c and q_c in the $(p_c - q_c)$ plane present bounded (see Figure 6(a)) We also note that the value of k is very close to 0 (see Figure 6(b)), and this translates into a fractional Gauss map with a fixed memory length that non chaotic behavior. When $b = -0.7$ we note the trajectories of p_c and q_c in the $(p_c - q_c)$ plane, similar to Brownian motion (Figure 6 (c)). We also note that the value of k is very close to 1 (see Figure 6(d)), and this translates into a fractional Gauss map with a fixed memory length that has chaotic behavior.

5 Conclusion

In this study, we proposed a fractional Gauss map with fixed memory length to efficiently model chaotic dynamics. The results demonstrate that the fixed-memory approach can significantly reduce computation time while maintaining the essential features of chaotic behavior.

Unlike traditional infinite-memory fractional systems, the proposed model achieves a balance between memory representation and numerical efficiency, making it more suitable for real-time or large-scale applications.

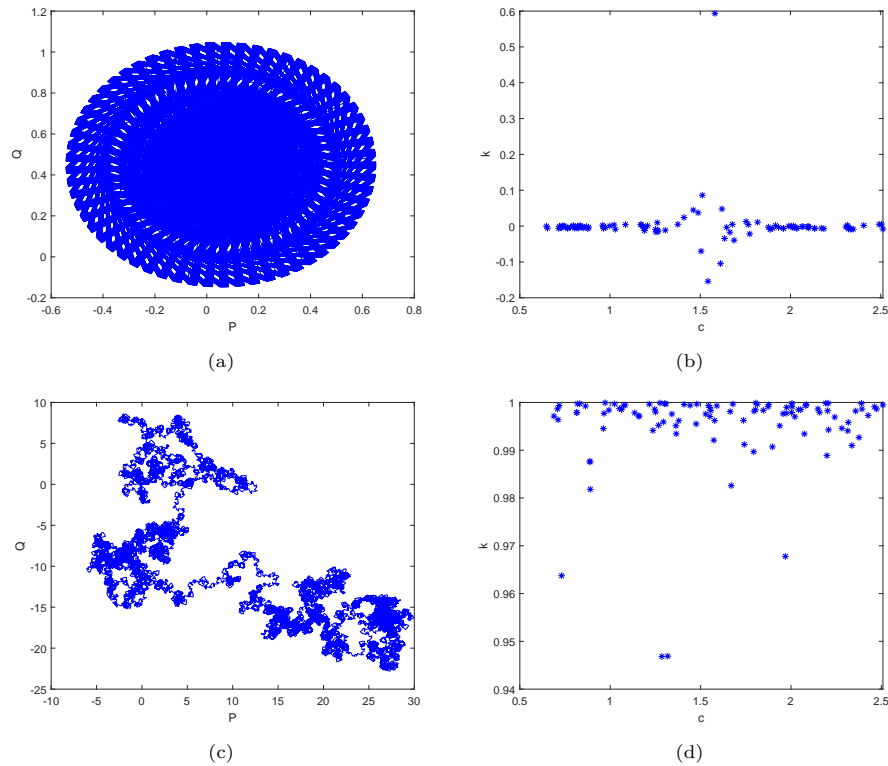


Figure 6: The 0 – 1 test of fractional Gauss map with fixed memory length $L = 50$, $v = 0.4$, $x_0 = 0$, (a) (b) for $b = 0$ and (c),(d) for $b = -0.7$.

However, the current study is limited to specific types of chaotic maps and a fixed memory structure. The influence of varying memory lengths, the stability of long-term dynamics, and the accuracy trade-offs require further investigation.

Future work will focus on extending this framework to other fractional maps, exploring adaptive memory strategies, and applying the model to real-world systems such as biological or economic time series. Additionally, more rigorous theoretical analysis of the stability and convergence properties of the fixed-memory fractional model would enhance its mathematical foundation and applicability.

Declarations

Conflicts of Interest: The authors declare no conflict of interest.

Acknowledgements

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

References

- [1] Abdeljawad, T. *On Riemann and Caputo fractional difference*, Comput. Math. Appl., 62 (2011), 1602–1611.
- [2] Abdelouahab, M.S. and Hamri, N. *The Grünwald–Letnikov fractional-order derivative with fixed memory length*, Mediterr. J. Math., 13(2) (2016), 557–572.
- [3] Abdlaziz, M.A.M., Ismail, A. I., Abdullah, F.A. and Mohd, H.M. *Bifurcations and chaos in a discrete SI epidemic model with fractional order*, Adv. Differ. Equ., 2018 (2018), 1–19.
- [4] Almatrafi, M.B. and Berkal, M. *Stability and bifurcation analysis of predator-prey model with Allee effect using conformable derivatives*, J. Math. Comput. Sci., 36(3) (2025), 299–316.
- [5] Baleanu, D., Jajarmi, A., Defterli, O., Wannan, R., Sajjadi, S.S. and Asad, J.H. *Fractional investigation of time-dependent mass pendulum*, J. Low Freq. Noise Vib. Act. Control, 43(1) (2024), 196–207.
- [6] Baleanu, D., Jajarmi, A., Sajjadi, S.S. and Mozyrska, D. *A new fractional model and optimal control of a tumor-immune surveillance with non-singular derivative operator*, Chaos, 29(8) (2019).
- [7] Baleanu, D., Shekari, P., Torkzadeh, L., Ranjbar, H., Jajarmi, A. and Nouri, K. *Stability analysis and system properties of Nipah virus trans-*

- mission: A fractional calculus case study*, Chaos Solitons Fractals, 166 (2023), 112990.
- [8] Bellout, A., Bououden, R., Houmor, T. and Berkal, M. *Nonlinear dynamics and chaos in fractional-order cardiac action potential duration mapping model with fixed memory length*, Gulf J. Math., 19(2) (2025), 369–383.
- [9] Bemporad, A. and Morari, M. *Control of systems integrating logic, dynamics, and constraints*, Automatica, 35(3) (1999), 407–427.
- [10] Berkal, M. and Almatrafi, M.B. *Bifurcation and stability of two-dimensional activator-inhibitor model with fractional-order derivative*, Fractal Fract., 7(5) (2023), 344.
- [11] Berkal, M. and Navarro, J.F. *Qualitative behavior of a chemical reaction system with fractional derivatives*, Rocky Mt. J. Math., 55(1) (2025), 11–24.
- [12] Bischi, G.I., Gardini, L. and Kopel, M. *Analysis of global bifurcations in a market share attraction model*, J. Econ. Dyn. Control, 24 (2000), 855–879.
- [13] Bououden, R. and Abdelouahab, M.S. *On efficient chaotic optimization algorithm based on partition of data set in global research step*, Nonlinear Dyn. Syst. Theory, 18 (2018), 42–52.
- [14] Bououden, R. and Abdelouahab, M.S. *Chaos in new 2-d discrete mapping and its application in optimisation*, Nonlinear Dyn. Syst. Theory, 20 (2020), 144–152.
- [15] Bououden, R. and Abdelouahab, M.S. *Chaotic optimization algorithm based on the modified probability density function of Lozi map*, Bol. da Soc. Parana. de Mat., 39 (2021), 9–22.
- [16] Bououden, R., Abdelouahab, M.S. and Jarad, F. *Non linear dynamics and chaos in a new 2D piecewise linear map and its fractional version*, Electron. Res. Arch., 28 (2020), 505–525.

- [17] Bououden, R., Abdelouahab, M.S., Jarad, F. and Hammouch, Z. *A novel fractional piecewise linear map regular and chaotic dynamics*, Int. J. General Syst., 50 (2021), 501–526.
- [18] Bourafa, S., Abdelouahab, M.S. and Lozi, R. *On Periodic Solutions of Fractional-Order Differential Systems with a Fixed Length of Sliding Memory*, J. Innov. Appl. Math. Comput. Sci., 1(1) (2021), 64–78.
- [19] Bouzeraa, S.E.I., Bououden, R. and Abdelouahab, M.S. *Fractional logistic map with fixed memory length*, Int. J. Gen. Syst., 52 (2023), 653–663.
- [20] Chen, F., Luo, X. and Zhou, Y. *Existence Results for Nonlinear Fractional Difference Equation*, Adv. Differ. Equ., 2011 (2011), 1–12.
- [21] Crampin, M. and Heal, B. *On the chaotic behaviour of the Tent map*, An Inter. J. of IMA, 13 (1994), 83–89.
- [22] Dai, W., Zhou, R., Lin, Y. and Liu, Y. *Lightweight cryptography for embedded systems—A survey*, Sensors, 23(2) (2023).
- [23] Donato, C. and Grassi, G. *Bifurcation and chaos in the fractional-order Chen system via a time-domain approach*, Int. J. Bifurcat. Chaos, 10 (2008), 1845–1863.
- [24] Ebrahimzadeh, A., Jajarmi, A. and Baleanu, D. *Enhancing water pollution management through a comprehensive fractional modeling framework and optimal control techniques*, J. Nonlinear Math. Phys., 31(1) (2024), 48.
- [25] Gümüş, M. and Türk, K. *Dynamical behavior of a hepatitis B epidemic model and its NSFD scheme*, J. Appl. Math. Comput., 70(4) (2024), 3767–3788.
- [26] Gümüş, M. and Teklu, S.W. *Cost-Benefit and dynamical investigation of a fractional-order corruption population dynamical system*, Fractal Fract., 9(4) (2025) 207.
- [27] Gümüş, M. and Türk, K. *Global analysis of a monkey-pox virus model considering government interventions*, Phys. Scr., 100(4) (2025), 045216.

- [28] Hénon, M. *A two dimensional mapping with a strange attractor*, Commun. Math. Phys., 50 (1976), 69–77.
- [29] Hilborn, R.C. *Chaos and nonlinear dynamics: an introduction for scientists and engineers*, Oxford Univ. Press, New York, 2011.
- [30] Hu, T. *Discrete chaos in fractional Henon map*, Appl. Math., 5 (2014), 2243–2248.
- [31] Jan, C. and Nechvatel, L. *Local bifurcations and chaos in the fractional Rossler system*, Int. J. Bifurcat. Chaos, 28 (2018), 1850–098.
- [32] Jianping, S., He, K. and Fang, H. *Chaos, Hopf bifurcation and control of a fractional-order delay financial system*, Math. Comput. Simulat., 194 (2022), 348–364.
- [33] Khennaoui, A.A., Ouannas, A., Bendoukha, S., Wang, X. and Pham, V.T. *On chaos in the fractional-order discrete-time unified system and its control synchronization*, Entropy, 20 (2018), 530–540.
- [34] Li, T.Y. and Ismail, J.A. *Period three implies chaos*, Am. Math. Mon., 82 (1975), 985–992.
- [35] Lozi, R. *Un attracteur étrange du type attracteur de Hénon*, J. Phys., 39 (1978), 9–10.
- [36] Magin, R.L. *Fractional calculus in bioengineering*, Begell House Publishers, 2006.
- [37] May, R. *Simple mathematical models with very complicated dynamics*, Nature, 261 (1976), 459–467.
- [38] Miller, K.S. and Ross, B. *Fractional difference calculus*, Proc. of the International Symposium on Univalent Functions, Koriyama, Japan, 1989, 139–152.
- [39] Sarmah, H.K., Das, M.C., Baishya, T.K. and Paul, R. *Chaos in Gaussian map*, Int. J. Adv. Sci. Techn. Res., 6 (2016), 160–172.

- [40] Sun, H., Chen, W. and Chen, Y. *Variable-order fractional differential models of cardiac action potential*, Commun. Nonlinear Sci. Numer., 69 (2019), 354–370.
- [41] Wu, G.C. and Baleanu, D. *Discrete fractional logistic map and its chaos*, Nonlinear Dyn., 75 (2014), 283–287.



Utilizing the Hybrid approach of the Ramadan group transform and accelerated Adomian method for solving nonlinear integro-differential equations

M.A. Ramadan*, M.M.A. Mansour and H.S. Osheba

Abstract

In this paper, we investigate the application of the combination of the Ramadan group transform and the accelerated Adomian polynomial

*Corresponding author

Received 22 May 2025; revised 10 July 2025; accepted 14 July 2025

Mohamed Abdellatif Ramadan

Mathematics and Computer Science Department, Faculty of Science,
Menoufia University, Egypt. e-mail: ramadanmohamed13@yahoo.com, mohamed.Abdellatif@science.menofia.edu.eg

Mariam M. A. Mansour

Department of basic science, Modern Academy of Computer Science and Management
Technology in Maadi, Egypt. e-mail: mariamatared2@gmail.com

Heba S. Osheba

Mathematics and Computer Science Department, Faculty of Science, Menoufia
University, Egypt. e-mail: heba_osheba@yahoo.com

How to cite this article

Ramadan, M.A., Mansour, M.M.A. and Osheba, H.S., Utilizing the Hybrid approach of the Ramadan group transform and accelerated Adomian method for solving nonlinear integro-differential equations. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1332-1360.
<https://doi.org/10.22067/ijnao.2025.93657.1653>

method for solving integro-differential equations. Integro-differential equations arise in various fields such as physics, engineering, and biology, often modeling complex phenomena. The Ramadan group transform, known for its transformation properties and its ability to simplify computational complexities, is coupled with the accelerated Adomian polynomial method, which is an effective series expansion technique. This combination enhances the convergence and efficiency of solving nonlinear integro-differential equations that are difficult to handle using traditional methods. The paper demonstrates the utility of this hybrid approach through several test cases, comparing it with existing methods in terms of accuracy, computational efficiency, and convergence rate.

AMS subject classifications (2020): Primary 45J05; Secondary 65R20, 65H10, 65L09, 44A10.

Keywords: Ramadan group transform (RGT); Adomian polynomials; accelerated Adomian; Integro-differential equations; Accuracy.

1 Introduction

An equation with the unknown function under the sign of integration and including the unknown function's derivatives is known as an integro-differential equation (IDE). It falls into one of two categories: Volterra equations or Fredholm equations. IDEs are one of the most important tools in mathematics [33].

Many researchers and scientists investigated IDEs while working on scientific applications such as heat transformers, neutron diffusion, and biological species coexisting with growing and decreasing rates of production and diffusion processes. Applications in physics, biology, and engineering, as well as models addressing complex integral equations like [14, 16], also use these kinds of equations. An IDE system can be solved using a variety of methods, such as the variational iteration method (VIM) [29], the rationalized Haar functions method [18], the Adomian decomposition method (ADM) [8, 26], and work by Younis and Al-Hayani [31, 3], the Galerkin method [19], and

He's homotopy perturbation method (HPM) [9, 32] and the work by Younis and Al-Hayani [30].

The analytical method known as ADM uses Adomian polynomials to evaluate the answer. Both linear and nonlinear issues can be solved using this method, which neither simplifies nor discretizes the provided problem. The Galerkin and rationalized Haar function methods are numerical techniques that can be used to solve IDEs in a variety of ways. The HPM, introduced by He in 1997 and further detailed in 2000, combines traditional perturbation techniques with the concept of Homotopy from topology [15]. He developed and extended this innovative method, which has since been applied to a wide range of linear and nonlinear problems.

Also, the use of the Laplace transform HPM by Al-Hayani [2]. Another analytical method, the VIM, is also capable of addressing various linear and nonlinear challenges. Additionally, Avudainayagam and Vani [5] explored the use of wavelet bases for solving IDEs. They proposed a method for computing a novel four-dimensional connection coefficient and validated their approach by solving two basic educational nonlinear IDEs [6].

Interest in linear and nonlinear Volterra integro-differential equations (VIDEs), which blend differential and integral components, has significantly increased in recent years [28]. Nonlinear VITEs are fundamental in various areas of nonlinear functional analysis and find widespread applications in engineering, mechanics, physics, electrostatics, biology, chemistry, and economics [7].

Recently, Ramadan et al. [21, 23, 22] have proposed the Ramadan group transform (RGT) and the accelerated Adomian method to address solutions for quadratic Riccati differential equations, the nonlinear Sharma–Tasso–Olver equation, and other forms of nonlinear partial differential equations.

In this paper, we present the RGT and accelerated Adomian method for solving the nonlinear VIDEs of the type:

$$y^{(i)}(x) = f(x) + \gamma \int_0^x K(x, t)G(y(x))dt ,$$

with the initial conditions

$$y^{(r)}(a) = b_r, \quad r = 0, 1, 2, \dots, (i-1),$$

where $y^{(i)}(x)$ is the i th derivative of the unknown function $y(x)$ that will be determined, $K(x, t)$ is the kernel of the equation, $f(x)$ is an analytic function, G is a nonlinear function of y , and a, b, γ and b_r are real finite constants. The main objective of this contribution is to present a comparative study of solving IDEs using the RGT method coupled with an accelerated Adomian method and solving them using other methods.

This paper is organized as follows:

Mathematical preliminaries and notions are stated in Section 2.

In Section 3, the analysis of the hybrid RGT accelerated Adomian method is explained thoroughly.

In Section 4, the proof of convergence of the hybrid RGT accelerated Adomian method when applied to a class of nonlinear Volterra-type IDEs, including the sufficient conditions guaranteeing existence and uniqueness are introduced.

To demonstrate the correctness and effectiveness of the suggested approach in comparison to the current one's numerical examples are solved in Section 5.

Concluding remarks are given in the last section.

2 Mathematical preliminaries and notions

We give the reader basic definitions and theorems in this section so they may comprehend RGT and its fundamental characteristics.

2.1 The Adomian polynomials [1]

A wide range of linear and nonlinear functional equations can be analytically approximated using the ADM.

The solution is defined by the infinite series in the standard ADM,

$$y = \sum_{n=0}^{\infty} y_n,$$

after which the nonlinear term Ny is broken down into an infinite series. Moreover,

$$Ny = \sum_{n=0}^{\infty} A_n ,$$

where the regular Adomian polynomials are denoted by the A_n and are derived using the definitional formula for the nonlinearity $Ny = f(y)$. Also,

$$A_n = \frac{1}{n!} \left(\frac{d^n}{d\lambda^n} [N(\sum_{i=0}^n \lambda^i y_i)] \right)_{\lambda=0}, \quad n = 0, 1, 2, \dots$$

If $N(y) = y^2(x)$, then Adomian polynomials [1, 10] are

$$\begin{aligned} A_0 &= y_0^2, \\ A_1 &= 2y_0y_1, \\ A_2 &= y_1^2 + 2y_0y_2. \end{aligned}$$

Ordinary and partial differential equations are solved by approximating the nonlinear term functions using the Adomian polynomials $\{A_n\}$.

2.2 Accelerated Adomian polynomials (El-Kalla Adomian polynomials) [27, 12, 13, 11]

The accelerated Adomian polynomials are given in the following form:

$$\bar{A}_n = N(s_n) - \sum_{i=0}^{n-1} \bar{A}_i,$$

where \bar{A}_n , are accelerated Adomian polynomials, $\bar{A}_0, \bar{A}_1, \bar{A}_2, \dots$ and $N(s_n)$.

Use the nonlinearity (n -times) to substitute the total of the responses.

If $N(y) = y^2(x)$, then accelerated Adomian polynomials are

$$\begin{aligned} \bar{A}_0 &= y_0^2, \\ \bar{A}_1 &= 2y_0y_1 + y_1^2, \\ \bar{A}_2 &= 2y_0y_2 + 2y_1y_2 + y_2^2, \end{aligned}$$

and if $N(y) = y^3(x)$, then accelerated Adomian polynomials are

$$\begin{aligned}\overline{A}_0 &= y_0^3, \\ \overline{A}_1 &= 3y_0^2y_1 + 3y_0y_1^2 + y_1^3, \\ \overline{A}_2 &= 3y_0^2y_2 + 6y_0y_1y_2 + 3y_1^2y_2 + 3y_0y_2^2 + 3y_1y_2^2 + y_2^3.\end{aligned}$$

2.3 Ramadan group integral transform [24]

For exponentially ordered functions, a novel integral RGT was introduced. Functions in set A are examined, as defined by

$$A = \{f(t) : \exists M, t_1, t_2 > 0 \text{ s.t. } |f(t)| < Me^{\frac{|t|}{t_n}}, \text{ if } t \in (-1)^n \times [0, \infty)\}.$$

The RGT is defined by

$$\begin{aligned}K(s, u) &= RG[f(t); (s, u)] \\ &= \begin{cases} \int_0^\infty e^{-st} f(ut) dt, & -t_1 < u \leq 0, \\ \int_0^\infty e^{-st} f(ut) dt, & 0 \leq u < t_2. \end{cases}\end{aligned}$$

2.4 Ramadan group transform (RGT) convolution theorem

Definition 1 (Convolution of two functions [20]). The convolution of piecewise continuous functions $f(x), g(x) : R \rightarrow R$ is the function $f * g : R \rightarrow R$ and is determined by the integral

$$f * g = \int_0^x f(t)g(x-t)dt.$$

Theorem 1 (Convolution theorem of RGT). [20]

Let $f(x)$ and $g(x)$ be two functions with RGTs $K_1(s, u)$ and $K_2(s, u)$, respectively. Then

$$RG[(f * g)(s, u)] = uK_1(s, u)K_2(s, u),$$

and

$$RG^{-1}[uK_1(s, u)K_2(s, u)] = f * g.$$

Proof. See [21] for theorem's proof. \square

Table 1: Ramadan group transform (RGT) of some functions

| $f(t)$ | $RG[f(t)] = K(s, u)$ |
|--------------------------|------------------------|
| 1 | $\frac{1}{s}$ |
| t | $\frac{u}{s^2}$ |
| $\frac{t^{n-1}}{(n-1)!}$ | $\frac{u^{n-1}}{s^n}$ |
| $\frac{1}{\sqrt{\pi t}}$ | $\frac{1}{\sqrt{su}}$ |
| e^{at} | $\frac{1}{s-au}$ |
| te^{at} | $\frac{u}{(s-au)^2}$ |
| $\frac{\sin Wt}{W}$ | $\frac{u}{s^2+u^2w^2}$ |
| $\cos Wt$ | $\frac{s}{s^2+u^2w^2}$ |
| $\frac{\sin at}{a}$ | $\frac{u}{s^2+u^2a^2}$ |

3 Analysis of the Hybrid RGT accelerated Adomian method

This section outlines the steps of the suggested method for solving nonlinear IDEs where the accelerated Adomian polynomial appears in the estimated solution.

$$y^{(n)}(x) = f(x) + \int_0^x K(x-t)G(y(x))dt. \quad (1)$$

Applying the RGT for both sides, we get

$$\begin{aligned} \frac{s^n}{u^n}RG[y(x)] - \frac{s^{n-1}}{u^n}y(0) - \frac{s^{n-2}}{u^{n-1}}y'(0) - \dots - \frac{1}{u}y^{(n-1)}(0) \\ = RG[f(x)] + RG\left[K(x) \otimes G(y(x))\right]. \end{aligned} \quad (2)$$

The RGT of convolution term $K(x) \otimes G(y(x))$ can be written as a product of terms, so,

$$\begin{aligned} \frac{s^n}{u^n} RG[y(x)] - \frac{s^{n-1}}{u^n} y(0) - \frac{s^{n-2}}{u^{n-1}} y'(0) - \dots - \frac{1}{u} y^{(n-1)}(0) \\ = RG[f(x)] + uRG[K(x)] RG[G(y(x))] . \end{aligned} \quad (3)$$

This can be reduced to

$$\begin{aligned} \frac{s^n}{u^n} RG[y(x)] = \frac{s^{n-1}}{u^n} y(0) + \frac{s^{n-2}}{u^{n-1}} y'(0) + \dots + \frac{1}{u} y^{(n-1)}(0) \\ + RG[f(x)] + uRG[K(x)] RG[G(y(x))] , \end{aligned} \quad (4)$$

$$\begin{aligned} RG[y(x)] = \frac{u^n}{s^n} \left[\frac{s^{n-1}}{u^n} y(0) + \frac{s^{n-2}}{u^{n-1}} y'(0) + \dots + \frac{1}{u} y^{(n-1)}(0) \right] \\ + \frac{u^n}{s^n} RG[f(x)] + \frac{u^{n+1}}{s^n} RG[K(x)] RG[G(y(x))] . \end{aligned} \quad (5)$$

Applying the inverse RGT for both sides, we get

$$\begin{aligned} y(x) = RG^{-1} \left[\frac{u^n}{s^n} \left[\frac{s^{n-1}}{u^n} y(0) + \frac{s^{n-2}}{u^{n-1}} y'(0) + \dots + \frac{1}{u} y^{(n-1)}(0) \right] \right] \\ + RG^{-1} \left[\frac{u^n}{s^n} RG[f(x)] \right] + RG^{-1} \left[\frac{u^{n+1}}{s^n} RG[K(x)] RG[G(y(x))] \right] . \end{aligned} \quad (6)$$

We represent the linear term $y(x)$ at the left side by an infinite series of components given by

$$y(x) = \sum_{n=0}^{\infty} y_n(x). \quad (7)$$

The nonlinear term $G(y(x))$ at the right side of (6) will be represented by an infinite series of the accelerated Adomian polynomials \bar{A}_n

$$G(y(x)) = \sum_{n=0}^{\infty} \bar{A}_n(x). \quad (8)$$

3.1 Accelerated Adomian polynomials \bar{A}_n formula

If the nonlinear function is $G(y(x)) = y^2(x)$, then the accelerated Adomian polynomials \bar{A}_n are

$$\begin{aligned}\bar{A}_0 &= y_0^2, \\ \bar{A}_1 &= 2y_0y_1 + y_1^2, \\ \bar{A}_2 &= 2y_0y_2 + 2y_1y_2 + y_2^2,\end{aligned}$$

and $G(y(x)) = y^3(x)$, the accelerated Adomian polynomials \bar{A}_n are

$$\begin{aligned}\bar{A}_0 &= y_0^3, \\ \bar{A}_1 &= 3y_0^2y_1 + 3y_0y_1^2 + y_1^3, \\ \bar{A}_2 &= 3y_0^2y_2 + 6y_0y_1y_2 + 3y_1^2y_2 + 3y_0y_2^2 + 3y_1y_2^2 + y_2^3,\end{aligned}$$

where $\bar{A}_n, n \geq 0$ can be obtained for all forms of nonlinearity. Substituting (7) and (8) into (5) leads to

$$\begin{aligned}\sum_{n=0}^{\infty} y_n(x) &= RG^{-1} \left[\frac{u^n}{s^n} \left[\frac{s^{n-1}}{u^n} y(0) + \frac{s^{n-2}}{u^{n-1}} y'(0) + \cdots + \frac{1}{u} y^{(n-1)}(0) \right] \right] \\ &\quad + RG^{-1} \left[\frac{u^n}{s^n} RG[f(x)] \right] \\ &\quad + RG^{-1} \left[\frac{u^{n+1}}{s^n} RG[K(x)] RG \left[\sum_{n=0}^{\infty} \bar{A}_n(x) \right] \right],\end{aligned}$$

where

$$\begin{aligned}y_0(x) &= RG^{-1} \left[\frac{u^n}{s^n} \left[\frac{s^{n-1}}{u^n} y(0) + \frac{s^{n-2}}{u^{n-1}} y'(0) + \cdots + \frac{1}{u} y^{(n-1)}(0) \right] \right] \\ &\quad + RG^{-1} \left[\frac{u^n}{s^n} RG[f(x)] \right], \\ y_{n+1}(x) &= RG^{-1} \left[\frac{u^{n+1}}{s^n} RG[K(x)] RG \left[\sum_{n=0}^{\infty} \bar{A}_n(x) \right] \right], \quad n \geq 0.\end{aligned}$$

4 Convergence of the proposed method

In this section, we present and prove a convergence theorem for the application of the hybrid RGT in combination with the accelerated Adomian polynomials.

Theorem 2. The solution of the nonlinear Volterra-type integro-differential equation

$$y^{(i)}(x) = f(x) + \int_{x_0}^x K(x, t)G(y(x))dt ,$$

using RGT converges if $G(y(x))$ satisfy Lipschitz condition in the interval of interest $J = [0, b]$ and this solution, is unique provided that $0 < MM_1 \frac{(x-x_0)^{i+1}}{(i+1)!} < 1$, for all $x \in J$, where M is the Lipschitz inequality constant.

Proof. Define a complete metric space $(C[0, b], d)$, the space of all continuous functions on J with the distance function

$$d(f_1(x), f_2(x)) = \max_{\forall x \in J} |f_1(x) - f_2(x)| .$$

Define the sequence $\{S_n\}$ such that $S_n = \sum_{i=0}^n y_i(x) = y_0 + y_1 + \dots + y_n$ is a sequence of partial sums of the series solution $\sum_{i=0}^{\infty} y_i(x)$, since

$$\begin{aligned} f\left(\sum_{i=0}^{\infty} y_i(x)\right) &= \sum_{i=0}^{\infty} \bar{A}_i(y_0, y_1, \dots, y_i), \\ f(S_n) &= \sum_{i=0}^{\infty} \bar{A}_i(y_0, y_1, \dots, y_i). \end{aligned}$$

Let, S_n and S_m be arbitrary partial sums with $n \geq m$. We prove that $\{S_n\}$ is a Cauchy sequence in this complete metric space:

$$\begin{aligned} d(S_n, S_m) &= \max_{\forall x \in J} \|S_n - S_m\| \\ &= \max_{\forall x \in J} \left| \sum_{i=m+1}^n y_i(x) \right| \\ &= \max_{\forall x \in J} \left| \sum_{i=m+1}^n RG^{-1} \left[\int_{x_0}^x K(x, t) \bar{A}_{i-1} dt \right] \right| \\ &= \max_{\forall x \in J} \left| \sum_{i=m+1}^n RG^{-1} \left[\int_{x_0}^x K(x, t) \bar{A}_{i-1} dt \right] \right| \end{aligned}$$

$$\begin{aligned}
&= \max_{\forall x \in J} \left| RG^{-1} \left[\int_{x_0}^x K(x, t) \sum_{i=m}^{n-1} \bar{A}_i dt \right] \right| \\
&= \max_{\forall x \in J} \left| RG^{-1} \left[\int_{x_0}^x K(x, t) [f(S_{n-1}) - f(S_{m-1})] dt \right] \right| \\
&\leq \max_{\forall x \in J} RG^{-1} \left[\int_{x_0}^x |K(x, t)| |f(S_{n-1}) - f(S_{m-1})| dt \right] \\
&\leq M_1 \max_{\forall x \in J} |f(S_{n-1}) - f(S_{m-1})| RG^{-1} \int_{x_0}^x dt.
\end{aligned}$$

Since $f(x)$ satisfy Lipschitz condition,

$$|f(S_{n-1}) - f(S_{m-1})| \leq M |S_{n-1} - S_{m-1}|,$$

so,

$$\begin{aligned}
d(S_m, S_n) &\leq MM_1 \max_{\forall x \in J} |S_{n-1} - S_{m-1}| \int_{x_0}^x \dots (i+1) - fold \dots \int_{x_0}^x dt \dots dt, \\
&\leq MM_1 \frac{(x - x_0)^{i+1}}{(i+1)!} d(S_{m-1}, S_{n-1}), \\
&\leq \alpha d(S_{m-1}, S_{n-1}), \quad \alpha = MM_1 \frac{(x - x_0)^{i+1}}{(i+1)!}.
\end{aligned}$$

Now, for $n = m + 1$,

$$d(S_{m+1}, S_m) \leq \alpha d(S_m, S_{m-1}) \leq \alpha^2 d(S_{m-1}, S_{m-2}) \leq \dots \leq \alpha^m d(S_1, S_0).$$

From the triangle inequality, we have

$$\begin{aligned}
d(S_m, S_n) &\leq \alpha [d(S_{m-1}, S_m) + d(S_m, S_{m+1}) + \dots + d(S_{n-2}, S_{n-1})], \\
&\leq \alpha [\alpha^{m-1} + \alpha^m + \dots + \alpha^{n-2}] d(S_1, S_0), \\
&\leq \alpha^m \frac{1 - \alpha^{n-m}}{1 - \alpha} d(S_1, S_0), \\
&\leq \frac{\alpha^m}{1 - \alpha} d(S_1, S_0).
\end{aligned}$$

Indeed, $d(S_1, S_0) = \max_{\forall x \in J} |S_1 - S_0| = \max_{\forall x \in J} |y_1|$, which is bounded. As $m \rightarrow \infty$, $d(S_m, S_n) \rightarrow 0$ we conclude that $\{S_n\}$ is a Cauchy sequence in this complete metric space, so the series $\sum_{n=0}^{\infty} y_n(x)$ con-

verges. For the uniqueness of the solution, assume that y and y^* are two different solutions. Then from (6), we have

$$\begin{aligned} d(y, y^*) &= \max_{\forall x \in J} \left| RG^{-1} \left[\int_{x_0}^x K(x, t) [f(y) - f(y^*)] dt \right] \right|, \\ &\leq \max_{\forall x \in J} RG^{-1} \left[\int_{x_0}^x |K(x, t)| |f(y) - f(y^*)| dt \right], \\ &\leq M_1 \max_{\forall x \in J} |f(y) - f(y^*)| RG^{-1} \int_{x_0}^x dt, \\ &\leq \alpha d(y, y^*). \end{aligned}$$

So, $(1 - \alpha)d(y, y^*) \leq 0$ and $0 < \alpha \leq 1$; then $d(y, y^*) = 0$, which implies $y = y^*$.

□

5 Numerical examples

In this numerical section, we apply the hybrid RGT and accelerated Adomian polynomials to solve several nonlinear IDEs. The results are compared with traditional methods, highlighting improvements in accuracy and computational efficiency, demonstrating the effectiveness of the proposed approach.

Example 1. Consider the nonlinear VIDE [25]

$$y'(x) = -1 + \int_0^x y^2(t) dt, \quad y(0) = 0,$$

whose exact solution takes the form

$$y(x) = \frac{-x + \frac{x^4}{28}}{1 + \frac{x^3}{21}}.$$

This example is solved by Rani and Mishra [25] where they used Laplace and a modification of ADM by computing the Adomian polynomials for the nonlinear term using the Newton-Raphson formula. We applied our hybrid method for combining RGT and the accelerated version of ADM. Four itera-

tions are carried out and the approximate series solution is evaluated at the corresponding points as in [25].

Applying the RGT for both sides, we get

$$\begin{aligned}RG[\hat{y}(x)] &= RG[-1] + RG\left[\int_0^x y^2(t)dt\right], \\ \frac{s}{u}RG[y(x)] - \frac{1}{u}y(0) &= RG[-1] + RG\left[\int_0^x y^2(t)dt\right], \\ \frac{s}{u}RG[y(x)] &= \frac{-1}{s} + RG\left[\int_0^x y^2(t)dt\right], \\ RG[y(x)] &= \frac{-u}{s^2} + \frac{u}{s}RG\left[\int_0^x y^2(t)dt\right].\end{aligned}$$

Applying the inverse RGT for both sides, we get

$$\begin{aligned}y(x) &= RG^{-1}\left[\frac{-u}{s^2}\right] + RG^{-1}\left[\frac{u}{s}RG\left[\int_0^x y(t)^2dt\right]\right], \\ y(x) &= -x + RG^{-1}\left[\frac{u}{s}RG\left[\int_0^x y^2(t)dt\right]\right].\end{aligned}$$

$$\text{Let } y^2(t) = \sum_{n=0}^{\infty} A_n,$$

$$\sum_{n=0}^{\infty} y_n(x) = -x + RG^{-1}\left[\frac{u}{s}RG\left[\int_0^x \sum_{n=0}^{\infty} A_n dx\right]\right].$$

By comparing both sides, we get

$$\begin{aligned}y_0(x) &= -x, \\ y_{n+1}(x) &= RG^{-1}\left[\frac{u}{s}RG\left[\int_0^x \sum_{n=0}^{\infty} A_n dx\right]\right].\end{aligned}$$

Using accelerated Adomian polynomials, we have

$$\overline{A}_0 = y_0^2, \quad \overline{A}_1 = 2y_0y_1 + y_1^2, \quad \overline{A}_2 = 2y_0y_2 + 2y_1y_2 + y_2^2, \quad \dots$$

Then

$$y_0(x) = -x, y_1(x) = \frac{x^4}{12}, y_2(x) = -\frac{x^7}{252} + \frac{x^{10}}{12960},$$

$$y_3(x) = \frac{11831339520x^{10} - 701537760x^{13} + 15992262x^{16} - 240240x^{19} + 1729x^{22}}{134167390156800}$$

$$y_4(x) = -\frac{x^{13}}{884520} + \frac{89x^{16}}{849139200} - \frac{10757x^{19}}{2032839244800} + \frac{350993x^{22}}{1878343462195200}$$

$$- \frac{4507x^{25}}{901604861853696} + \frac{24354871x^{28}}{221524314557453107200}$$

$$- \frac{1312457x^{31}}{628869391142952960000} + \frac{253524431x^{34}}{7647700951798842654720000}$$

$$- \frac{1709x^{37}}{4053164656463290368000} + \frac{241247x^{40}}{59943018919370607820800000}$$

$$- \frac{x^{43}}{39132841298576965632000} + \frac{x^{46}}{12464483949901696204800000}.$$

$$y(x) (\text{approximate}) = y_0 + y_1 + y_2 + y_3 + y_4$$

$$= -x + \frac{x^4}{12} - \frac{x^7}{252} + \frac{x^{10}}{6048} - \frac{x^{13}}{157248} + \frac{2663x^{16}}{11887948800} + \dots$$

Table 2: Comparison of the approximate solutions and absolute error against the method of Laplace Adomian using Newton–Raphson formula [25]

| x | Exact solution | Approximate solution (presented method, 4 iterations) | Absolute error | Approximate solution [25] (4 iterations) | Absolute error |
|--------|---------------------|---|-------------------------|--|-------------------------|
| 0. | 0. | 0. | 0. | 0. | 0. |
| 0.0625 | -0.0624987284490275 | -0.0624987284490275 | 2.082×10^{-17} | -0.062499682 | 3.1789×10^{-7} |
| 0.125 | -0.124979656839952 | -0.124979656839974 | 2.2×10^{-14} | -0.124994914 | 1.4914×10^{-5} |
| 0.1875 | -0.187397035493881 | -0.187397035495149 | 1.268×10^{-12} | -0.187474253 | 7.4253×10^{-5} |
| 0.25 | -0.249674721189591 | -0.249674721212078 | 2.249×10^{-11} | -0.249918635 | 2.4863×10^{-4} |
| 0.3125 | -0.311706424640996 | -0.311706424850075 | 2.091×10^{-10} | -0.31230139 | 5.9139×10^{-3} |
| 0.375 | -0.373356178680768 | -0.373356179972127 | 1.291×10^{-9} | -0.374588271 | 1.2283×10^{-3} |
| 0.4375 | -0.434459096619924 | -0.434459102631869 | 6.012×10^{-9} | -0.436737503 | 2.2775×10^{-3} |
| 0.5 | -0.494822485207101 | -0.494822507955013 | 2.275×10^{-8} | -0.498699852 | 3.8799×10^{-3} |

Table 2 and Figure 1 show that the proposed method achieves higher accuracy and improved computational efficiency, primarily because the accelerated Adomian polynomials eliminate the requirement to compute derivatives of the nonlinear functions. Another notable advantage of using the accelerated polynomial is its superior rate of convergence compared to the traditional polynomials.

Example 2. Consider the nonlinear VIDE [17]

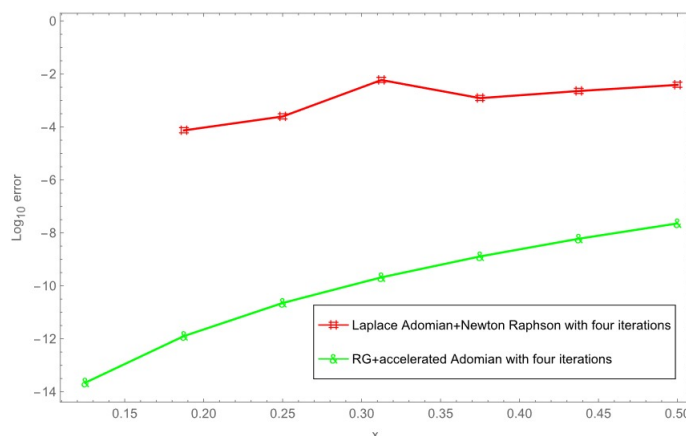


Figure 1: Comparison the absolute errors of the two approaches at four iterations.

$$y''(x) = 2 + 2x + x^2 - x^2 e^x - e^{2x} + \int_0^x e^{x-t} y^2(t) dt, \quad y(0) = 1, \quad y'(0) = 2,$$

whose exact solution takes the form $y(x) = x + e^x$.

Khanlari and Paripour [17] solved this example using a combination of Laplace and the homotopy analysis method (*HAM*) by computing the Adomian polynomials for the nonlinear term. We applied our hybrid method for combining the RGT and the accelerated version of ADM. Three iterations were carried out, and the approximate solution and absolute error were evaluated at the corresponding points as in [4].

We note that the integral term uses the RGT convolution theorem of the two functions e^x and $y^2(x)$ [13].

Applying the RGT for both sides, we get

$$RG[y''(x)] = RG[2 + 2x + x^2 - x^2 e^x - e^{2x}] + RG[e^x \otimes y^2(x)],$$

$$RG[y''(x)] = RG[2 + 2x + x^2 - x^2 e^x - e^{2x}] + uRG[e^x]RG[y^2(x)],$$

$$\begin{aligned} \frac{s^2}{u^2}RG[y(x)] - \frac{s}{u^2}y(0) - \frac{1}{u}y'(0) &= \frac{2}{s} + \frac{2u}{s^2} + \frac{2u^2}{s^3} - \frac{2u^2}{(s-u)^3} \\ &\quad - \frac{1}{-2u+s} + \frac{u}{s-u}RG[y^2(x)], \end{aligned}$$

$$\begin{aligned}\frac{s^2}{u^2}RG[y(x)] &= \frac{s}{u^2} + \frac{2}{u} + \frac{2}{s} + \frac{2u}{s^2} + \frac{2u^2}{s^3} - \frac{2u^2}{(s-u)^3} - \frac{1}{-2u+s} \\ &\quad + \frac{u}{s-u}RG[y^2(x)], \\ RG[y(x)] &= \frac{1}{s} + \frac{2u}{s^2} + \frac{2u^2}{s^3} + \frac{2u^3}{s^4} + \frac{2u^4}{s^5} - \frac{2u^4}{s^2(s-u)^3} - \frac{u^2}{s^2(-2u+s)} \\ &\quad + \frac{u^3}{s^2(s-u)}RG[y^2(x)].\end{aligned}$$

Applying the inverse RGT for both sides, we get

$$\begin{aligned}y(x) &= RG^{-1} \left[\frac{1}{s} + \frac{2u}{s^2} + \frac{2u^2}{s^3} + \frac{2u^3}{s^4} + \frac{2u^4}{s^5} - \frac{2u^4}{s^2(s-u)^3} - \frac{u^2}{s^2(-2u+s)} \right] \\ &\quad + RG^{-1} \left[\frac{u^3}{s^2(s-u)}RG[y^2(x)] \right], \\ \sum_{n=0}^{\infty} y_n(x) &= RG^{-1} \left[\frac{1}{s} + \frac{2u}{s^2} + \frac{2u^2}{s^3} + \frac{2u^3}{s^4} + \frac{2u^4}{s^5} - \frac{2u^4}{s^2(s-u)^3} - \frac{u^2}{s^2(-2u+s)} \right] \\ &\quad + RG^{-1} \left[\frac{u^3}{s^2(s-u)}RG \left[\sum_{n=0}^{\infty} A_n \right] \right].\end{aligned}$$

By comparing both sides and using the Taylor series from 0 to 4, we get

$$\begin{aligned}y_0(x) &= 1 + 2x + \frac{x^2}{2} - \frac{x^4}{6} + \dots, \\ y_{n+1}(x) &= RG^{-1} \left[\frac{u^3}{s^2(s-u)}RG \left[\sum_{n=0}^{\infty} A_n \right] \right].\end{aligned}$$

Using accelerated Adomian polynomials, we have

$$\begin{aligned}\overline{A}_0 &= y_0^2, \\ \overline{A}_1 &= 2y_0y_1 + y_1^2, \\ \overline{A}_2 &= 2y_0y_2 + 2y_1y_2 + y_2^2, \\ &\vdots\end{aligned}$$

Then

$$\begin{aligned}
y_0(x) &= 1 + 2x + \frac{x^2}{2} - \frac{x^4}{6} + \dots, \\
y_1(x) &= \frac{x^3}{6} + \frac{5x^4}{24} + \frac{x^5}{8} + \frac{3x^6}{80} + \dots, \\
y_2(x) &= \frac{x^6}{360} + \frac{x^7}{180} + \frac{89x^8}{20160} + \dots, \\
y_3(x) &= \frac{x^9}{90720} + \frac{29x^{10}}{907200} + \dots, \\
y(x) (\text{approximate}) &= y_0 + y_1 + y_2 + y_3 = 1 + 2x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \dots.
\end{aligned}$$

Table 3: Comparison of the approximate solutions and absolute error against the combination of the HAM and Laplace transform-Adomian method [4]

| x | Exact solution | Approximate solution (presented method, 3 iterations) | Absolute error | Approximate solution [4] (4 iterations) | Absolute error |
|------|------------------|---|-------------------------|---|--------------------------|
| 0.00 | 1.00000000000000 | 1.00000000000000 | 0.0000 | 1.00000 | 0.00000 |
| 0.02 | 1.0402013400268 | 1.0402013400268 | 7.105×10^{-15} | 1.04042 | 2.16577×10^{-4} |
| 0.04 | 1.0808107741924 | 1.0808107741933 | 9.064×10^{-13} | 1.08175 | 9.37538×10^{-4} |
| 0.06 | 1.1218365465454 | 1.1218365465611 | 1.573×10^{-11} | 1.12412 | 2.28194×10^{-3} |
| 0.08 | 1.1632870676750 | 1.1632870677947 | 1.197×10^{-10} | 1.16767 | 4.38752×10^{-3} |
| 0.10 | 1.2051709180756 | 1.2051709186553 | 5.796×10^{-10} | 1.21259 | 7.41437×10^{-3} |
| 0.12 | 1.2474968515794 | 1.2474968536878 | 2.108×10^{-9} | 1.25905 | 1.15497×10^{-3} |
| 0.14 | 1.2902737988572 | 1.2902738051525 | 6.295×10^{-9} | 1.30729 | 1.70138×10^{-2} |
| 0.16 | 1.3335108709918 | 1.3335108872586 | 1.627×10^{-8} | 1.35758 | 2.40683×10^{-2} |
| 0.18 | 1.3772173631218 | 1.3772174007597 | 3.764×10^{-8} | 1.41024 | 3.30265×10^{-2} |
| 0.20 | 1.4214027581602 | 1.4214028379772 | 7.982×10^{-8} | 1.46567 | 4.42678×10^{-2} |

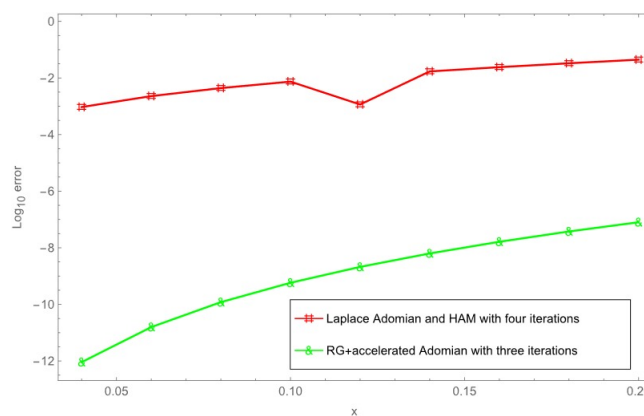


Figure 2: Comparison the approximate solutions of the two approaches.

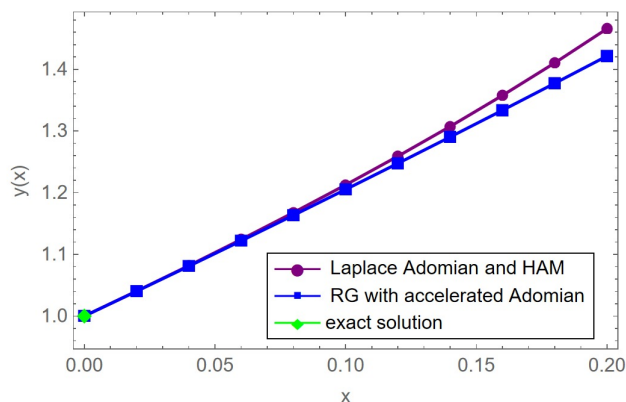


Figure 3: Comparison the absolute errors of the two approaches.

Based on Table 3 and Figures 2 and 3, the RGT combined with the accelerated Adomian method yields higher accuracy compared to the hybrid approach of the HAM and the Laplace transform-Adomian method. Notably, this improved performance is achieved using only three iterations, whereas HAM requires four.

Example 3. Consider the nonlinear VIDE [4]

$$y'''(x) = \frac{-2}{3} - \frac{5}{3}\cos(x) + \frac{4}{3}\cos^2(x) + \int_0^x \cos(x-t)y^2(t)dt,$$

$$y(0) = y'(0) = 1, \quad y''(0) = -1,$$

whose exact solution takes the form

$$y(x) = \sin(x) + \cos(x).$$

Almoussa et al. [4] approached this example by employing a combination of Laplace transform and HAM. They calculated the Adomian polynomials for the nonlinear term. In our study, we utilized a hybrid method that combines the RGT with an accelerated Adomian polynomial. We conducted two iterations and evaluated both the approximate solutions and absolute error at the corresponding points, as described in [20].

We note that the integral term uses the RGT convolution theorem of two functions $\cos(x)$ and $y^2(x)$ [10].

Applying the RGT for both sides, we get

$$\begin{aligned}
 RG[y'''(x)] &= RG\left[\frac{-2}{3} - \frac{5}{3}\cos(x) + \frac{4}{3}\cos^2(x)\right] + RG\left[\cos(x) \otimes y^2(x)\right], \\
 RG[y'''(x)] &= RG\left[\frac{-2}{3} - \frac{5}{3}\cos(x) + \frac{4}{3}\cos^2(x)\right] + uRG[\cos(x)] RG[y^2(x)], \\
 \frac{s^3}{u^3}RG[y(x)] - \frac{s^2}{u^3}y(0) - \frac{s}{u^2}y'(0) - \frac{1}{u}y''(0) \\
 &= -\frac{s^3 + 6su^2}{s^4 + 5s^2u^2 + 4u^4} + \frac{us}{s^2 + u^2}RG[y^2(x)], \\
 \frac{s^3}{u^3}RG[y(x)] &= \frac{s^2}{u^3} + \frac{s}{u^2} - \frac{1}{u} - \frac{s^3 + 6su^2}{s^4 + 5s^2u^2 + 4u^4} + \frac{us}{s^2 + u^2}RG[y^2(x)], \\
 RG[y(x)] &= \frac{1}{s} + \frac{u}{s^2} - \frac{u^2}{s^3} - \frac{u^3(s^3 + 6su^2)}{s^3(s^4 + 5s^2u^2 + 4u^4)} + \frac{u^4}{s^2(s^2 + u^2)}RG[y^2(x)].
 \end{aligned}$$

Applying the inverse RGT for both sides, we get

$$\begin{aligned}
 y(x) &= RG^{-1}\left[\frac{1}{s} + \frac{u}{s^2} - \frac{u^2}{s^3} - \frac{u^3(s^3 + 6su^2)}{s^3(s^4 + 5s^2u^2 + 4u^4)}\right] \\
 &\quad + RG^{-1}\left[\frac{u^4}{s^2(s^2 + u^2)}RG[y^2(x)]\right], \\
 \sum_{n=0}^{\infty} y_n(x) &= RG^{-1}\left[\frac{1}{s} + \frac{u}{s^2} - \frac{u^2}{s^3} - \frac{u^3(s^3 + 6su^2)}{s^3(s^4 + 5s^2u^2 + 4u^4)}\right] \\
 &\quad + RG^{-1}\left[\frac{u^4}{s^2(s^2 + u^2)}RG\left[\sum_{n=0}^{\infty} A_n\right]\right].
 \end{aligned}$$

By comparing both sides and using the Taylor series from 0 to 9, we get

$$\begin{aligned}
 y_0(x) &= 1 + x - \frac{x^2}{2} - \frac{x^3}{6} - \frac{x^5}{120} + \frac{x^7}{560} - \frac{41x^9}{362880} + \dots, \\
 y_{n+1}(x) &= RG^{-1}\left[\frac{u^4}{s^2(s^2 + u^2)}RG\left[\sum_{n=0}^{\infty} A_n\right]\right].
 \end{aligned}$$

Using accelerated Adomian polynomials

$$\bar{A}_0 = y_0^2, \bar{A}_1 = 2y_0y_1 + y_1^2, \bar{A}_2 = 2y_0y_2 + 2y_1y_2 + y_2^2,$$

$$\vdots$$

Then

$$y_0(x) = 1 + x - \frac{x^2}{2} - \frac{x^3}{6} - \frac{x^5}{120} + \frac{x^7}{560} - \dots,$$

$$y_1(x) = \frac{x^4}{24} + \frac{x^5}{60} - \frac{x^6}{720} - \frac{x^7}{504} - \frac{x^8}{40320} + \dots,$$

$$y_2(x) = \frac{x^8}{20160} + \dots,$$

$$y(x) (\text{approximate}) = y_0 + y_1 + y_2.$$

Table 4: Comparison of the approximate solutions and absolute error against the method of (HAM) [4]

| x | Exact solution | Approximate solution (presented method, 2 iterations) | Absolute error | Approximate solution [4] (4 iterations) | Absolute error |
|------|--------------------|---|-------------------------|---|--------------------------|
| 0.00 | 1.00000000000000 | 1.00000000000000 | 0 | 1.00000 | 0 |
| 0.02 | 1.019798673359911 | 1.019798673359911 | 0 | 1.01980 | 1.37991×10^{-6} |
| 0.04 | 1.0391894408476121 | 1.0391894408476123 | 2.22×10^{-16} | 1.03920 | 1.14101×10^{-5} |
| 0.06 | 1.0581645464146487 | 1.0581645464146487 | 0 | 1.05820 | 3.97525×10^{-5} |
| 0.08 | 1.0767164002717922 | 1.0767164002717917 | 4.441×10^{-16} | 1.07681 | 9.71538×10^{-5} |
| 0.10 | 1.094837581924854 | 1.0948375819248513 | 2.665×10^{-15} | 1.09503 | 1.95418×10^{-4} |
| 0.12 | 1.1125208431427855 | 1.1125208431427716 | 1.399×10^{-14} | 1.11287 | 3.47374×10^{-4} |
| 0.14 | 1.1297591108568736 | 1.1297591108568175 | 5.618×10^{-14} | 1.13033 | 5.66843×10^{-4} |
| 0.16 | 1.1465454899898728 | 1.1465454899896865 | 1.863×10^{-13} | 1.14741 | 8.68596×10^{-4} |
| 0.18 | 1.1628732662139456 | 1.1628732662134090 | 5.367×10^{-13} | 1.16414 | 1.26832×10^{-3} |
| 0.20 | 1.1787359086363027 | 1.1787359086349205 | 1.382×10^{-12} | 1.18052 | 1.78257×10^{-3} |

From Table 4, RGT with accelerated Adomian gives better accuracy compared with the HAM and Laplace transform-Adomian method. Although, RGT with accelerated Adomian polynomials uses less iterations than HAM.

Example 4. Consider the following nonlinear VIDE [4]:

$$y'(x) = \frac{3}{2}e^x - \frac{1}{2}e^{3x} + \int_0^x e^{x-t} y^3(t) dt, \quad y(0) = 1,$$

whose exact solution takes the form $y(x) = e^x$.

This example is solved by Almousa et al. [4], and they used a hybrid ADM with Modified Bernstein Polynomials by using the ADM for the nonlinear term. We applied our hybrid method for combining the RGT and the accelerated Adomian polynomial. Four iterations are carried out, and the

approximate series solution and absolute error are evaluated at the corresponding points as in [4].

We note that the integral term uses the RGT convolution theorem of two functions e^x and $y^3(x)$ [12].

Applying RGT for both sides, we get

$$\begin{aligned} RG[y'(x)] &= RG\left[\frac{3}{2}e^x\right] - RG\left[\frac{1}{2}e^{3x}\right] + RG\left[e^x \otimes y^3(x)\right], \\ RG[y'(x)] &= RG\left[\frac{3}{2}e^x\right] - RG\left[\frac{1}{2}e^{3x}\right] + uRG[e^x]RG[y^3(x)], \\ \frac{s}{u}RG[y(x)] - \frac{1}{u}y(0) &= \frac{3}{2s-2u} - \frac{1}{2s-6u} + \frac{u}{s-u}RG[y^3(x)], \\ \frac{s}{u}RG[y(x)] &= \frac{1}{u} + \frac{3}{2s-2u} - \frac{1}{2s-6u} + \frac{u}{s-u}RG[y^3(x)], \\ RG[y(x)] &= \frac{1}{s} + \frac{3u}{2s(s-u)} - \frac{u}{2s(s-3u)} + \frac{u^2}{s(s-u)}RG[y^3(x)]. \end{aligned}$$

Applying the inverse RGT for both sides, we get

$$y(x) = \frac{1}{6}(-2 + 9e^x - e^{3x}) + RG^{-1}\left[\frac{u^2}{s(s-u)}RG[y^3(x)]\right],$$

$$\text{Let } y(x) = \sum_{n=0}^{\infty} y_n(x), \quad y^3(x) = \sum_{n=0}^{\infty} A_n(x),$$

$$\sum_{n=0}^{\infty} y_n(x) = \frac{1}{6}(-2 + 9e^x - e^{3x}) + RG^{-1}\left[\frac{u^2}{s(s-u)}RG\left[\sum_{n=0}^{\infty} A_n(x)\right]\right].$$

By comparing both sides and using the Taylor series from 0 to 6, we get

$$\begin{aligned} y_0(x) &= 1 + x - \frac{x^3}{2} - \frac{x^4}{2} - \frac{13x^5}{40} - \frac{x^6}{6} + \dots, \\ y_{n+1}(x) &= RG^{-1}\left[\frac{u^2}{s(s-u)}RG\left[\sum_{n=0}^{\infty} A_n(x)\right]\right]. \end{aligned}$$

Using accelerated Adomian polynomials, we have

$$\begin{aligned}
\bar{A}_0 &= y_0^3, \\
\bar{A}_1 &= 3y_0^2 y_1 + 3y_0 y_1^2 + y_1^3, \\
\bar{A}_2 &= 3y_0^2 y_2 + 6y_0 y_1 y_2 + 3y_1^2 y_2 + 3y_0 y_2^2 + 3y_1 y_2^2 + y_2^3, \\
\bar{A}_3 &= 3y_0^2 y_3 + 6y_0 y_1 y_3 + 3y_1^2 y_3 + 6y_0 y_2 y_3 + 6y_1 y_2 y_3 + 3y_2^2 y_3 \\
&\quad + 3y_0 y_3^2 + 3y_1 y_3^2 + 3y_2 y_3^2 + y_3^3, \\
&\quad \vdots
\end{aligned}$$

Then

$$\begin{aligned}
y_0(x) &= 1 + x - \frac{x^3}{2} - \frac{x^4}{2} - \frac{13x^5}{40} - \frac{x^6}{6} + \dots, \\
y_1(x) &= \frac{x^2}{2} + \frac{2x^3}{3} + \frac{5x^4}{12} + \frac{7x^5}{120} - \frac{101x^6}{720} + \dots, \\
y_2(x) &= \frac{x^4}{8} + \frac{11x^5}{40} + \frac{71x^6}{240} + \dots, \\
y_3(x) &= \frac{x^6}{80} + \dots, \\
y_4(x) &= \frac{3x^8}{4480} + \frac{x^9}{896} + \dots, \\
y(x) (\text{approximate}) &= y_0 + y_1 + y_2 + y_3 + y_4.
\end{aligned}$$

Table 5: Comparison of the approximate solutions and absolute error against Hybrid ADM with modified Bernstein polynomials [4]

| x | Exact solution | Approximate solution (presented method, 4 iterations) | Absolute error | Approximate solution [4] (4 iterations) | Absolute error |
|-----|-------------------|---|-------------------------|---|------------------------|
| 0.0 | 1.000000000000000 | 1.000000000000000 | 0 | 1.000000 | 0 |
| 0.1 | 1.105170918075648 | 1.105170918063368 | 1.228×10^{-11} | 1.105170917 | 1.333×10^{-9} |
| 0.2 | 1.221402758160170 | 1.221402757841270 | 3.189×10^{-10} | 1.221402667 | 9.133×10^{-8} |
| 0.3 | 1.349858807576003 | 1.349858828402902 | 2.083×10^{-8} | 1.349857750 | 1.058×10^{-6} |
| 0.4 | 1.491824697641270 | 1.491825086984127 | 3.893×10^{-7} | 1.491818667 | 6.031×10^{-6} |
| 0.5 | 1.648721270700128 | 1.648724413674975 | 3.143×10^{-6} | 1.648697917 | 2.335×10^{-5} |
| 0.6 | 1.822118800390509 | 1.822135294857143 | 1.649×10^{-5} | 1.822048000 | 7.080×10^{-5} |
| 0.7 | 2.013752707470477 | 2.013818459141493 | 6.575×10^{-5} | 2.013571417 | 1.813×10^{-4} |
| 0.8 | 2.225540928492468 | 2.225756899555555 | 2.16×10^{-4} | 2.225130667 | 4.103×10^{-4} |
| 0.9 | 2.459603111156950 | 2.460217010731026 | 6.139×10^{-4} | 2.458758250 | 8.449×10^{-4} |
| 1.0 | 2.718281828459045 | 2.719841269841270 | 1.559×10^{-3} | 2.716666667 | 1.615×10^{-3} |

According to Table 5, the proposed method is both more accurate and computationally simpler than the Adomian hybrid decomposition method with modified Bernstein polynomials, which involves extensive calculations when the same number of iterations is used.

Example 5. Consider the nonlinear VIDE [28]

$$y^{(4)}(x) = e^{-3x} + e^{-x} - 1 + 3 \int_0^x y^3(t) dt ,$$

with the conditions

$$y(0) = y''(0) = 1, \quad y'(0) = y'''(0) = -1 ,$$

whose exact solution takes the form $y(x) = e^{-x}$.

This example is solved by Sharif, Hamoud, and Ghadle [28] using a Laplace and modified homotopy perturbation method (MHPM) by using the ADM for the nonlinear term. We applied our hybrid method for combining the RGT and the accelerated Adomian polynomial. Three iterations are carried out, and the approximate series solution and absolute error are evaluated at the corresponding points as in [28].

Applying RGT for both sides, we get

$$\begin{aligned} RG[y^{(4)}(x)] &= RG[e^{-3x}] + RG[e^{-x}] - RG[1] + 3RG\left[\int_0^x y^3(t) dt\right], \\ \frac{s^4}{u^4}RG[y(x)] - \frac{s^3}{u^4}y(0) - \frac{s^2}{u^3}y'(0) - \frac{s}{u^2}y''(0) - \frac{1}{u}y'''(0) \\ &= \frac{1}{(s+3u)} + \frac{1}{(s+u)} - \frac{1}{s} + 3RG\left[\int_0^x y^3(t) dt\right], \\ \frac{s^4}{u^4}RG[y(x)] &= \frac{s^3}{u^4} - \frac{s^2}{u^3} + \frac{s}{u^2} - \frac{1}{u} - \frac{1}{s} + \frac{1}{(s+3u)} + \frac{1}{(s+u)} + 3RG\left[\int_0^x y^3(t) dt\right], \\ RG[y(x)] &= \frac{1}{s} - \frac{u}{s^2} + \frac{u^2}{s^3} - \frac{u^3}{s^4} - \frac{u^4}{s^5} + \frac{u^4}{s^4(s+3u)} + \frac{u^4}{s^4(s+u)} \\ &\quad + \frac{3u^4}{s^4}RG\left[\int_0^x y^3(t) dt\right], \end{aligned}$$

Applying the inverse RGT for both sides, we get

$$y(x) = -\frac{1}{81} + \frac{e^{-3x}}{81} + e^{-x} + \frac{x}{27} - \frac{x^2}{18} + \frac{x^3}{18} - \frac{x^4}{24} + RG^{-1}\left[\frac{3u^4}{s^4}RG\left[\int_0^x y^3(t) dt\right]\right],$$

$$y^3(t) = \sum_{n=0}^{\infty} A_n ,$$

$$\sum_{n=0}^{\infty} y_n(x) = -\frac{1}{81} + \frac{e^{-3x}}{81} + e^{-x} + \frac{x}{27} - \frac{x^2}{18} + \frac{x^3}{18} - \frac{x^4}{24} \\ + RG^{-1} \left[\frac{3u^4}{s^4} RG \left[\int_0^x \sum_{n=0}^{\infty} A_n dx \right] \right],$$

By comparing both sides and using the Taylor series from 0 to 5, we get

$$y_0(x) = 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \frac{x^5}{30} + \dots,$$

$$y_{n+1}(x) = RG^{-1} \left[\frac{3u^4}{s^4} RG \left[\int_0^x A_n dx \right] \right].$$

Using accelerated Adomian polynomials, we have

$$\bar{A}_0 = y_0^3, \quad \bar{A}_1 = 3y_0^2y_1 + 3y_0y_1^2 + y_1^3,$$

$$\bar{A}_2 = 3y_0^2y_2 + 6y_0y_1y_2 + 3y_1^2y_2 + 3y_0y_2^2 + 3y_1y_2^2 + y_2^3,$$

$$\vdots$$

Then

$$y_0(x) = 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \frac{x^5}{30} + \dots,$$

$$y_1(x) = \frac{x^5}{40} - \frac{x^6}{80} + \frac{3x^7}{560} - \frac{9x^8}{4480} + \dots,$$

$$y_2(x) = \frac{x^{10}}{134400} - \frac{x^{11}}{98560} + \frac{3x^{12}}{394240} + \dots,$$

$$y_3(x) = \frac{x^{15}}{5381376000} + \dots,$$

$$y(x) (\text{approximate}) = y_0 + y_1 + y_2 + y_3.$$

Based on Table 6, the proposed method outperforms the others at both two and three iterations specifically when two iterations are used in the MHPM and three iterations in the LADM [15].

Table 6: Comparison of the approximate solution against LADM [17] and MHPM [7]

| x | Exact solution | Approximate solution for presented method using three iterations | Absolute Error for presented method using three iterations | LADM [17] | Absolute Error for LADM [17] | MHPM [7] | Absolute Error for MHPM [7] |
|------|----------------|--|--|--------------|------------------------------|--------------|-----------------------------|
| 0.00 | 1.0000000000 | 1.0000000000 | 0.0000×10^0 | 1.0000000000 | 0.0000×10^0 | 1.0000000000 | 0.0000×10^0 |
| 0.04 | 0.9607894392 | 0.9607894391 | 5.5992×10^{-11} | 0.9607895450 | 1.0580×10^{-7} | 0.9608106692 | 2.1230×10^{-5} |
| 0.08 | 0.9231163464 | 0.9231163429 | 3.5278×10^{-9} | 0.9231180530 | 1.7066×10^{-6} | 0.9232854120 | 1.6906×10^{-4} |
| 0.12 | 0.8869204367 | 0.8869203971 | 3.9569×10^{-8} | 0.8869290770 | 8.6403×10^{-6} | 0.8874885866 | 5.6815×10^{-4} |
| 0.16 | 0.8521437890 | 0.8521435700 | 2.1898×10^{-7} | 0.8521710940 | 2.7305×10^{-5} | 0.8534850921 | 1.3413×10^{-3} |
| 0.20 | 0.8187307531 | 0.8187299301 | 8.2298×10^{-7} | 0.8187974190 | 6.6670×10^{-5} | 0.8213405980 | 2.6098×10^{-3} |
| 0.24 | 0.7866278611 | 0.7866254393 | 2.4218×10^{-6} | 0.7867661000 | 1.3824×10^{-4} | 0.7911217470 | 4.4938×10^{-3} |
| 0.28 | 0.7557837415 | 0.7557777214 | 6.0201×10^{-6} | 0.7560398470 | 2.5611×10^{-4} | 0.7628963355 | 7.1125×10^{-3} |
| 0.32 | 0.7261490371 | 0.7261358094 | 1.3228×10^{-5} | 0.7265859450 | 4.3691×10^{-4} | 0.7367334755 | 1.0584×10^{-2} |
| 0.36 | 0.6976763261 | 0.6976498733 | 2.6453×10^{-5} | 0.6983761680 | 6.9984×10^{-4} | 0.7127037390 | 1.5027×10^{-2} |

6 Conclusions

In this study, a hybrid approach combining the RGT with the accelerated Adomian polynomial is introduced to solve IDEs numerically. The resulting method is straightforward and efficient, as demonstrated by the numerical results presented in the tables.

These results highlighted the improved accuracy achieved through this combination, outperforming other existing methods. An important advantage and as a key contribution of the proposed convergence analysis is the use of the classical fixed-point theorem in conjunction with accelerated polynomials, rather than traditional polynomials.

This approach enhanced the robustness and efficiency of analysis. All computations were carried out using MATHEMATICA 12.

Acknowledgements

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

References

- [1] Adomian, G. *Nonlinear stochastic systems theory and applications to physics*, Kluwer Academic Publishers, Dordrecht, 1989.

- [2] Al-Hayani, W. *Combined Laplace transform-homotopy perturbation method for Sine-Gordon equation*, Appl. Math. Inf. Sci. 10(5) (2016), 1–6.
- [3] Al-Hayani, W. and Younis, M.T. *The homotopy perturbation method for solving nonlocal initial-boundary value problems for parabolic and hyperbolic partial differential equations*, Eur. J. Pure Appl. Math. 16(3) (2023), 155–156.
- [4] Almousa, M., Saidat, S., Al-Hammouri, A., Alsaadi, S. and Banihani, G. *Solutions of nonlinear integro-differential equations using a hybrid Adomian decomposition method with modified Bernstein polynomials*, Int. J. Fuzzy Log. Intell. Syst. 24(3) (2024) 271–279.
- [5] Avudainayagam, A. and Vani, C. *Wavelet–Galerkin method for integro-differential equations*, Appl. Math. Comput. 32 (2000), 247–254.
- [6] Bahuguna, D., Ujlayan, A. and Pandey, D.N. *A comparative study of numerical methods for solving an integro-differential equation*, Comput. Math. Appl. 57 (2009), 1485–1493.
- [7] Behzadi, S., Abbasbandy, S., Allahviranloo, T. and Yildirim, A. *Application of homotopy analysis method for solving a class of nonlinear Volterra-Fredholm integro-differential equations*, J. Appl. Anal. Comput. 2(2) (2012), 127–136.
- [8] Biazar, J., Babolian, E. and Islam, R. *Solution of a system of Volterra integral equations of the first kind by Adomian method*, Appl. Math. Comput. 139 (2003), 249–258.
- [9] Biazar, J., Ghazvini, H. and Eslami, M. *He’s homotopy perturbation method for systems of integro-differential equations*, Chaos Solitons Fractals 39 (2007), 1253–1258.
- [10] Duan, J., Rach, R., Baleanu, D. and Wazwaz, A. *A review of the Adomian decomposition method and its applications to fractional differential equations*, Commun. Frac. Calc. 3(2) (2012), 73–99.


- [11] El-Kalla, I.L. *Error analysis of Adomian series solution to a class of nonlinear differential equations*, Appl. Math. E-Notes 7 (2007), 214–221.
- [12] El-Kalla, I.L. *New results on the analytic summation of Adomian series for some classes of differential and integral equations*, Appl. Math. Comput. 217 (2010), 3756–3763.
- [13] El-Kalla, I.L. *Piece-wise continuous solution to a class of nonlinear boundary value problem*, Ain Shams Eng. J. 4 (2013), 325–331.
- [14] Golberg, A.M. *Solution methods for Integral Equations: Theory and Applications*, Plenum Publishing Corporation, New York, 1979.
- [15] He, J.H. *A coupling method of a homotopy technique and a perturbation technique for nonlinear problems*, Int. J. Non-Linear Mech. 35(1) (2000), 37–43.
- [16] Jerri, A.J. *Introduction to integral equations with applications*, Marcel Dekker, New York, 1999.
- [17] Khanlari, N. and Paripour, M. *Solving nonlinear integro-differential equations using the combined homotopy analysis transform method with Adomian polynomials*, RGN Publ. 9(4) (2018), 637–650.
- [18] Maleknejad, K., Mirzaee, F. and Abbasbandy, S. *Solving linear integro-differential equations system by using rationalized Haar functions method*, Appl. Math. Comput. 155 (2004), 317–328.
- [19] Maleknejad, K. and Tavassoli Kajani, M. *Solving linear integro-differential equation system by Galerkin methods with hybrid functions*, Appl. Math. Comput. 159 (2004), 603–612.
- [20] Ramadan, M.A. *The convolution for Ramadan group integral transform: Theory and applications*, J. Adv. Trends Basic Appl. Sci. (2017), 191–197.
- [21] Ramadan, M.A., Mansour, M.M.A., El-Shazly, N.M. and Osheba, H.S. *A blended numerical procedure for quadratic Riccati differential equations utilizing Ramadan group transform and variations of Adomian decomposition*, Eng. Appl. Sci. Lett. 7(3) (2024), 11–25.

- [22] Ramadan, M.A., Mansour, M.M.A., El-Shazly, N.M. and Osheba, H.S. *The double Ramadan group accelerated Adomian decomposition method for solving nonlinear partial differential equations*, Comput. Methods Differ. Equ. (2025).
- [23] Ramadan, M.A., Mansour, M.M.A. and El-Shazly, N.M. *Semi-analytic solution of the nonlinear Sharma-Tasso-Olver equation via Ramadan group integral transform and accelerated Adomian decomposition*, J. Umm Al-Qura Univ. Appl. Sci. (2025) 1–14.
- [24] Ramadan, M.A. and Mesrega, A.K. *Solution of partial and integro-differential equations using the convolution of Ramadan group transform*, Asian Res. J. Math. 11(3) (2018), 1–15.
- [25] Rani, D. and Mishra, V. *Modification of Laplace Adomian decomposition method for solving nonlinear Volterra integral and integro-differential equations based on Newton Raphson Formula*, Eur. J. Pure Appl. Math. 11(1) (2018), 202–214.
- [26] Sadeghi Goghary, H., Javadi, Sh. and Babolian, E. *Restarted Adomian method for system of nonlinear Volterra integral equations*, Appl. Math. Comput. 161 (2005), 745–751.
- [27] Sayed, A.Y., Sayed, E.A., Rashwan, M.H. and El-Kalla, I.L. *Using an accelerated technique of the Laplace Adomian decomposition method in solving a class of nonlinear integro-differential equations*, Eng. Res. J. 175 (2022), 371–385.
- [28] Sharif, A.A., Hamoud, A.A. and Ghadle, K.P. *Solving nonlinear integro-differential equations by using numerical technique*, Acta Univ. Apulensis 61 (2019), 45–53.
- [29] Wang, S.Q. and He, J.H. *Variational iteration method for solving integro-differential equations*, Phys. Lett. A 367 (2007), 188–191.
- [30] Younis, M.T. and Al-Hayani, W. *Solving fuzzy system of Volterra integro-differential equations by using Adomian decomposition method*, Eur. J. Pure Appl. Math. 15(1) (2022), 290–313.

- [31] Younis, M.T. and Al-Hayani, W. *A numerical study for solving the systems of fuzzy Fredholm integral equations of the second kind using the Adomian decomposition method*, Iraqi J. Sci. 64(7) (2023), 4407–4430.
- [32] Yusufoglu, E. (Agadjanov) *An efficient algorithm for solving integro-differential equations system*, Appl. Math. Comput. 192 (2007), 51–55.
- [33] Zhao, J. and Corless, R.M. *Compact finite difference method for integro-differential equations*, Appl. Math. Comput. 1(6) (2006), 271–288.



Efficient numerical schemes on modified graded mesh for singularly perturbed parabolic convection-diffusion problems

K.K. Sah*, 

Abstract

In this study, numerical approaches to the singularly perturbed problems of convection diffusion type are presented. The backward Euler method is applied to a uniform mesh in the temporal domain, while in the spatial domain, we utilize both the hybrid midpoint finite difference scheme and the high order via differential identity expansion scheme on a modified graded mesh. The solution to the problem introduces a boundary layer on the right side of the domain. Both of the above methods are proven to have identical convergence with respect to the perturbation parameter. We also provide numerical results in order to verify the theoretical conclusions. We demonstrate that the applied approaches provide uniform convergence of first-order in the temporal variable and second-order up to a logarithmic factor with respect to the spatial variable.

AMS subject classifications (2020): 65M06, 65M12, 65M15.

*Corresponding author

Received 14 May 2025; revised 3 August 2025; accepted 14 August 2025

Kishun Kumar Sah

Department of Mathematics, National Institute of Technology Patna, India.

e-mail: kishuns.phd20.ma@nitp.ac.in

How to cite this article

Sah, K.K., Efficient numerical schemes on modified graded mesh for singularly perturbed parabolic convection-diffusion problems. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1361-1391. <https://doi.org/10.22067/ijnao.2025.93523.1646>

Keywords: Perturbation problems; Uniform convergence; Modified graded mesh, Boundary layers; Hybrid midpoint finite difference scheme; HODIE finite difference scheme

1 Introduction and summary

A singular perturbation problem in the context of parabolic partial differential equations involves a small parameter (usually denoted by ε) multiplying the highest-order time derivative term in the equation. One example of a singular perturbation problem is the parabolic convection-diffusion equation, which models the transport of a scalar quantity, such as heat or chemical concentration, in a fluid medium that is subject to both diffusion and convection. The equation takes the form: $\varepsilon u_t = Du_{xx} - vu_x$, where $u(x, t)$ is the scalar quantity being transported and ε is the small parameter that measures the relative strength of diffusion to convection. This equation is called the parabolic convection-diffusion equation because it is a parabolic partial differential equation that combines convection and diffusion terms. The convection term vu_x represents the transport of the scalar quantity by the fluid flow, while the diffusion term Du_{xx} represents the spreading of the scalar quantity due to molecular diffusion. The singular perturbation aspect of this problem arises because the εu_t term introduces a time scale that is much faster than the time scale of the diffusion and convection terms. As a result, the solution to this equation exhibits behavior that is very different depending on whether ε is small or not. These problems frequently occur in a variety of applied mathematics fields, including fluid dynamics, elasticity, and many others. The study of singular perturbation problems like the parabolic convection-diffusion equation is important in many fields, including fluid dynamics, chemical engineering, and mathematical biology. Techniques for analyzing these problems include matched asymptotic expansions, boundary layer theory, and numerical methods that accurately depict the solution's behavior as the parameter ε tends towards zero. When the value of ε is small, the problem exhibits heightened sensitivity to alterations in initial or boundary conditions, rendering conventional numerical methods for solving parabolic equations seemingly insufficient and imprecise.

Addressing singular perturbation problems necessitates the application of specialized techniques like asymptotic analysis, matched asymptotic expansions, or numerical methods explicitly tailored for these specific types of problems. The goal is to accurately capture the behavior of the solution in the boundary layer regions while avoiding excessive computational cost or numerical instability. Approximate solutions are required in these situations since it is often impossible or very difficult to find the precise answer to these mathematical issues. By using perturbation techniques, it is possible to find a rough answer. These approaches fundamental premise is to start by finding a solution to a reduced problem and thereafter get consistently excellent estimates. The solution of the singular perturbation in the parabolic partial differential equations relies on both the resolution at the previous stage and the resolution at the present stage; it is more analogous to events that occur in the actual world. Many publications addressing singularly perturbed parabolic problems are accessible in the literature.

For instance, Claver, Gracia, and Jorge [3] developed high-order numerical methods for one-dimensional parabolic singularly perturbed problems, providing valuable insights into handling regular and singular layers. Clavero, Gracia, and Lisbona [5] extended these methods by implementing higher-order schemes on Shishkin meshes for convection-diffusion problems, ensuring uniform convergence. Izadi and Yuzbasi [8] proposed a hybrid approximation scheme that effectively tackled convection-diffusion problems with singular perturbations. Mukherjee and Natesan [17] introduced parameter-uniform hybrid schemes for convection-dominated initial-boundary-value problems, while their subsequent work [18] employed Richardson extrapolation techniques to enhance solution accuracy and robustness. Furthermore, Tia, Liu, and An [22] devised a higher-order finite difference scheme for singularly perturbed parabolic problems, emphasizing improved computational efficiency.

Despite these advancements, analytical solutions to singularly perturbed differential equations remain challenging due to the inherent complexity of boundary and interior layers.

Furthermore, the problem of the solution displays border and interior layers with a modest perturbation parameter of ε . Also, on a uniform mesh, the classical numerical technique suddenly needs a lot of mesh points to correctly

represent the layer in the solution, which is not feasible. In this sense, the aforementioned approach is unsuccessful. So, a uniform convergent approach has been developed as a result of the specific consideration needed for the numerical solution of singularly perturbed partial differential equations.

There are numerous studies focused on the analytical and numerical treatment of singularly perturbed parabolic problems, particularly utilizing finite difference and finite element methods. For instance, Cai and Liu [1] proposed a Reynolds-uniform scheme for addressing such problems, emphasizing uniform convergence. Chi-kuang [2] applied finite element methods to tackle singular perturbation problems, showcasing their versatility in handling boundary layers. Moreover, Clavero, Jorge, and Lisbona [4] developed uniformly convergent schemes that integrated alternating directions and exponential fitting techniques, enhancing solution accuracy. Kadalbajoo and Yadaw [9] investigated parameter-uniform finite element methods for two-parameter problems, extending their applicability to reaction-diffusion systems. Additionally, Kumar and Vigo-Aguiar [15] devised a parameter-uniform grid equidistribution method, offering improved robustness in degenerate parabolic problems. Sun and Stynes [21] employed finite element methods for high-order elliptic singularly perturbed problems. However, analytical solutions and numerical approaches for singular perturbation convection diffusion problems are only briefly explored in a few papers. Mukherjee and Natesan [17] proposed hybrid numerical schemes that maintain uniform convergence in convection-diffusion settings. Vulcanović and Nhan [24] advanced this by developing robust higher-order hybrid schemes, which effectively handle steep gradients. Similarly, the higher-order monotone schemes designed by Vulcanović [23] demonstrate significant accuracy in nonlinear singular perturbation problems. In terms of the diffusion parameter, the numerical technique is uniformly convergent, with an order close to two in space, but in all these works, the authors have described a singularly perturbed parabolic problem on a Shishkin mesh only.

There are currently no known papers relating to the convergence of difference schemes on modified graded meshes. As a result, we are now in a position to develop a different scheme for a modified graded mesh. Motivated by the work of Claver, Gracia, and Jorge [3], who developed high-

order numerical methods for singularly perturbed problems on layer-adapted meshes, Kaushik et al. [10], who introduced a modified graded mesh for singularly perturbed reaction-diffusion problems, achieving enhanced accuracy, Mukherjee and Natesan [18], who demonstrated robust convergence for convection-diffusion problems by using the Richardson extrapolation technique, Clavero, Gracia, and Stynes [6], who provided a simplified analysis of hybrid numerical methods, and Kaushik et al. [11], who applied higher-order methods to two-parameter singular perturbation problems, we aim to extend this research direction.

In this article, we propose two finite difference schemes: the hybrid midpoint finite difference scheme and the high order via differential identity expansion (HODIE) finite difference scheme on a modified graded mesh for the convection-diffusion parabolic problem. Consider the singularly perturbed initial-boundary value problem:

$$\left\{ \begin{array}{ll} \frac{\partial y(r, \theta)}{\partial \theta} + \mathcal{L}_\varepsilon y = f(r, \theta) & \text{on } \Lambda := \Lambda_r \times \Lambda_\theta, \\ & \text{where } \Lambda_r = (0, 1) \text{ and } \Lambda_\theta = (0, \mathcal{T}], \\ y(r, 0) = y_0(r) & \text{for } 0 \leq r \leq 1, \\ y(0, \theta) = 0 & \text{for } 0 < \theta \leq \mathcal{T}, \\ y(1, \theta) = 0 & \text{for } 0 < \theta \leq \mathcal{T}, \end{array} \right. \quad (1)$$

where

$$\mathcal{L}_\varepsilon y(r, \theta) \equiv -\varepsilon \frac{\partial^2 y(r, \theta)}{\partial r^2} + \kappa_1(r) \frac{\partial y(r, \theta)}{\partial r} + \kappa_2(r, \theta) y(r, \theta), \quad (2)$$

with $\kappa_1(r) > \lambda > 0$ and $\kappa_2 = \kappa_2(r, \theta) \geq 0$ on $\bar{\Lambda}$, where ε is a small perturbation. In section 2, there will be more presumptions made regarding the problem of the data. From (8), it can be found that the solution y of (1) contains an exponential boundary layer at the side $r = 1$ of Λ . Throughout this paper, we concentrate on two finite difference techniques (hybrid difference scheme and second-order HODIE) for (1) that were introduced and examined in [3, 17]. These studies verify convergence for these approaches, uniformly in ε , with the caveat that $\kappa_2 = \kappa_2(r)$, but the mesh is the same in both papers.

Our main goal in this study is to suggest and examine a higher-order hybrid finite difference strategy for the problem (1) on the modified graded mesh, which shall be discussed in the forthcoming section. In Section 3, we define the meshes for temporal and spatial discretization and introduce some special difference operators and the finite difference scheme. Also, we will prove that the methods finite difference techniques (hybrid difference scheme and second-order HODIE) of [3, 17] are essentially the same. In Section 4, we show the convergence of these numerical techniques, uniformly in ε when applied to (1). In Section 5, we present the numerical results for two linear test problems to validate the theoretical results. Finally, in Section 6, we summarize the main conclusions.

The functions based on the mesh assumption (16), which proves to be considerably less limiting compared to the mesh constraint $\mathcal{N}^{-k} \leq \mathcal{C}\Delta\theta$ imposed in [3, 17], where $k \in (0, 1)$. When $\varepsilon \leq \mathcal{N}^{-1}$, our convergence result Theorem 1 becomes

$$\max_{i,j} |y(r_i, \theta_j) - Y_i^j| \leq \mathcal{C}[\Delta\theta + (\mathcal{N}^{-1} \ln(1/\varepsilon))^2]. \quad (3)$$

This sharpens the weaker result

$$\max_{i,j} |y(r_i, \theta_j) - Y_i^j| \leq \mathcal{C}[\Delta\theta + \mathcal{N}^{-2+k}(\ln 1/\varepsilon)^2].$$

It was obtained from [3, 17]. The numerical findings shown in these papers demonstrate that the factor \mathcal{N}^k in this instance is an antiquity of the analysis; that is, that our bound (3) is sharp. In section 5, we provide yet another numerical example to demonstrate the accuracy of our convergence results. In section 6, some final conclusions are given.

Notation: Throughout the paper, the symbol \mathcal{C} represents a general positive constant that remains unaffected by both ε and the mesh size.

2 Assumptions on the data

Before we analyze the problem, some of the compatibility conditions are necessary. Therefore, the following compatibility conditions at the corners

for functions and their zero-order and first-order derivatives are assumed to satisfy:

$$\begin{cases} y_0(0) = y_0(1) = 0, \\ -\varepsilon y_0''(0) + \kappa_1(0)y_0'(0) = f(0,0), \\ -\varepsilon y_0''(1) + \kappa_1(1)y_0'(1) = f(1,0). \end{cases} \quad (4)$$

Then (1) has a unique solution in the Holder space $\mathcal{C}^{2+\lambda,1+\lambda/2}(\bar{\Lambda})$ see in [19, 7]. We also make the assumption that the corner compatibility conditions of second order are met, ensuring the validity of $\mathcal{C}^{4+\lambda,2+\lambda/2}(\bar{\Lambda})$. These conditions can be explicitly stated within the terms of the problem of data in the following manner. Differentiating (1) with respect to θ we get

$$f_\theta = y_{\theta\theta} + \mathcal{L}_\varepsilon y_\theta + \kappa_{2\theta} y = y_{\theta\theta} + \mathcal{L}_\varepsilon (f - \mathcal{L}_\varepsilon y) + \kappa_{2\theta} y.$$

Therefore, by invoking (1) and (4), we can express the second-order corner compatibility conditions as

$$\mathcal{L}_\varepsilon(\mathcal{L}_\varepsilon y_0) = \mathcal{L}_\varepsilon f - f_\theta \quad (5)$$

at the corners $(0,0)$ and $(1,0)$. Given these assumptions, the solution y to (1) exhibits an exponential layer along the boundary $r = 1$ of Λ and adheres to the specified bound

$$\left| \frac{\partial^{s+l} y(r, \theta)}{\partial r^s \partial \theta^l} \right| \leq \mathcal{C}(1 + \varepsilon^{-s} e^{-\lambda(1-r)/\varepsilon}) \quad \text{for } (r, \theta) \in \bar{\Lambda} \text{ and } s + 2l \leq 4. \quad (6)$$

This result was proved in [25] for $0 \leq s+l \leq 2$. Under necessary compatibility conditions and sufficient smoothness on the data, the proof of the estimate (6) for higher values of s, l follows similarly from [3, Lemma 2.1]. The approaches given in [19] may be used to prove the aforementioned bound. The inequality (6) a priori is sufficient for the majority of our analysis. It becomes necessary for us to additionally assume that the data of the problem (1) adhere to the third-order compatibility condition

$$f_{\theta\theta} = \mathcal{L}_\varepsilon(f_\theta - \mathcal{L}_\varepsilon(f - \mathcal{L}_\varepsilon y_0) - \kappa_{2\theta} y_0) \text{ at the corners } (0,0) \text{ and } (1,0). \quad (7)$$

Then, similarly to (6), the bounds on the derivatives can be shown as

$$\left| \frac{\partial^{s+l} y(r, \theta)}{\partial r^s \partial \theta^l} \right| \leq \mathcal{C}(1 + \varepsilon^{-s} e^{-\lambda(1-r)/\varepsilon}) \quad \text{for } (r, \theta) \in \overline{\Lambda} \text{ and } s + 2l \leq 6. \quad (8)$$

In [3, 17], authors assumed that (8) is valid for $s + l \leq 4$, $l \leq 2$.

Remark 1. The order of convergence of the our numerical technique on a modified graded mesh, applied the finite difference scheme (midpoint technique), and the HODIE finite difference scheme is unaffected when (7) is broken, according to the findings of our calculations. For an illustration, see section 5.

Remark 2. As the variable ε can assume a range of values, the compatibility condition (4) indicates that

$$\begin{cases} y_0(0) = y_0(1) = 0, \\ \kappa_1(0)y_0'(0) = f(0, 0), \\ \kappa_1(1)y_0'(1) = f(1, 0), \\ y_0''(0) = y_0''(1) = 0. \end{cases} \quad (9)$$

Likewise, by utilizing (9), it becomes apparent that the equivalence of (5) is contingent upon the condition of requiring.

$$\begin{cases} (\kappa_1' + \kappa_2)f = \kappa_1 f_r - f_\theta, \\ (\kappa_1'' + 2\kappa_{2r})y_0' = f_{rr}, \\ y_0^A = 0 \end{cases}$$

at the corners $(0, 0)$ and $(1, 0)$.

Further requirements are imposed on the data by assumption (7), although as Remark 1 shows, they may not be necessary in reality. Despite the fact that these requirements place limits on the types of data that are allowed, it is nonetheless evident that some types of data meet these requirements. For instance, if enough derivatives of the y_0 and f disappear at the corners $(0, 0)$ and $(1, 0)$.

3 Numerical discretization

Grids for spatial and temporal direction and bounds on them are defined in this section. We apply two finite difference schemes (the hybrid difference scheme and second-order HODIE) for the spatial derivative and the Euler-backward difference for the temporal derivative to discretize the problem (1).

3.1 The uniform mesh

In the time domain interval $[0, \mathcal{T}]$, we employ a uniform mesh with a time step $\Delta\theta$, ensuring that

$$\Lambda_{\theta}^{\mathcal{M}} = \{\theta_k = k\Delta\theta, \quad k = 0, 1, \dots, \mathcal{M}, \quad \Delta\theta = \frac{\mathcal{T}}{\mathcal{M}}\},$$

Here, \mathcal{M} represents the number of mesh points in the θ -direction within the interval $[0, \mathcal{T}]$.

3.2 Spatial discretization

We generate a modified graded mesh, $\Lambda_r^{\mathcal{N}}$ in the interval $[0, 1]$ and order to resolve the boundary layer at $r = 1$, which is plotted in Figure 1 as follows:

$$\sigma_i = 1 - \chi_{\mathcal{N}-1} \quad \text{for } i = 1, \dots, \mathcal{N},$$

where χ is defined as follows:

$$\begin{cases} \chi_0 = 0, \\ \chi_i = 2\varepsilon \frac{i}{\mathcal{N}}, & 1 \leq i \leq \frac{\mathcal{N}}{2}, \\ \chi_{i+1} = \chi_i(1 + \rho h), & \frac{\mathcal{N}}{2} \leq i \leq \mathcal{N} - 2, \\ \chi_{\mathcal{N}} = 1, \end{cases} \quad (10)$$

where the parameter h satisfies the following nonlinear equation:

$$\ln(1/\varepsilon) = (\mathcal{N}/2) \ln(1 + \rho h). \quad (11)$$

The above section of the parameter h ensures that there are $\mathcal{N}/2$ grid points in the interval $[0, 1 - \varepsilon]$, which are distributed gradedly in the interval $[0, 1 - \varepsilon]$. Numerical verification stimulate us that the interval $(\chi_{\mathcal{N}-1}, 1)$ is not too small in comparison with the previous one $(\chi_{\mathcal{N}-2}, \chi_{\mathcal{N}-1})$. In the subinterval $[1 - \varepsilon, 1]$ we distribute $\mathcal{N}/2$ points with uniform step length $2\varepsilon/\mathcal{N}$, while in the subinterval $[0, 1 - \varepsilon]$ we first find h for some fix \mathcal{N} by means of the nonlinear equation (11), and corresponding to that h we distribute $\mathcal{N}/2$ points in the interval $[0, 1 - \varepsilon]$. The mesh length is denoted by $h_i = \chi_i - \chi_{i-1}$, for $i = 1, 2, \dots, \mathcal{N}$.

Remark 3. The mesh size in piecewise uniform and the modified graded region is given by

$$h_i = \begin{cases} 2\varepsilon/\mathcal{N} & \text{for } i = 1, 2, \dots, \mathcal{N}/2, \\ \rho h \chi_{i-1} & \text{for } i = \mathcal{N}/2 + 1, \mathcal{N}/2 + 2, \dots, \mathcal{N}. \end{cases}$$

Lemma 1. The mesh defined in (10) satisfies the following estimates:

$$|h_{i+1} - h_i| \leq \begin{cases} 0 & \text{for } i = 1, 2, \dots, \mathcal{N}/2, \\ \mathcal{C}h & \text{for } i = \mathcal{N}/2 + 1, \mathcal{N}/2 + 2, \dots, \mathcal{N}. \end{cases}$$

Proof. Initially, we consider $i = 1, 2, \dots, \mathcal{N}/2$. As the mesh is uniform in this portion, so nothing to prove.

For $i = \mathcal{N}/2 + 1, \mathcal{N}/2 + 2, \dots, \mathcal{N}$. We have

$$\begin{aligned} |h_{i+1} - h_i| &= |\rho h \chi_i - \rho h \chi_{i-1}| \\ &= \rho h |\chi_i - \chi_{i-1}| \\ &= \rho^2 h^2 \chi_{i-1} \\ &\leq \mathcal{C}h. \end{aligned}$$

Here, we have taken $0 < \rho, h < 1$. □

Lemma 2. For the modified graded mesh defined in (10), the parameter h satisfies the following bound:

$$h \leq \mathcal{C} \mathcal{N}^{-1} \ln(1/\varepsilon).$$

Proof. Let \mathcal{K}_1 be the number of points χ_i in the partition (10) such that $\chi_i \leq \varepsilon$, for $i = 1, 2, \dots, \mathcal{N}/2$. Clearly $\mathcal{K}_1 \leq \mathcal{C}/h$ and \mathcal{K}_2 be the number of points in the partition (10) such that $\chi_i > \varepsilon$. Let $\chi_{\mathcal{N}/2+1}$ be the smallest point such that $\chi_i > \varepsilon$. We have to estimate the bound for \mathcal{K}_2 . Assuming $\rho h \leq 1$, we have

$$\begin{aligned} \mathcal{K}_2 &= \sum_{\mathcal{N}/2+1}^{\mathcal{N}} 1 = \sum_{\mathcal{N}/2+1}^{\mathcal{N}} (\chi_{i+1} - \chi_i)^{-1} \int_{\chi_i}^{\chi_{i+1}} d\chi \\ &= \sum_{\mathcal{N}/2+1}^{\mathcal{N}} (h_{i+1})^{-1} \int_{\chi_i}^{\chi_{i+1}} d\chi \\ &= \sum_{\mathcal{N}/2+1}^{\mathcal{N}} (\rho h \chi_i)^{-1} \int_{\chi_i}^{\chi_{i+1}} d\chi \\ &\leq \sum_{\mathcal{N}/2+1}^{\mathcal{N}} (2/\rho h \chi_{i+1})^{-1} \int_{\chi_i}^{\chi_{i+1}} d\chi, \end{aligned}$$

because $\chi_{i+1} < 2\chi_i$. For any $\chi \in [\chi_i, \chi_{i+1}]$, we have

$$\begin{aligned} \mathcal{K}_2 &\leq \sum_{\mathcal{N}/2+1}^{\mathcal{N}} 2(\rho h)^{-1} \int_{\chi_i}^{\chi_{i+1}} \frac{1}{\chi} d\chi \\ &\leq 2(\rho h)^{-1} \int_{\varepsilon}^1 \frac{1}{\chi} d\chi \\ &\leq 2(\rho h)^{-1} \ln(1/\varepsilon). \end{aligned}$$

Recalling $\mathcal{N} = \mathcal{K}_1 + \mathcal{K}_2$, we have

$$\begin{aligned} \mathcal{N} &\leq \mathcal{C}/\rho h + 2(\rho h)^{-1} \ln(1/\varepsilon), \\ \mathcal{N} &\leq 1/h(\mathcal{C}\rho + 2(\rho)^{-1} \ln(1/\varepsilon)), \\ \mathcal{N} &\leq 1/h(\mathcal{C} \ln(1/\varepsilon)), \end{aligned}$$

Finally, we get

$$h \leq \mathcal{C} \mathcal{N}^{-1} \ln(1/\varepsilon),$$

where \mathcal{N} is the number of grid points in the r -direction. \square

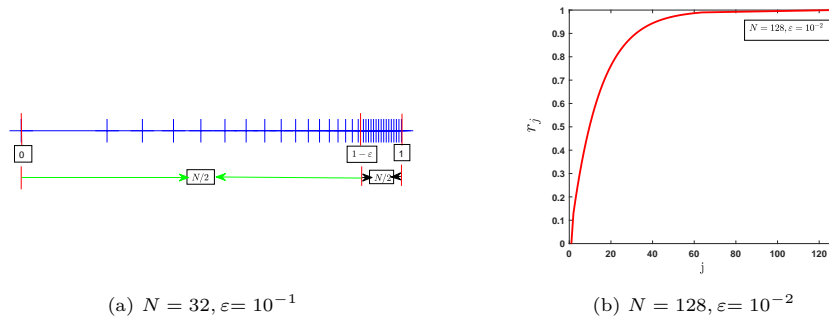


Figure 1: Distribution of modified graded mesh points for the problems with the layer on the right side of the boundary, that is, $r = 1$, which is plot in Figure 1.

Figure 1a shows the distribution of the domain $[0, 1]$, while Figure 1b illustrates the layer within the domain $[0, 1]$.

3.3 The finite difference scheme

In this section, we used the backward-Euler difference in the time direction and two finite difference schemes (the hybrid midpoint method and the HODIE method) for the spatial direction on the modified graded mesh. Now, a free parameter p_i^1 that is defined by the following, is utilized to characterize the second-order HODIE finite difference scheme of [3], which is used to discretize the spatial derivative of (1):

$$p_i^1 = \begin{cases} \frac{d_i}{d_{i-1} + d_i} & \text{for } i = 1, 2, 3, \dots, \mathcal{N}/2, \\ 0, & \text{for } i = \mathcal{N}/2 + 1, \dots, \mathcal{N} - 1. \end{cases} \quad (12)$$

Here, we suppose that $\mathcal{H}||d||_\infty \geq 2\varepsilon$ (see [3]). Let us denote the step sizes in space by $h_i := r_i - r_{i-1}$ and $\hat{h}_i := (h_i + h_{i+1})/2$ for all i , and $\Delta\theta = \theta_j - \theta_{j-1}$

for all j . Then, Y_i^j is the analytical solution at the grid point (r_i, θ_i) , and also we define the HODIE and midpoint difference scheme of [3] is

$$p_i^1 \frac{Y_{i-1}^j - Y_{i-1}^{j-1}}{\Delta\theta} + (1 - p_i^1) \frac{Y_i^j - Y_i^{j-1}}{\Delta\theta} + [\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} Y]_i^j = p_i^1 f_{i-1}^j + (1 - p_i^1) f_i^j \quad (13a)$$

for $i = 1, 2, 3, \dots, \mathcal{N}/2$, and

$$\frac{Y_i^j - Y_i^{j-1}}{\Delta\theta} + [\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} Y]_i^j = f_i^j, \quad \text{for } i = \mathcal{N}/2 + 1, \dots, \mathcal{N} - 1, \quad (13b)$$

where

$$\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} Y]_i^j = q_{ij}^- Y_{i-1}^j + q_{ij}^c Y_i^j + q_{ij}^+ Y_{i+1}^j \quad (13c)$$

with

$$\begin{cases} q_{ij}^- = -\frac{\varepsilon}{h_i \hat{h}_i} - \frac{2p_i^1 d_{i-1}}{h_i} + p_i^1 \kappa_{2i-1}^j, \\ q_{ij}^+ = -\frac{\varepsilon}{h_{i+1} \hat{h}_i}, \\ q_{ij}^c = -q_{ij}^- - q_{ij}^+ + p_i^1 \kappa_{2i-1}^j + (1 - p_i^1) \kappa_{2i}^j \end{cases} \quad (13d)$$

for $i = 1, 2, 3, \dots, \mathcal{N}/2$, and

$$\begin{cases} q_{ij}^- = -\frac{\varepsilon}{h_i \hat{h}_i} - \frac{d_i}{2\hat{h}_i}, \\ q_{ij}^+ = -\frac{\varepsilon}{h_{i+1} \hat{h}_i} + \frac{d_i}{2\hat{h}_i}, \\ q_{ij}^c = -q_{ij}^- - q_{ij}^+ + \kappa_{2i}^j \end{cases} \quad (13e)$$

for $i = \mathcal{N}/2 + 1, \dots, \mathcal{N} - 1$.

When $i = 1, 2, 3, \dots, \mathcal{N}/2$, it is easy to see that

$$p_i^1 = \frac{1}{2} + O(h_i), \quad \text{and} \quad 2p_i^1 d_{i-1} = d_{i-1/2} + O(h_i^2), \quad (14)$$

where $d_{i-1/2} := (d_{i-1} + d_i)/2$. thus, replacing $2p_i^1 d_{i-1}$ by $d_{i-1/2}$ in (13d) and p_i^1 by $1/2$ elsewhere, the scheme (13) becomes

$$\frac{1}{2} \left(\frac{Y_{i-1}^j - Y_{i-1}^{j-1}}{\Delta\theta} \right) + \frac{1}{2} \left(\frac{Y_i^j - Y_i^{j-1}}{\Delta\theta} \right) + [\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} Y]_i^j = \frac{1}{2} (f_{i-1}^j + f_i^j), \quad (15a)$$

for $i = 1, 2, 3, \dots, \mathcal{N}/2$, and

$$\frac{Y_i^j - Y_i^{j-1}}{\Delta\theta} + [\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} Y]_i^j = f_i^j, \quad \text{for } i = \mathcal{N}/2 + 1, \dots, \mathcal{N} - 1, \quad (15b)$$

where

$$[\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} Y]_i^j = q_{ij}^- Y_{i-1}^j + q_{ij}^c Y_i^j + q_{ij}^+ Y_{i+1}^j, \quad (15c)$$

with

$$\begin{cases} q_{ij}^- = -\frac{\varepsilon}{h_i \hat{h}_i} - \frac{d_{i-1/2}}{h_i} + \frac{1}{2} \kappa_{2i-1}^j, \\ q_{ij}^+ = -\frac{\varepsilon}{h_{i+1} \hat{h}_i}, \\ q_{ij}^c = -q_{ij}^- - q_{ij}^+ + \frac{1}{2} \kappa_{2i-1}^j + \frac{1}{2} \kappa_{2i}^j \end{cases} \quad (15d)$$

for $i = 1, 2, 3, \dots, \mathcal{N}/2$, and

$$\begin{cases} q_{ij}^- = -\frac{\varepsilon}{h_i \hat{h}_i} - \frac{d_i}{2\hat{h}_i}, \\ q_{ij}^+ = -\frac{\varepsilon}{h_{i+1} \hat{h}_i} + \frac{d_i}{2\hat{h}_i}, \\ q_{ij}^c = -q_{ij}^- - q_{ij}^+ + \kappa_{2i}^j \end{cases} \quad (15e)$$

for $i = \mathcal{N}/2 + 1, \dots, \mathcal{N} - 1$,

According to the description above, the numerical solution of the problems (1) in section 5 will demonstrate that the schemes (13) and (15) provide outcomes that are almost equal.

4 Analysis of the uniform convergence

In this section, we aim to establish uniform convergence using a new concept involving modified graded meshes. Our examination will focus on the hybrid midpoint finite difference method (15). This choice is made due to the relatively simpler coefficients in the midpoint finite difference scheme compared to the HODIE finite difference scheme (13). It is important to note that the analysis presented here for the hybrid midpoint finite difference method can be readily extended to apply to the HODIE finite difference scheme (13). The analysis of the schemes (15) and (13) will be second order of convergence with respect to the perturbation parameter ε , and also show that the schemes (15) and (13) are identical.

Lemma 3. Assume that

$$\eta \|d\|_\infty < \frac{\mathcal{N}}{\ln(1/\varepsilon)} \quad \text{and} \quad \lambda \mathcal{N} \geq (\|\kappa_2\|_\infty + (\Delta\theta)^{-1}). \quad (16)$$

Then, the coefficient of (15) satisfies the following for every j :

- (a) $q_{ij}^+ \leq 0$, (b) $q_{ij}^- + (2\Delta\theta)^{-1} \leq 0$ and (c) $q_{ij}^c + (2\Delta\theta)^{-1} \geq 0$,
for $i = 1, 2, 3, \dots, \mathcal{N}/2$,
(d) $q_{ij}^- \leq 0$, (e) $q_{ij}^c + (\Delta\theta)^{-1} \geq 0$, and (f) $q_{ij}^+ \leq 0$ for $i = \mathcal{N}/2 + 1, \dots, \mathcal{N} - 1$.

Moreover, the tridiagonal matrix associated with computing the discrete solution at each time level θ_j is an M-matrix.

Proof. In the case of $1 \leq i \leq \mathcal{N}/2$, the proofs are provided as follows. Since $q_{ij}^+ = -\frac{\varepsilon}{h_{i+1}\hat{h}_i}$, which is less than zero for all i, j . Therefore we have $q_{ij}^+ \leq 0$, hence the result (a). In order to prove the result (b), we observe that the term $q_{ij}^- + (2\Delta\theta)^{-1}$ satisfies

$$\begin{aligned} q_{ij}^- + (2\Delta\theta)^{-1} &= -\frac{\varepsilon}{h_i\hat{h}_i} - \frac{d_{i-1/2}}{h_i} + \frac{1}{2}\kappa_{2i-1}^j + \frac{1}{2\Delta\theta} \\ &\leq -\frac{\varepsilon}{h_i\hat{h}_i} - \frac{d_{i-1/2}}{h_i} + \frac{1}{2}\kappa_{2i-1}^j + \frac{1}{2}(\|\kappa_2\| + (\Delta\theta)^{-1}) \\ &\leq -\frac{\varepsilon}{h_i\hat{h}_i} - \frac{d_{i-1/2}}{h_i} + \frac{1}{2}\kappa_{2i-1}^j + \lambda\mathcal{N}, \\ &\quad \text{using } \lambda\mathcal{N} \geq \|\kappa_2\| + (\Delta\theta)^{-1} \text{ and (16).} \end{aligned}$$

The inequality $\kappa_1(r) > \lambda > 0$ implies that $\lambda < d_i$ as well as $\lambda < d_{i-1}$. Therefore we have, $2\lambda < \kappa_{1i} + \kappa_{1i-1}$, that is $\lambda < \frac{\kappa_{1i} + \kappa_{1i-1}}{2}$. Moreover, $\kappa_{1i-1/2} \frac{\lambda}{h_i} < \frac{\kappa_{1i-1/2}}{h_i}$ results $\lambda\mathcal{N} < \frac{\lambda}{h_i} < \frac{\kappa_{1i-1/2}}{h_i}$, which leads to the inequality $\lambda\mathcal{N} < \frac{\kappa_{1i-1/2}}{h_i} < 0$, thus we have $q_{ij}^- + (2\Delta\theta)^{-1} \leq 0$, which proves the result in the case of $1 \leq i \leq \mathcal{N}/2$.

From (15d), we get

$$\begin{aligned} q_{ij}^c + (2\Delta\theta)^{-1} &= -q_{ij}^- - q_{ij}^+ + \frac{1}{2}\kappa_{2i-1}^j + \frac{1}{2}\kappa_{2i}^j \\ &= \frac{\varepsilon}{h_i\hat{h}_i} + \frac{a_{i-1/2}}{h_i} - \frac{1}{2}\kappa_{2i-1}^j + \frac{\varepsilon}{\hat{h}_i h_{i+1}} + \frac{1}{2}\kappa_{2i-1}^j + \frac{1}{2}\kappa_{2i}^j \end{aligned}$$

$$= \frac{\varepsilon}{\hat{h}_i} \left(\frac{h_{i+1} + h_i}{h_{i+1} h_i} \right) + \frac{\kappa_{1i-1/2}}{h_i} + \frac{1}{2} \kappa_{2i}^j \geq 0.$$

Hence, $q_{ij}^- + (2\Delta\theta)^{-1} \geq 0$. Thus we have the result (c).

In the case of $i = \mathcal{N}/2 + 1, \dots, \mathcal{N} - 1$, the results are provided as follows. From (15e), it follows that $q_{ij}^- \leq 0$ since $q_{ij}^- = -\frac{\varepsilon}{h_i \hat{h}_i} - \frac{d_i}{2\hat{h}_i}$. Hence we have established the result (d), that is, $q_{ij}^- \leq 0$.

Observing from (15e), we have

$$-q_{ij}^- - q_{ij}^+ + \kappa_{2i}^j + (\Delta\theta)^{-1} = \frac{\varepsilon}{h_i \hat{h}_i} + \frac{d}{2\hat{h}_i} + \frac{\varepsilon}{h_{i+1} \hat{h}_i} - \frac{d}{2\hat{h}_i} + \kappa_{2i}^j + \frac{1}{\Delta\theta} \geq 0.$$

Therefore, $q_{ij}^c + (\Delta\theta)^{-1} \geq 0$. Hence we have established the result (e).

In order to prove the result (f), we note that

$$\begin{aligned} q_{ij}^+ &= -\frac{\varepsilon}{h_{i+1} \hat{h}_i} + \frac{d_i}{2\hat{h}_i} \leq \frac{\varepsilon}{h_{i+1} \hat{h}_i} + \frac{\|d_i\|}{2\hat{h}_i} \\ &\leq \frac{\varepsilon}{h_{i+1} \hat{h}_i} + \frac{\mathcal{N}}{\ln(1/\varepsilon)} \frac{1}{2\hat{h}_i} = \frac{\varepsilon}{h_{i+1} \hat{h}_i} + \frac{\mathcal{N}\varepsilon}{(1-\varepsilon)} \frac{1}{2\hat{h}_i} \\ &= \frac{-\varepsilon}{\hat{h}_i} \left(\frac{1}{h_{i+1}} - \frac{\mathcal{N}}{2(1-\varepsilon)} \right). \end{aligned}$$

From the inequality $\frac{1}{\mathcal{N}} > h_{i+1}$ it is straight forward to observe that $\frac{\mathcal{N}}{2(1-\varepsilon)} < \frac{1}{h_{i+1}}$. Substituting this in the previous inequality we obtain $q_{ij}^+ \leq 0$, which proves the result for the case (f). This completes the proof. \square

Assuming the validity of (16), we proceed to establish that Lemma 3 implies the existence of a unique solution for the scheme (15) at each time level. Furthermore, the solution adheres to a discrete maximum principle. By incorporating the maximum principle with a barrier function expressed as $\mathcal{C}(1+r)$, a priori bound $\|Y\|_{\infty, d} \leq \mathcal{C}\|f\|_{\infty}$ with a constant \mathcal{C} is derived. Here, the discrete maximum norm is defined as $\|z\|_{\infty, d} := \max_{i,j} |z_i^j|$ for each mesh function z .

Now, we will present our main result. To obtain the estimation we will follow the Koptewa's methodology [13, 14] and also the result presented in [16].

Theorem 1. Assume (16) is valid. Then there exists a constant \mathcal{C} such that

$$\max_{i,j} |y(r_i, \theta_j) - Y_i^j| \leq \mathcal{C}[\Delta\theta + \varepsilon \mathcal{N}^{-1} + (\mathcal{N}^{-1} \ln(1/\varepsilon))^2]. \quad (17)$$

Proof. Suppose that $\zeta_i^j = y_i^j - Y_i^j$ is the error of discrete solution of the problem (1) on the modified graded mesh applied the scheme (15) at each grid point (r_i, θ_j) . We can write the scheme of midpoint, which is given in (15) as

$$[\tilde{\gamma}_\theta Y]_i^j + [\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} Y]_i^j = \hat{f}_i^j \quad \text{for } i = 1, 2, 3, \dots, \mathcal{N}-1, \text{ and } j = 1, \dots, \mathcal{M}, \quad (18)$$

where

$$\hat{f}_i^j = \begin{cases} \frac{1}{2}[f(r_{i-1}, \theta_j) + f(r_i, \theta_j)] & \text{if } i \leq \frac{\mathcal{N}}{2}, j \leq \frac{\mathcal{M}}{2}, \\ f(r_i, \theta_j) & \text{if } i > \frac{\mathcal{N}}{2}, j > \frac{\mathcal{M}}{2}, \end{cases} \quad (19)$$

and the backward difference operator $\tilde{\gamma}_\theta$ can be defined analogously. Therefore, at each point $(r_i, \theta_j) \in \Lambda$, the truncation error of the scheme is

$$[\tilde{\gamma}_\theta \zeta + \mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} \zeta]_i^j = \vartheta_{1;i}^j + \vartheta_{2;i}^j, \quad (20)$$

where

$$\vartheta_{1;i}^j := [\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} y]_i^j - (\overline{\mathcal{L}_\varepsilon y})_i^j \quad \text{and} \quad \vartheta_{2;i}^j := \tilde{\gamma}_\theta y_i^j - (\tilde{y}_\theta)_i^j \quad (21)$$

with $(\tilde{y}_\theta)_i^j$ define similarly to (19), and

$$(\overline{\mathcal{L}_\varepsilon y})_i^j = \begin{cases} \frac{1}{2}[(\mathcal{L}_\varepsilon y)(r_{i-1}, \theta_j) + (\mathcal{L}_\varepsilon y)(r_i, \theta_j)], & \text{if } i \leq \mathcal{N}/2, \\ (\mathcal{L}_\varepsilon y)(r_i, \theta_i), & \text{if } i > \mathcal{N}/2. \end{cases}$$

Decompose ζ as $\zeta = \mu + \nu$. the functions $\{\mu_i^j\}$, $j = 0, \dots, \mathcal{M}$ are the solutions to the discrete boundary value problem with two-point,

$$[\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} \mu]_i^j = \vartheta_{1;i}^j \quad \text{for } i = 1, \dots, \mathcal{N}-1, \quad \mu_0^j = \mu_{\mathcal{N}}^j = 0, \quad (22)$$

while $\{\nu_i^j\}$ are the solution of a discrete parabolic problem defined by

$$[\tilde{\gamma}_\theta \nu + \mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} \nu]_i^j = \vartheta_{2;i}^j - \tilde{\gamma}_\theta \mu_i^j \quad \text{for } i = 1, \dots, \mathcal{N} - 1, \quad (23a)$$

with the boundary conditions

$$\nu_0^j = \nu_{\mathcal{N}}^j = 0 \quad \text{for } j = 1, \dots, \mathcal{M}, \quad (23b)$$

and the initial condition

$$\nu_i^0 = -\nu_i^0 \quad \text{for } i = 0, \dots, \mathcal{N}. \quad (23c)$$

Equation (22) precisely represents the identity obtained when examining the error μ in a two-point boundary value problem that has undergone discretization using \mathcal{L}_ε , with $\vartheta_{1;i}^j$ serving as the truncation error. Utilized the bound (6) with the value of $l = 0$, we obtained the same bound on the $\vartheta_{1;i}^j$ as for a convection-diffusion two point boundary value problems. As a result, it is possible to use the error bound determined in [20],

$$|\mu_i^j| \leq \mathcal{C}[\varepsilon \mathcal{N}^{-1} + (\mathcal{N}^{-1} \ln(1/\varepsilon))^2] \quad \text{for all } i, j. \quad (24)$$

Again, we include the one-more error component ν . Lemma 1 implies that the problem (23) satisfies a discrete maximum principle just as (15) does, so

$$\begin{aligned} \|\nu\|_{\infty, d} &\leq \mathcal{C} \left(\max_i |\mu_i^0| + \|\vartheta_2 - \tilde{\gamma}_\theta \mu\|_{\infty, d} \right) \\ &\leq \mathcal{C}[\Delta\theta + \varepsilon \mathcal{N}^{-1} + (\mathcal{N}^{-1} \ln(1/\varepsilon))^2 + \|\tilde{\gamma}_\theta \mu\|_{\infty, d}], \end{aligned} \quad (25)$$

where we used (24) with $j = 0$ and also

$$|\vartheta_{2;i}^j| \leq \mathcal{C} \Delta\theta \quad \text{for } i = 1, \dots, \mathcal{N} - 1, \quad \text{and } j = 1, \dots, \mathcal{M}, \quad (26)$$

The verification has been completed using Taylor's series expansion and (6). It remains to estimate $\tilde{\gamma}_\theta \mu$ appears in (23). Utilizing the assumption that $\kappa_1 = \kappa_1(r)$ is independent of θ , a straightforward calculation reveals that, for each fixed j , the definition (22) implies satisfaction for $\tilde{\gamma}_\theta \mu$

$$[\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}}(\tilde{\gamma}_\theta \mu)]_i^j = \tilde{\gamma}_\theta \vartheta_{1;i}^j - ((\tilde{\gamma}_\theta \kappa_2) \mu^{j-1})_i \quad \text{for } i = 1, \dots, \mathcal{N} - 1, \quad (27a)$$

$$(\tilde{\gamma}_\theta \mu)_0^j = (\tilde{\gamma}_\theta \mu)_{\mathcal{N}}^j = 0, \quad (27b)$$

The notation $(\cdot)_i$ employed here carries the same meaning as in (18).

Based on the decomposition $\gamma_\theta \mu = \Phi + \Psi$, wherein, for each fixed $j \in 1, 2, \dots, \mathcal{M}$, the following relationship holds:

$$[\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} \Phi]_i^j = \tilde{\gamma}_\theta \vartheta_{1;i}^j \quad \text{for } i = 1, \dots, \mathcal{N} - 1 \quad \text{with } \Phi_0^j = \Phi_{\mathcal{N}}^j = 0, \quad (28a)$$

$$[\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} \Psi]_i^j = -((\tilde{\gamma}_\theta \kappa_2) \mu^{j-1})_i \quad \text{for } i = 1, \dots, \mathcal{N} - 1 \quad \text{with } \Psi_0^j = \Psi_{\mathcal{N}}^j = 0. \quad (28b)$$

To examine the discrete two-point boundary value problem (28a), an analysis will be conducted and observe that for $i \leq \mathcal{N}/2$ the right-hand side of (28a) is

$$\begin{aligned} \tilde{\gamma}_\theta \vartheta_{1;i}^j &= \frac{1}{2\Delta\theta} (\vartheta_{1;i-1}^j - \vartheta_{1;i-1}^{j-1}) + \frac{1}{2\Delta\theta} (\vartheta_{1;i}^j - \vartheta_{1;i}^{j-1}) \\ &= \frac{1}{2\Delta\theta} [([\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} y]_{i-1}^j - [\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} y]_{i-1}^{j-1}) - ([\mathcal{L}_\varepsilon y]_{i-1}^j - [\mathcal{L}_\varepsilon y]_{i-1}^{j-1})] \\ &\quad + \frac{1}{2\Delta\theta} [([\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} y]_i^j - [\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} y]_i^{j-1}) - ([\mathcal{L}_\varepsilon y]_i^j - [\mathcal{L}_\varepsilon y]_i^{j-1})]. \end{aligned}$$

Set $\overline{\mathcal{L}_\varepsilon} y = -\varepsilon y_{rr} + \kappa_1 y_r$. Let $[\overline{\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}}} Y]_i^j$ be defined by setting $\kappa_2 \equiv 0$ in $[\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}} y]_i^j$ for all i, j ; that is, $\overline{\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}}}$ is the discretization of $\overline{\mathcal{L}_\varepsilon}$. Then for $i \leq \mathcal{N}/2$, we can express the above formula in the form

$$\begin{aligned} \tilde{\gamma}_\theta \vartheta_{1;i}^j &= \frac{1}{2\Delta\theta} \int_{\theta_{j-1}}^{\theta_j} [(\overline{\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}}} y_\theta(r_{i-1}, \theta) + \overline{\mathcal{L}_\varepsilon^{\mathcal{N}, \mathcal{M}}} y_\theta(r_i, \theta)) \\ &\quad - (\overline{\mathcal{L}_\varepsilon} y_\theta(r_{i-1}, \theta) + \overline{\mathcal{L}_\varepsilon} y_\theta(r_i, \theta))] d\theta. \end{aligned}$$

It is important to note that here we employed the assumption that $\kappa_1 = \kappa_2(r)$ is not dependent on θ , since this yields $\overline{\mathcal{L}_\varepsilon} y_\theta = (\overline{\mathcal{L}_\varepsilon} y)_\theta$. Hence, we employed the Peano kernel theorem that is also used in the article [12], and we get

$$\begin{aligned} |\tilde{\gamma}_\theta \vartheta_{1;i}^j| &= \mathcal{C} \varepsilon \int_{r_{i-1}}^{r_{i+1}} \max_{\theta \in [\theta_{j-1}, \theta_j]} |y_{rrr\theta}(r, \theta)| d\theta \\ &\quad + \mathcal{C} h_i \int_{r_{i-1}}^{r_i} \max_{\theta \in [\theta_{j-1}, \theta_j]} (|y_{r\theta}| + |y_{rr\theta}| + |y_{rrr\theta}|)(r, \theta) d\theta. \end{aligned}$$

The bounds of (8) are unaffected by the addition of the θ -derivative, resulting in an estimate that is equivalent to the corresponding truncation error limits appearing in [20] for a typical two-point boundary value problem. When $i \geq \mathcal{N}/2$, then the bound from the last inequality in the proof of a bound that is identical to the equivalent truncation error bound derivative [20]. These results show that analysis of (28a) may be performed in a similar manner to that of (22), with the exception that one utilizes the bound (8) with $l = 1$. We therefore obtain

$$|\Phi_i^j| \leq \mathcal{C}[\varepsilon \mathcal{N}^{-1} + (\mathcal{N}^{-1} \ln(1/\varepsilon))^2] \quad \text{for all } i \text{ and } j. \quad (29)$$

To handle (27b), note that \mathcal{L}_ε is an M -matrix and therefore fulfills the discrete maximum principle. The easy conclusion that one has to satisfy for every j

$$\begin{aligned} \max_i |\Psi_i^j| &\leq C \max_i |((\tilde{\gamma}_\theta b) \mu^{j-1})_i| \\ &\leq \mathcal{C} \max_i |\mu_i^{j-1}| \leq \mathcal{C}[\varepsilon \mathcal{N}^{-1} + (\mathcal{N}^{-1} \ln(1/\varepsilon))^2], \end{aligned} \quad (30)$$

where we used $|\tilde{\gamma}_\theta \kappa_2| \leq \mathcal{C}$ and (24).

Combining (24), (25), (29), and (30), we get (17)

$$\max_{i,j} |y(r_i, \theta_j) - Y_i^j| \leq \mathcal{C}[\Delta\theta + \varepsilon \mathcal{N}^{-1} + (\mathcal{N}^{-1} \ln(1/\varepsilon))^2].$$

and Theorem 1 also holds true for the scheme (13). This completes the proof. \square

5 Examples and their numerical results

In this section, we shall present the numerical results obtained by the two finite difference schemes, the hybrid midpoint method (15) and the HODIE method (13) of the problem (1) on the modified graded mesh and also calculate the maximum point-wise error and order of convergence with the different values of ε and \mathcal{N}, \mathcal{M} . We tackled two Examples to showcase the effectiveness and efficiency of the proposed schemes. It's important to note that this

article does not provide an exact solution for Examples 1 and 2. Instead, we employ the double mesh approach outlined below to assess the maximum point-wise errors and determine the order of convergence. We demonstrate the efficiency of the proposed numerical scheme by two examples to show that the schemes (13) and (15) yield very similar results and confirm the convergence estimate of Theorem 1.

Example 1. Consider the following parabolic initial-boundary value problem:

$$\begin{cases} y_\theta - \varepsilon y_{rr} + \left(1 + r^2 + \frac{\sin(\pi r)}{2}\right) y_r + (1 + r^2 + \sin(\pi\theta)) y = f(r, \theta), \\ f(r, \theta) = r^3(1-r)^3 + \theta(1-\theta) \sin(\pi\theta), \quad \text{for } (r, \theta) \in (0, 1) \times (0, 1), \\ y(0, \theta) = y(1, \theta) = y(r, 0) = 0 \quad \text{for } (r, \theta) \in [0, 1], \end{cases} \quad (31)$$

The exact solution of the $y(r, \theta)$ of (31) is not provided and also the results of this problem satisfy only the first-order and second-order corner compatibility conditions (4) and (5). The point-wise errors $|y(r_i, \theta_i) - Y_i^j|$ are obtained on the our mesh $\Lambda_{\mathcal{M}, \mathcal{N}}$. The double mesh technique can also be found in the reference [3, 17]. That is a new approximate solution $\{\hat{Y}_i^j\}$ is computed using the same scheme but on the mesh the comprises the points of the original mesh and their midpoints $((r_{i-1} + r_i)/2, \theta_i)$, $((r_i, (\theta_{i-1} + \theta_i)/2)$ and $((r_{i-1} + r_i)/2, (\theta_{i-1} + \theta_i)/2)$. Thus, the values Y_i^j and \hat{Y}_{2i}^{2j} are computed at the same physical point (r_i, θ_i) of $\Lambda_{\mathcal{M}, \mathcal{N}}$. Then at the mesh points of the original mesh $\Lambda_{\mathcal{M}, \mathcal{N}}$ one calculates the maximum and uniform two mesh differences defined by

$$d_{\varepsilon}^{\mathcal{N}, \mathcal{M}} = \max_{0 \leq j \leq \mathcal{M}} \max_{0 \leq i \leq \mathcal{N}} |Y_i^j - \hat{Y}_{2i}^{2j}|, \quad d^{\mathcal{N}, \mathcal{M}} = \max_{\varepsilon \in S} d_{\varepsilon}^{\mathcal{N}, \mathcal{M}}, \quad (32)$$

where $S := \{2^{-3}, 2^{-6}, 2^{-9}, 2^{-12}, 2^{-15}, \dots, 2^{-30}\}$. From these values one computes the order of convergence and the uniform orders of convergence in the standard way:

$$p_{\varepsilon}^{\mathcal{N}, \mathcal{M}} := \frac{\log(d_{\varepsilon}^{\mathcal{N}, \mathcal{M}} / \log(d_{\varepsilon}^{2\mathcal{N}, 2\mathcal{M}}))}{\log 2}, \quad p_{uni}^{\mathcal{N}, \mathcal{M}} := \frac{\log(d^{\mathcal{N}, \mathcal{M}} / \log(d^{2\mathcal{N}, 2\mathcal{M}}))}{\log 2}.$$

We employ the proposed schemes (15) and (13), the hybrid midpoint finite difference scheme and the HODIE finite difference scheme on the modified

graded mesh to solve the Examples 1 and 2 for different values of perturbation parameter ε with the spatial mesh grid size \mathcal{N} and time grid size \mathcal{M} . We have also calculated the maximum point-wise errors and their corresponding order of convergent. From Tables 1 to 4, we can analyze that the proposed schemes (15) and (13) with the modified graded mesh are ε -uniformly convergent for distinct values of ε and \mathcal{N}, \mathcal{M} . Because of this as a result of this observation, we can assert that the computationally achieved order of convergence surpasses the one predicted in the preceding section. It has been demonstrated that the theoretical rate of convergence for the developed method is second order in the spatial direction and first order in the time direction. Besides, the comparison of numerical results obtained by the proposed scheme and results in [3] and [17] are tabulated in Tables 5 and 6 for Example 1. From these tables, one can conclude that the proposed scheme gives better results than the scheme considered in [3] and [17].

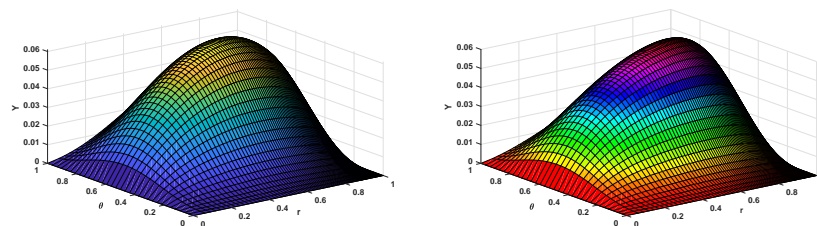
Figure 2 shows the numerical solution profile for Example 1 for various values of ε and step sizes \mathcal{N} and \mathcal{M} for schemes (13) and (15), respectively. The calculated maximum point-wise errors $d^{\mathcal{N}, \mathcal{M}}$ and the corresponding order of convergence $p_{uni}^{\mathcal{N}, \mathcal{M}}$ for Example 1 with schemes (13) and (15) on modified graded mesh are shown in Tables 1 and 2, respectively. From these results one can observe the ε -uniform second-order convergence of the numerical solution.

Example 2. Consider the following parabolic initial-boundary value problem:

$$\begin{cases} y_\theta - \varepsilon y_{rr} + \left(1 + r^2 + \frac{\sin(\pi r)}{2}\right) y_r + \left(1 + r^2 + \frac{1}{2} \sin(\pi\theta/2)\right) y = f(r, \theta) \\ f(r, \theta) = r^3(1-r)^3\theta(1-\theta)\sin(\pi\theta), \quad \text{for } (r, \theta) \in (0, 1) \times (0, 1) \\ y(0, \theta) = y(1, \theta) = y(r, 0) = 0, \quad \text{for } (r, \theta) \in [0, 1], \end{cases} \quad (33)$$

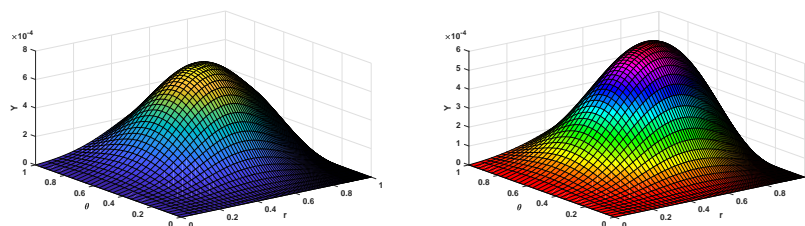
for which the exact solution is again unknown. Similarly, numerical solution profiles for Example 2 for various values of ε and step sizes \mathcal{N} and \mathcal{M} are provided in Figure 3 for the schemes (13) and (15). The results reveal the presence of a boundary layer on the right side of the domain. The calculated maximum point-wise errors $d^{\mathcal{N}, \mathcal{M}}$ and the corresponding order of convergence $p_{uni}^{\mathcal{N}, \mathcal{M}}$ for Example 2 with schemes (13) and (15) on modified

graded mesh are shown in Tables 3 and 4, respectively. From these results one can observe the ε -uniform second-order convergence of the numerical solution. The maximum point-wise errors are plotted in log-log scale in Figure 4, for the solution. From these figures, one can easily observe the second-order ε -uniform convergence.



(a) $\mathcal{N} = 128, \mathcal{M} = 32, \varepsilon = 10^{-3}$, Scheme (13) (b) $\mathcal{N} = 128, \mathcal{M} = 32, \varepsilon = 10^{-3}$, Scheme (15)

Figure 2: Solution profile for Example 1 using schemes (13) and (15) on modified graded mesh



(a) $\mathcal{N} = 128, \mathcal{M} = 32, \varepsilon = 10^{-3}$, Scheme (13) (b) $\mathcal{N} = 128, \mathcal{M} = 32, \varepsilon = 10^{-3}$, Scheme (15)

Figure 3: Solution profile for Example 2 using schemes (13) and (15) on modified graded mesh.

Table 1: *Maximum point-wise errors and the corresponding order of convergence for Example 1 on a modified graded mesh using scheme (13)*

| ϵ | Number of Intervals \mathcal{N}, \mathcal{M} | | | | |
|--------------------------------------|--|---|---|--|--|
| | $\mathcal{N} = 84$ $\mathcal{M} = 5$ | $\mathcal{N} = 168$ $\mathcal{M} = 20$ | $\mathcal{N} = 336$ $\mathcal{M} = 80$ | $\mathcal{N} = 672$ $\mathcal{M} = 320$ | $\mathcal{N} = 1344$ $\mathcal{M} = 1280$ |
| 2^{-3} | $9.4000e-03$ 1.8260 | $2.7000e-03$ 1.9338 | $6.9703e-04$ 1.9544 | $1.7986e-04$ 1.9270 | $4.7299e-05$ |
| 2^{-6} | $1.1000e-03$ 1.6292 | $3.6000e-03$ 1.8075 | $1.0000e-03$ 1.8428 | $2.8406e-04$ 1.7847 | $8.2442e-05$ |
| 2^{-9} | $1.1100e-03$ 1.5443 | $3.8000e-03$ 1.6842 | $1.2000e-03$ 1.7058 | $3.6372e-04$ 1.6212 | $1.1823e-04$ |
| 2^{-12} | $1.0700e-03$ 1.5221 | $3.7000e-03$ 1.6259 | $1.2000e-03$ 1.6130 | $3.9593e-04$ 1.5205 | $1.3800e-04$ |
| 2^{-15} | $1.0400e-03$ 1.5315 | $3.6000e-03$ 1.4188 | $1.3000e-03$ 1.5127 | $4.7031e-04$ 1.6206 | $1.5294e-04$ |
| 2^{-18} | $1.0500e-03$ 1.4580 | $3.8000e-03$ 1.1733 | $1.7000e-03$ 1.4249 | $6.3183e-04$ 1.6947 | $1.9519e-04$ |
| 2^{-21} | $1.2500e-03$ 1.3934 | $4.7000e-03$ 1.2186 | $2.0000e-03$ 1.3449 | $8.0241e-04$ 1.6453 | $2.5652e-04$ |
| 2^{-24} | $1.4600e-03$ 1.3731 | $5.6000e-03$ 1.2492 | $2.4000e-03$ 1.2745 | $9.7847e-04$ 1.5968 | $3.2349e-04$ |
| 2^{-27} | $1.6600e-03$ 1.3648 | $6.4000e-03$ 1.2618 | $2.7000e-03$ 1.2152 | $1.2000e-03$ 1.5495 | $3.9531e-04$ |
| 2^{-30} | $1.8400e-03$ 1.3617 | $7.2000e-03$ 1.2538 | $3.0000e-03$ 1.1680 | $1.3000e-03$ 1.5035 | $4.7123e-04$ |
| $d^{\mathcal{N}, \mathcal{M}}$ | $1.8400e-03$ | $7.2000e-03$ | $3.0000e-03$ | $1.3000e-03$ | $4.7123e-04$ |
| $p_{uni}^{\mathcal{N}, \mathcal{M}}$ | 1.3617 | 1.2538 | 1.1680 | 1.5035 | |

Table 2: *Maximum point-wise errors and the corresponding order of convergence for Example 1 on a modified graded mesh using scheme (15)*

| ϵ | Number of Intervals \mathcal{N}, \mathcal{M} | | | | |
|--------------------------------------|--|---|---|--|--|
| | $\mathcal{N} = 84$ $\mathcal{M} = 5$ | $\mathcal{N} = 168$ $\mathcal{M} = 20$ | $\mathcal{N} = 336$ $\mathcal{M} = 80$ | $\mathcal{N} = 672$ $\mathcal{M} = 320$ | $\mathcal{N} = 1344$ $\mathcal{M} = 1280$ |
| 2^{-3} | $9.4000e-03$ 1.8112 | $2.7000e-03$ 1.9293 | $7.0292e-04$ 1.9422 | $1.8291e-04$ 1.8988 | $4.9051e-05$ |
| 2^{-6} | $1.1100e-03$ 1.6599 | $3.5000e-03$ 1.8145 | $9.9420e-04$ 1.8411 | $2.7750e-04$ 1.7513 | $8.2428e-05$ |
| 2^{-9} | $1.1300e-03$ 1.6180 | $3.7000e-03$ 1.7392 | $1.1000e-03$ 1.7442 | $3.2987e-04$ 1.6266 | $1.0683e-04$ |
| 2^{-12} | $1.1300e-03$ 1.6181 | $3.7000e-03$ 1.6834 | $1.1000e-03$ 1.6735 | $3.5990e-04$ 1.5385 | $1.2389e-04$ |
| 2^{-15} | $1.1300e-03$ 1.6119 | $4.3000e-03$ 1.4536 | $1.6000e-03$ 1.6975 | $4.7883e-04$ 1.7850 | $1.3894e-04$ |
| 2^{-18} | $1.1900e-03$ 1.7785 | $5.2000e-03$ 1.6518 | $2.1000e-03$ 1.6354 | $6.6140e-04$ 1.8175 | $1.8765e-04$ |
| 2^{-21} | $1.3100e-03$ 1.8940 | $6.1000e-03$ 1.6578 | $2.6000e-03$ 1.5750 | $8.6310e-04$ 1.7861 | $2.5027e-04$ |
| 2^{-24} | $1.4200e-03$ 1.6196 | $7.0000e-03$ 1.7805 | $3.1000e-03$ 1.5176 | $1.1000e-03$ 1.7547 | $3.2024e-04$ |
| 2^{-27} | $1.5200e-03$ 1.9542 | $7.8000e-03$ 1.6154 | $3.6000e-03$ 1.7623 | $1.3000e-03$ 1.7233 | $3.9701e-04$ |
| 2^{-30} | $1.6000e-03$ 1.9022 | $8.6000e-03$ 1.5571 | $4.1000e-03$ 1.6089 | $1.6000e-03$ 1.6922 | $4.8006e-04$ |
| $d^{\mathcal{N}, \mathcal{M}}$ | $1.6000e-03$ | $8.6000e-03$ | $4.1000e-03$ | $1.6000e-03$ | $4.8006e-04$ |
| $p_{uni}^{\mathcal{N}, \mathcal{M}}$ | 1.9022 | 1.5571 | 1.6089 | 1.6922 | |

Table 3: *Maximum point-wise errors and the corresponding order of convergence for Example 2 on a modified graded mesh using scheme (13)*

| ϵ | Number of Intervals \mathcal{N}, \mathcal{M} | | | | |
|--------------------------------------|--|---|---|--|--|
| | $\mathcal{N} = 84$ $\mathcal{M} = 5$ | $\mathcal{N} = 168$ $\mathcal{M} = 20$ | $\mathcal{N} = 336$ $\mathcal{M} = 80$ | $\mathcal{N} = 672$ $\mathcal{M} = 320$ | $\mathcal{N} = 1344$ $\mathcal{M} = 1280$ |
| 2^{-3} | $1.0565e-04$ | $3.0122e-05$ | $8.3177e-06$ | $2.3937e-06$ | $7.6229e-07$ |
| | 1.8104 | 1.8565 | 1.7970 | 1.6508 | |
| 2^{-6} | $1.5408e-04$ | $5.3824e-05$ | $1.6708e-05$ | $5.2492e-06$ | $1.8022e-06$ |
| | 1.5173 | 1.6877 | 1.6704 | 1.5424 | |
| 2^{-9} | $1.6659e-04$ | $6.3801e-05$ | $2.1436e-05$ | $7.2210e-06$ | $2.6525e-06$ |
| | 1.3846 | 1.5735 | 1.5698 | 1.4448 | |
| 2^{-12} | $1.6920e-04$ | $6.6955e-05$ | $2.3753e-05$ | $8.4545e-06$ | $3.2711e-06$ |
| | 1.3375 | 1.4951 | 1.4903 | 1.3699 | |
| 2^{-15} | $1.7144e-04$ | $6.8827e-05$ | $2.5610e-05$ | $9.5467e-06$ | $3.8417e-06$ |
| | 1.3166 | 1.4263 | 1.4236 | 1.3133 | |
| 2^{-18} | $1.8052e-04$ | $7.0612e-05$ | $2.7351e-05$ | $1.0617e-05$ | $4.4059e-06$ |
| | 1.3542 | 1.3683 | 1.3651 | 1.2689 | |
| 2^{-21} | $2.0035e-04$ | $7.5085e-05$ | $2.9004e-05$ | $1.1683e-05$ | $4.9695e-06$ |
| | 1.4160 | 1.3723 | 1.3118 | 1.2332 | |
| 2^{-24} | $2.3038e-04$ | $9.8989e-05$ | $3.0620e-05$ | $1.2744e-05$ | $5.5335e-06$ |
| | 1.2187 | 1.6928 | 1.2646 | 1.2036 | |
| 2^{-27} | $2.7570e-04$ | $1.2149e-04$ | $3.2179e-05$ | $1.3797e-05$ | $6.0980e-06$ |
| | 1.1822 | 1.9167 | 1.2217 | 1.1780 | |
| 2^{-30} | $3.1311e-04$ | $1.4101e-04$ | $3.7471e-05$ | $1.4841e-05$ | $6.6619e-06$ |
| | 1.1509 | 1.9119 | 1.3362 | 1.1556 | |
| $d^{\mathcal{N}, \mathcal{M}}$ | $3.1311e-04$ | $1.4101e-04$ | $3.7471e-05$ | $1.4841e-05$ | $6.6619e-06$ |
| $p_{uni}^{\mathcal{N}, \mathcal{M}}$ | 1.1509 | 1.9119 | 1.3362 | 1.1556 | |

Table 4: *Maximum point-wise errors and the corresponding order of convergence for Example 2 on a modified graded mesh using scheme (15)*

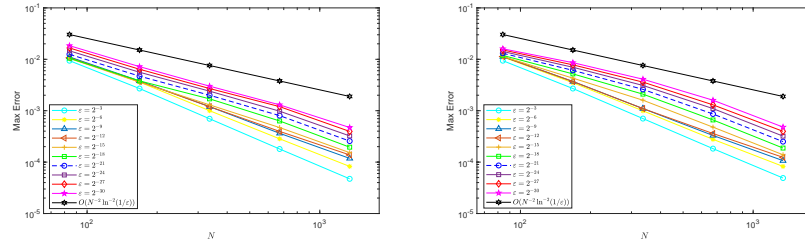
| ϵ | Number of Intervals \mathcal{N}, \mathcal{M} | | | | |
|--------------------------------------|--|---|---|--|--|
| | $\mathcal{N} = 84$ $\mathcal{M} = 5$ | $\mathcal{N} = 168$ $\mathcal{M} = 20$ | $\mathcal{N} = 336$ $\mathcal{M} = 80$ | $\mathcal{N} = 672$ $\mathcal{M} = 320$ | $\mathcal{N} = 1344$ $\mathcal{M} = 1280$ |
| 2^{-3} | $1.0178e-04$ | $2.9341e-05$ | $8.0678e-06$ | $2.2905e-06$ | $7.0284e-07$ |
| | 1.7945 | 1.8626 | 1.8165 | 1.7044 | |
| 2^{-6} | $1.2021e-04$ | $3.9857e-05$ | $1.1346e-05$ | $3.1890e-06$ | $9.6295e-07$ |
| | 1.5926 | 1.8127 | 1.8310 | 1.7276 | |
| 2^{-9} | $1.1696e-04$ | $4.1225e-05$ | $1.2193e-05$ | $3.5503e-06$ | $1.1220e-06$ |
| | 1.5044 | 1.7575 | 1.7801 | 1.6619 | |
| 2^{-12} | $1.1120e-04$ | $4.0477e-05$ | $1.2584e-05$ | $3.8367e-06$ | $1.2732e-06$ |
| | 1.4580 | 1.6856 | 1.7136 | 1.5913 | |
| 2^{-15} | $1.1026e-04$ | $3.9240e-05$ | $1.2814e-05$ | $4.1096e-06$ | $1.4249e-06$ |
| | 1.4906 | 1.6146 | 1.6407 | 1.5281 | |
| 2^{-18} | $1.1865e-04$ | $5.6710e-05$ | $1.2882e-05$ | $4.3681e-06$ | $1.5760e-06$ |
| | 1.7650 | 2.1382 | 1.5603 | 1.7708 | |
| 2^{-21} | $1.4002e-04$ | $7.2639e-05$ | $1.7165e-05$ | $4.6091e-06$ | $1.7255e-06$ |
| | 1.9468 | 2.0813 | 1.8969 | 1.4175 | |
| 2^{-24} | $1.7524e-04$ | $8.5364e-05$ | $2.4497e-05$ | $4.8268e-06$ | $1.8733e-06$ |
| | 1.8376 | 1.8010 | 2.3435 | 1.3655 | |
| 2^{-27} | $2.2202e-04$ | $9.6868e-05$ | $3.2831e-05$ | $5.5598e-06$ | $2.0192e-06$ |
| | 1.6966 | 1.5610 | 2.0620 | 1.7612 | |
| 2^{-30} | $2.7531e-04$ | $1.0342e-04$ | $4.1820e-05$ | $7.3340e-06$ | $2.1624e-06$ |
| | 1.8125 | 1.9063 | 2.0115 | 1.7619 | |
| $d^{\mathcal{N}, \mathcal{M}}$ | $2.7531e-04$ | $1.0342e-04$ | $4.1820e-05$ | $7.3340e-06$ | $2.1624e-06$ |
| $p_{uni}^{\mathcal{N}, \mathcal{M}}$ | 1.8125 | 1.9063 | 2.0115 | 1.7619 | |

Table 5: Comparison of maximum point-wise errors and the corresponding order of convergence for Example 1 on a modified graded mesh using scheme (13)

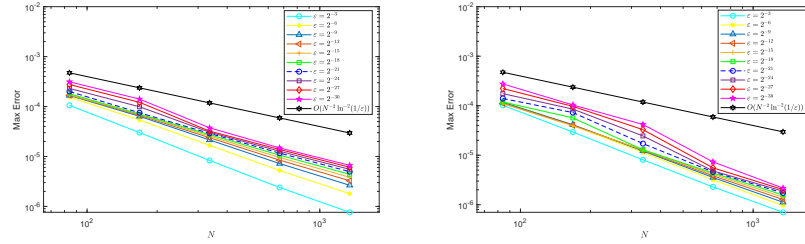
| ε | HODIE scheme on modified graded mesh | | | | |
|---------------|--------------------------------------|-----------------------|------------------------|------------------------|------------------------|
| | $\mathcal{N} = 32$ | $\mathcal{N} = 64$ | $\mathcal{N} = 128$ | $\mathcal{N} = 256$ | $\mathcal{N} = 512$ |
| | $\Delta t = 0.025$ | $\Delta t = 0.025/4$ | $\Delta t = 0.025/4^2$ | $\Delta t = 0.025/4^3$ | $\Delta t = 0.025/4^4$ |
| 2^{-6} | $8.999e-3$ 1.6443 | $3.8000e-3$ 1.6842 | $1.2000e-3$ 1.7058 | $3.6372e-4$ 1.6212 | $1.1823e-4$ |
| 2^{-8} | $9.322e-3$ 1.6221 | $3.7000e-3$ 1.6259 | $1.2000e-3$ 1.6130 | $3.9593e-4$ 1.5205 | $1.3800e-4$ |
| 2^{-10} | $1.0400e-3$ 1.5315 | $3.6000e-3$ 1.4188 | $1.3000e-3$ 1.5127 | $4.7031e-4$ 1.6206 | $1.5294e-4$ |
| Result in [3] | On the Shishkin mesh | | | | |
| 2^{-6} | $8.998e-3$ 1.630 | $2.906e-3$ 1.492 | $1.033e-3$ 1.647 | $3.298e-4$ 1.677 | $1.032e-4$ |
| 2^{-8} | $9.322e-3$ 1.631 | $3.009e-3$ 1.616 | $9.817e-4$ 1.650 | $3.128e-4$ 1.563 | $1.059e-4$ |
| 2^{-10} | $9.411e-3$ 1.631 | $3.038e-3$ 1.609 | $9.961e-4$ 1.638 | $3.201e-4$ 1.669 | $1.007e-4$ |

Table 6: Comparison of maximum point-wise errors and the corresponding order of convergence for Example 1 on a modified graded mesh using scheme (15)

| ε | Midpoint Scheme on Modified graded mesh | | | | |
|----------------|---|-----------------------|------------------------|------------------------|------------------------|
| | $\mathcal{N} = 32$ | $\mathcal{N} = 64$ | $\mathcal{N} = 128$ | $\mathcal{N} = 256$ | $\mathcal{N} = 512$ |
| | $\Delta t = 0.025$ | $\Delta t = 0.025/4$ | $\Delta t = 0.025/4^2$ | $\Delta t = 0.025/4^3$ | $\Delta t = 0.025/4^4$ |
| 10^{-1} | $9.4000e-3$ 1.8112 | $2.7000e-3$ 1.9293 | $7.0292e-4$ 1.9422 | $1.8291e-4$ 1.8988 | $4.9051e-5$ |
| 10^{-2} | $1.1100e-3$ 1.6599 | $3.5000e-3$ 1.8145 | $9.9420e-4$ 1.8411 | $2.7750e-4$ 1.7513 | $8.2428e-5$ |
| 10^{-3} | $1.1300e-3$ | $3.7000e-3$ | $1.1000e-3$ | $3.2987e-4$ | $1.0683e-4$ |
| Result in [17] | On the Shishkin mesh | | | | |
| 10^{-1} | $2.3969e-3$ 1.4720 | $8.6402e-4$ 1.2873 | $3.5400e-4$ 1.1589 | $1.5854e-4$ 1.0831 | $7.4832e-5$ |
| 10^{-2} | $1.2246e-2$ 1.4631 | $4.4419e-3$ 1.4509 | $1.6249e-3$ 1.4146 | $6.0951e-4$ 1.3592 | $2.3759e-4$ |
| 10^{-3} | $1.1994e-2$ 1.4561 | $4.3716e-3$ 1.4438 | $1.6070e-3$ 1.4084 | $6.0543e-4$ 1.3477 | $2.3789e-4$ |



(a) Log-log plot of Example 1 using scheme (13) (b) Log-log plot of Example 1 using scheme (15)



(c) Log-log plot of Example 2 using scheme (13) (d) Log-log plot of Example 2 using scheme (15)

Figure 4: Log-log plot of Examples 1 and 2

6 Discussion and conclusions

In this article, for the first time, we propose a modified graded mesh for convection-diffusion problems that provides second-order uniform convergence with respect to the perturbation parameter. We have presented effective numerical approaches in this work that are based on a modified graded mesh. In this two schemes are discussed namely hybrid finite difference schemes (15) and HODIE finite difference schemes (13), on a modified graded mesh. Both the above schemes show identical convergence, which can be viewed from the theoretical and numerical results established in this work.

In order to verify the theoretical estimation established, we conduct numerical experiments for two test problems for various values of ε and step sizes N and M . In order to find maximum point-wise error and corresponding order of convergence, we double the number of mesh points in the spatial direction and quadruple the number of mesh points in the time direction and apply the schemes (15) and (13) on the modified graded mesh. Through this procedure, we get the second-order convergence. These can be observed from

the results presented in Tables 1–2 for Example 1 and Tables 3–4 for Example 2. From the above tables, it can be confirmed that overall second-order uniform convergence. Corresponding log-log plots are provided for Examples 1 and 2. Figure 4 shows the overall second-order of convergence for various values of ε for Examples 1) and 2 with the schemes (15) and (13) on modified graded mesh.

It has been shown theoretically that the proposed methods, namely the hybrid finite difference scheme and the HODIE finite difference scheme, are uniformly convergent with first-order accuracy in time and almost second-order accuracy in space. We have also provided numerical results in order to verify the theoretical conclusions. The uniform convergence of the proposed methods is shown by the numerical results obtained for two test problems. Though the proposed method provides second-order convergence in space, the overall convergence rate of the method is not improved due to the backward-Euler approach used for the temporal direction. The ability to build higher-order, more time-accurate numerical schemes using the current setting is a feasible extension that may be used to improve accuracy while reducing computing costs.

Data Availability: Enquiries about data availability should be directed to the authors

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

Acknowledgements

The authors are grateful to the anonymous reviewers and editor for their valuable suggestions, which helped significantly improve the manuscript.

References

- [1] Cai, X. and Liu, F. *A Reynolds uniform scheme for singularly perturbed parabolic differential equation*, ANZIAM Journal, 47 (2005) C633–C648.
- [2] Chi-kuang, W. *The finite element method of singular perturbation prob-*


- lem*, Appl. Math. Mech. 5(1) (1984) 1011–1018.
- [3] Clavero, C., Gracia, J.L. and Jorge, J.C. *High-order numerical methods for one-dimensional parabolic singularly perturbed problems with regular layers*, Numer. Methods Partial Differ. Equ. 21(1) (2005) 149–169.
 - [4] Clavero, C., Jorge, J.C. and Lisbona, F. *Uniformly convergent schemes for singular perturbation problems combining alternating directions and exponential fitting techniques*, Adv. Comput. Methods Bound. Inter. Layers. (1993) 33–52.
 - [5] Clavero, C., Gracia, J.L. and Lisbona, F. *High order methods on Shishkin meshes for singular perturbation problems of convection–diffusion type*, Numer. Algorithms, 22(1) (1999) 73–97.
 - [6] Clavero, C., Gracia, J.L. and Stynes, M. *A simpler analysis of a hybrid numerical method for time-dependent convection–diffusion problems*, J. Comput. Appl. Math. 235(17) (2011) 5240–5248.
 - [7] Friedman, A. *Partial differential equations of parabolic type*, Courier Dover Publications, 2008.
 - [8] Izadi, M. and Yuzbasi, S. *A hybrid approximation scheme for 1-d singularly perturbed parabolic convection-diffusion problems*, Math. Commun. 27(1) (2022) 47–62.
 - [9] Kadalbajoo, M.K. and Yadaw, A.S. *Parameter-uniform finite element method for two-parameter singularly perturbed parabolic reaction-diffusion problems*, Int. J. Comput. Methods 9(04) (2012) 1250047.
 - [10] Kaushik, A., Kumar, V., Sharma, M. and Sharma, N. *A modified graded mesh and higher order finite element method for singularly perturbed reaction–diffusion problems*, Math. Comput. Simul. 185 (2021) 486–496.
 - [11] Kaushik, A., Kumar, V., Sharma, M. and Vashishth, A.K. *A higher order finite element method with modified graded mesh for singularly perturbed two-parameter problems*, Math. Methods Appl. Sci. 43(15) (2020) 8644–8656.

- [12] Kellogg, R.B. and Tsan, A. *Analysis of some difference approximations for a singular perturbation problem without turning points*, Math. Comput. 32(144) (1978) 1025–1039.
- [13] Kopteva, N., *Uniform pointwise convergence of difference schemes for convection-diffusion problems on layer-adapted meshes*, Comput. 66(2) (2001) 179–197.
- [14] Kopteva, N.V., *On the convergence, uniform with respect to a small parameter, of a scheme with weights for a one-dimensional nonstationary convection-diffusion equation*, Zh. Vychisl. Mat. Mat. Fiz, 37 (1997) 1213–1220.
- [15] Kumar, S. and Vigo-Aguiar, J. *A parameter-uniform grid equidistribution method for singularly perturbed degenerate parabolic convection–diffusion problems*, J. Comput. Appl. Math. 404 (2022) 113273.
- [16] Linss, T. *Layer-adapted meshes and fem for time-dependent singularly perturbed reaction-diffusion problems*, Int. J. Comput. Sci. Math. 1(2-4) (2007) 259–270.
- [17] K. Mukherjee and S. Natesan. *Parameter-uniform hybrid numerical scheme for time-dependent convection-dominated initial-boundary-value problems*, Comput., 84(3) (2009) 209–230.
- [18] Mukherjee, K. and Natesan, S. *Richardson extrapolation technique for singularly perturbed parabolic convection–diffusion problems*, Comput., 92 (2011) 1–32.
- [19] Roos, H.G., Stynes, M. and Tobiska, L. *Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems*, Springer Science & Business Media, volume 24, 2008.
- [20] Stynes, M. and Roos, H.G., *The midpoint upwind scheme*, Appl. Numer. Math. 23(3) (1997) 361–374.
- [21] Sun, G. and Stynes, M. *Finite-element methods for singularly perturbed high-order elliptic two-point boundary value problems. I: reaction-diffusion-type problems*, IMA J. Numer. Anal. 15(1) (1995) 117–139.

- [22] Tian, S., Liu, X. and An, R., *A higher-order finite difference scheme for singularly perturbed parabolic problem*, Math. Probl. Eng. 2021 (2021) 1–11.
- [23] Vulanovic, R. *Higher-order monotone schemes for a nonlinear singular perturbation problem*, ZAMM Z. fur Angew. Math. 68(5), (1988) T428–T430.
- [24] Vulanović, R. and Nhan, T.A. *Robust hybrid schemes of higher order for singularly perturbed convection-diffusion problems*, Appl. Math. Comput. 386 (2020) 125495.
- [25] Ng-Stynes, M.J., O’Riordan, E. and Stynes, M. *Numerical methods for time-dependent convection-diffusion equations*, J. Comput. Appl. Math. 21(3) (1988) 289–310.



A new exact solution method for bi-level linear fractional problems with multi-valued optimal reaction maps

F.Y. Feleke and S.M. Kassa*, 

Abstract

In many practical applications, some problems are being modeled as bi-level programming problems where the upper and lower level objectives are linear fractional functions with polyhedral constraints. If the rational reaction set of (or the set of optimal solutions to) the lower level is not a singleton, then it is known that an optimal solution to the linear fractional bi-level programming problem may not occur at a boundary feasible extreme point. Hence, existing methods cannot solve such problems in general. In this article, a novel method is introduced to find the set of all feasible leader's variables that can induce multi-valued reaction map from

*Corresponding author

Received 19 May 2025; revised 7 August 2025; accepted 14 August 2025

Flagot Yohannes Feleke

Department of Mathematics, Addis Ababa University, P.O.Box 1176, Addis Ababa, Ethiopia. e-mail: yzzdfe@gmail.com

Semu Mitiku Kassa

Department of Mathematics and Statistical Sciences, Botswana International University of Science and Technology (BIUST), P/Bag 016, Palapye, Botswana. e-mail: kassas@biust.ac.bw.

How to cite this article

Feleke, F.Y. and Kassa, S.M., A new exact solution method for bi-level linear fractional problems with multi-valued optimal reaction maps. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1392-1419. <https://doi.org/10.22067/ijnao.2025.93619.1651>

the follower. The proposed algorithm combines the k th best procedure with a branch-and-bound method to find an exact global optimal solution for continuous optimistic bi-level linear fractional problems without assuming the lower level rational reaction map is single valued. The branching constraint is constructed depending on the coefficients of the objective function of the lower-level problem. The algorithm is shown to converge to the exact solution of the bi-level problem. The effectiveness of the algorithm is also demonstrated using some numerical examples.

AMS subject classifications (2020): Primary 90C32, 91A65; Secondary 90C26, 90C57, 65K10.

Keywords: Bi-level programming problem; Bi-level linear fractional programming problem; Multi-valued rational reaction map; k th best method; Branch-and-bound scheme.

1 Introduction

A bi-level problem is a constrained optimization problem where two optimization levels are involved, and one is considered as a constraint for the other. It models decentralized planning problems with two decision agents in two levels of hierarchy. Each decision maker is assumed to control a different set of variables, and the decisions are made sequentially according to a predefined order. The decision makers at the upper and lower levels are called, respectively, leader and follower. The leader and the follower each try to optimize their own objective functions, but the decision at one level affects the objective values and/or the choice of strategies of the other level.

Generally, a bi-level programming problem can be formulated as

$$\begin{aligned} \max_x \quad & F(x, y), \\ \max_y \quad & f(x, y), \\ \text{s.t.} \quad & (x, y) \in \Omega, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^m$ is the variable vector controlled by the upper level decision maker, $y \in \mathbb{R}^n$ is the variable vector controlled by the lower level decision maker, $F, f : \mathbb{R}^m \times \mathbb{R}^n \longrightarrow \mathbb{R}$ are the objective functions of the leader and follower, respectively, and $\Omega \subseteq \mathbb{R}^m \times \mathbb{R}^n$ defines the common constraint region.

Let $\Omega_1 = \{x \in \mathbb{R}^m : \exists y \text{ such that } (x, y) \in \Omega\}$ be a projection of Ω onto the *Leader's decision space*. For a fixed choice $x \in \Omega_1$ of the leader, the follower is expected to react rationally by solving

$$\begin{aligned} \max_y \quad & f(x, y), \\ \text{s.t.} \quad & y \in \Omega(x), \end{aligned} \tag{2}$$

where $\Omega(x) = \{y \in \mathbb{R}^n : (x, y) \in \Omega\}$ is the follower's feasible set for a given x , assuming that this problem has a solution. The set of optimal solution of (2) denoted by $R(x)$ is usually termed as the *rational reaction set* for the bi-level problem (1). For any decision (choice x) taken by the leader, we assume that the follower has some room to respond, that is, $R(x) \neq \emptyset$. The *inducible region*, which represents the set over which the leader may optimize his/her objective or the feasible region of the upper level decision maker, is given by $\mathcal{R} = \{(x, y) \in \Omega : y \in R(x)\}$.

Thus, in terms of the inducible region, the bi-level problem can be equivalently [8] written as

$$\begin{aligned} \max_{x, y} \quad & F(x, y) \\ \text{s.t.} \quad & (x, y) \in \mathcal{R}. \end{aligned}$$

To assure the existence of the solution of bi-level problem, we may assume that the constraint set Ω is compact, and the inducible region \mathcal{R} is nonempty. When the rational reaction map $R(x)$ is not single-valued, difficulties may arise in finding a meaningful solution to the bi-level problem, and hence the problem become *not well-posed*. Various approaches have been proposed in literature to avoid this difficulty and to insure the well-posedness of the bi-level problem (see [8] and the references therein). Among the possible assumptions, the *optimistic* approach, where the leader assumes that the

follower chooses a value that suits the choice of the leader, is more popular in application.

A linear fractional bi-level programming problem, which is a subclass of bi-level nonlinear problems, where the objective functions in both levels are linear fractional and the common constraint region is a polyhedron, can be given by the form:

$$\begin{aligned} \max_x \quad & F(x, y) = \frac{c_{11}^T x + c_{12}^T y + \alpha_{11}}{c_{21}^T x + c_{22}^T y + \alpha_{12}}, \\ \max_y \quad & f(x, y) = \frac{d_{11}^T x + d_{12}^T y + \alpha_{21}}{d_{21}^T x + d_{22}^T y + \alpha_{22}}, \\ \text{s.t.} \quad & A_1 x + A_2 y \leq b, \\ & x, y \geq 0, \end{aligned} \quad (3)$$

where for $i, j \in \{1, 2\}$, α_{ij} are scalars, c_{ij}, d_{ij}, b are vectors, A_i 's are matrices with appropriate dimensions, and with a common constraint region given by

$$\Omega = \{(x, y) : A_1 x + A_2 y \leq b, x, y \geq 0\}.$$

Linear fractional bi-level programming problems appear in various areas of application, for instance in problems that optimize some efficiency measure of a system [5].

Given a feasible choice $x \in \Omega_1$ of the leader, the solution of the lower level problem:

$$\begin{aligned} \max_y \quad & f(x, y) = \frac{d_{11}^T x + d_{12}^T y + \alpha_{21}}{d_{21}^T x + d_{22}^T y + \alpha_{22}}, \\ \text{s.t.} \quad & y \in \Omega(x), \end{aligned} \quad (4)$$

where $\Omega(x) = \{y : A_2 y \leq b - A_1 x, y \geq 0\}$, is the rational reaction set $R(x)$.

Since linear fractional problems are quasi-monotonic [5], their solutions are known to appear on a vertex of the inducible region. In terms of the inducible region, problem (3) can be equivalently written as

$$\begin{aligned} \max_{x, y} \quad & F(x, y) = \frac{c_{11}^T x + c_{12}^T y + \alpha_{11}}{c_{21}^T x + c_{22}^T y + \alpha_{12}}, \\ \text{s.t.} \quad & (x, y) \in \mathcal{R}, \end{aligned}$$

and the relaxation for the upper level problem can be given by

$$\begin{aligned} \max_{x,y} \quad & F(x,y) = \frac{c_{11}^T x + c_{12}^T y + \alpha_{11}}{c_{21}^T x + c_{22}^T y + \alpha_{12}}, \\ \text{s.t.} \quad & (x,y) \in \Omega. \end{aligned} \quad (5)$$

Related Works – A theoretical framework for solving problem (3) was developed in [5] and is used to justify the use of the k th best algorithm to solve linear fractional bi-level problems when $R(x)$ is single-valued for each feasible x . This algorithm produces exact solution for a linear fractional bi-level programming problem. An enumerative method is further tuned in [7] by applying an upper bound filter scheme. Earlier studies [4] used parametric approach (which was introduced by [12]) to solve bi-level linear fractional programming problems.

A weighting method together with the analytic hierarchy process is used to convert the bi-level problem into a single level problem in [11] to solve a bi-level linear fractional programming problem, while Toksari [15] proposed the Taylor series approach to transform the bi-level linear fractional programming problem into equivalent linear objective functions by using first order approximation. A duality gap approach is used in [16] to transform the bi-level problem into an equivalent single-level one and used an enumerative scheme to search vertices that produce the best duality gap.

Vertex search methods, like the k th best solution approach in [5], upper bound filter scheme in [7], and the enumerative scheme used in [16], search over the vertex of the constraint region Ω , with the assumption that the set $R(x)$ is single-valued for any feasible x . In the case when $R(x)$ is a single-valued map, the set of vertices of the inducible region of the problem is shown in [5] to be the subset of the vertex set of the constraint region. However, when $R(x)$ is not single-valued the set of vertices of the inducible region is not necessarily a subset of the vertices of the polyhedral constraint region Ω , and the optimal solution for bi-level linear fractional problem does not necessarily occur at the vertices of Ω (for further discussion on this, interested readers may refer to [8].) That means, even if the optimistic approach is used, then vertex search methods cannot be applied in their usual sense unless all the vertices of the inducible region are known in advance.

Linear fractional optimization problems can also be equivalently converted to linear optimization problems by using either variable transformation approach [2, 14] or through the first order Taylor series approximation [1, 9, 13]. However, the resulting bi-level linear programming problem only locates its solutions if they are at the extreme points of the constraint region [3, 18, 17], which still fails to identify solutions that lie on the boundaries but not on the extreme points of the constraint region.

To the best knowledge of the authors, there is no exact method so far that can solve the general form of problem (3) if $R(x)$ is multi-valued for some feasible x . This is due to the fact that if $R(x)$ is multiple-valued for some feasible x , then the inducible region is not necessarily formed by the union of the faces of the polyhedral constraint region Ω as indicated in [10]. This implies that some vertices of the inducible region do not coincide with the vertices of the polyhedral region. Therefore, the methods that are reviewed above, including those described in [5, 7] cannot solve problem (3) when $R(x)$ is multi-valued for some feasible x as they miss some vertices of the inducible region that do not belong to the vertex set of the constraint region.

Contributions – The purpose of this article is to propose a procedure that can solve linear fractional bi-level problems by using the k th best solution technique together with the branch-and-bound method. A novel method is proposed in this article that helps to find the set of all feasible leader's variables that can induce multi-valued reaction map from the follower. Then, the coefficients of the objective function of the lower level problem are used to define the branching constraints, which contributes to formulation of an easily implementable solution algorithm for a general linear fractional bi-level programming problem. The proposed algorithm can also solve problems with single-valued reaction maps.

Outline – The paper is organized as follows: Section 2 provides review of some definitions and background concepts for the proposed method. Furthermore, the dependence of the actual relation between extreme points

of the inducible region and extreme points of the constraint region, on the structure of the optimal solution set of the lower level are shown using examples. The proposed algorithm is presented in Section 3. Section 4 shows the effectiveness of the algorithm by giving illustrative examples. Finally, some limitations of the proposed method and their possible extensions are highlighted in the conclusion part, Section 5.

2 Background of the proposed method

Before we start the solution procedure for a bi-level linear fractional problem, let us consider the maximization form of a linear fractional problem:

$$\begin{aligned} \max_x h(x) &= \frac{c_1^T x + \alpha_1}{c_2^T x + \alpha_2}, \\ \text{s.t. } x &\in S = \{Ax \leq b, x \geq 0\}. \end{aligned} \quad (6)$$

To assure existence of a solution, assume that the constraint set S is nonempty, closed, and bounded. Since the solution of a quasi-monotonic problem occurs at the extreme points of the feasible region and every linear fractional function is explicitly quasi-monotonic in its domain, the optimal solution of a linear fractional problem lies at some of the extreme points of the polyhedral constraint region [5, 6, 16]. Therefore, we search the optimal solution over extreme points of the constraint region. To do that, we start from one vertex of the constraint region, then move along a side adjacent to it such that the functional value increases. The process continues until an extreme point is obtained, where one cannot find a point at which the value of the function increases any more. The solution procedure is similar to the simplex method except for the formulation of the objective row. Since the objective function is linear fractional, it is a ratio of two linear functions. Then we can use a simplex-like method to solve the linear fractional problem by applying a few modifications as described in [2].

To formulate the appropriate modification, we consider the gradient of the objective function, which becomes

$$\nabla h = \frac{c_1(c_2^T x + \alpha_2) - c_2(c_1^T x + \alpha_1)}{(c_2 x + \alpha_2)^2}.$$

After rearranging, we get

$$\nabla h = \frac{1}{(c_2 x + \alpha_2)^2} (\alpha_2 c_1 - \alpha_1 c_2).$$

Since $\frac{1}{(c_2 x + \alpha_2)^2}$ is always positive for nonzero α_2 , the sign of ∇h depends on the sign of $\alpha_2 c_1 - \alpha_1 c_2$, and hence it is usually called the reduced cost. At each iteration of the simplex method, the value of $\alpha_2 c_1 - \alpha_1 c_2$ determines the direction of increase or decrease of h . Therefore, depending on the value of the coefficient $\alpha_2 c_1 - \alpha_1 c_2$ corresponding to the nonbasic variables, we have three possibilities for the next move in solving problem (6). The first possibility is when $\alpha_2 c_1 - \alpha_1 c_2 > 0$ corresponding to some nonbasic variables. In this case, the current extreme point is not an optimal solution for problem (6). The second possibility is when $\alpha_2 c_1 - \alpha_1 c_2 < 0$ corresponding to all nonbasic variables. In this case, we cannot make any improvement on the value of h , which means the current extreme point is an optimal solution for problem (6). However, when $\alpha_2 c_1 - \alpha_1 c_2 = 0$ for some nonbasic variables while $\alpha_2 c_1 - \alpha_1 c_2 < 0$ for all other nonbasic variables, the current extreme point is an optimal solution for problem (6) and there is a possibility for another alternative optimal solution.

At a basic feasible solution x , let $z_1 = -(c_1^T x + \alpha_1)$ and $z_2 = -(c_2^T x + \alpha_2)$ be the numerator and the denominator functions, respectively, of the objective function h . Then the corresponding simplex tableau becomes like in Table 1.

Now, by using the above concept, we have the following properties.

Theorem 1. For any linear fractional problem (6), with objective function $h(x) = \frac{c_1^T x + \alpha_1}{c_2^T x + \alpha_2}$, the problem has multiple optimal solutions if and only if $(c_1 \alpha_2 - c_2 \alpha_1)_i = 0$ for some i , and $(c_1 \alpha_2 - c_2 \alpha_1)_j < 0$ for all $j \neq i$, where i and j are indices for the nonbasic variables which make the reduced cost to be zero and negative, respectively.

Proof. Let the problem have multiple optimal solutions, say x_1 and x_2 , which are distinct. Then

Table 1: Simplex tableau for linear fractional problem given in (6)

| | | |
|-------|-------------------------------|---|
| h | $\alpha_2 c_1 - \alpha_1 c_2$ | $\frac{c_1^T x + \alpha_1}{c_2^T x + \alpha_2}$ |
| z_1 | c_1^T | $-(c_1^T x + \alpha_1)$ |
| z_2 | c_2^T | $-(c_2^T x + \alpha_2)$ |
| BV | x^T | RHS |
| x_B | A | b |

1. the reduced costs for x_1 and x_2 satisfy $\nabla h(x_1) \leq 0$ and $\nabla h(x_2) \leq 0$, that is, $(c_1 \alpha_2 - c_2 \alpha_1)_k \leq 0, \forall k$, where k is the index for the nonbasic variables at the given iteration.
2. $h(x_1) = h(x_2)$,
or equivalently,

$$(c_1^T x_1 + \alpha_1)(c_2^T x_2 + \alpha_2) = (c_1^T x_2 + \alpha_1)(c_2^T x_1 + \alpha_2).$$

After rearranging the values in the equality, we get

$$(\alpha_1 c_2 - \alpha_2 c_1)^T (x_1 - x_2) = 0.$$

Indeed since x_1 and x_2 are assumed to be distinct optimal solutions of the problem, $(x_1 - x_2)_i \neq 0$ for some i . Then we must have

$$(c_1 \alpha_2 - c_2 \alpha_1)_i = 0 \text{ for some } i,$$

and

$$(c_1 \alpha_2 - c_2 \alpha_1)_j < 0 \quad \text{for all other indices } j.$$

Conversely, let x_1 and x_2 be distinct feasible points that have different values corresponding to their i th components and the same values for each of their other components and both satisfy $(c_1 \alpha_2 - c_2 \alpha_1)_i = 0$ for some i and $(c_1 \alpha_2 - c_2 \alpha_1)_j < 0$ for all other indices j . Then since the components of the reduced cost of the problem (6) at both x_1 and x_2 are zero or negative,

depending on their corresponding functional values, either x_1 or x_2 or both are optimal solutions.

Let us check which condition is satisfied. From the given conditions, we have

$$(\alpha_1 c_2 - \alpha_2 c_1)^T (x_1 - x_2) = 0.$$

Equivalently we can write it as

$$(c_1^T x_1 + \alpha_1)(c_2^T x_2 + \alpha_2) = (c_1^T x_2 + \alpha_1)(c_2^T x_1 + \alpha_2).$$

Rearranging this equations gives

$$\frac{c_1^T x_1 + \alpha_1}{c_2^T x_1 + \alpha_2} = \frac{c_1^T x_2 + \alpha_1}{c_2^T x_2 + \alpha_2}.$$

Hence $h(x_1) = h(x_2)$, which means both x_1 and x_2 are optimal solutions that the problem has at least two optimal solutions. \square

When we return to the bi-level form of the problem, one searches the optimal solution over extreme points of the inducible region \mathcal{R} . If $R(x)$ is single-valued, then the solution of (3) occurs at the extreme points of the constraint region Ω , because extreme points of the inducible region \mathcal{R} are also extreme points of Ω [5], but this may not be the case when $R(x)$ is a nonsingleton map for some x [10].

To see what \mathcal{R} , may look like for a bi-level linear fractional problem, one may refer to Examples 1 and 2.

Example 1. Consider

$$\begin{aligned} & \max_x \frac{3x + 2y}{4x + y + 6}, \\ & \max_y \frac{-5x - 3y - 9}{x + 2y + 3}, \\ & \text{s.t.} \\ & \quad x + y \leq 5, \\ & \quad x + 3y \leq 10, \\ & \quad y \leq 3, \\ & \quad x, y \geq 0, \end{aligned}$$

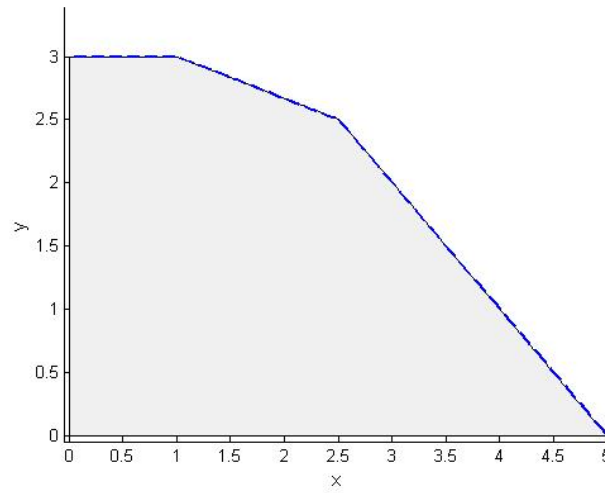


Figure 1: Constraint and inducible regions of Example 1

The common constraint region, Ω , and the inducible region \mathcal{R} of these examples are shown in Figure 1 and Figure 2, respectively. The hatched lines in these figures denote \mathcal{R} . The lower-level problem of Example 1 has multiple optimal solutions corresponding to the point $x = -1.2857$, which is not part of the feasible region Ω_1 . This means $R(x)$ of Example 1 is single-valued for all x in Ω_1 ; hence, \mathcal{R} is the union of faces of the polyhedron Ω as shown in Figure 1. That means, the set of extreme points of \mathcal{R} of Example 1 is $\{(5, 0), (0, 3), (1, 3), (2.5, 2.5)\}$, which is a subset of the set of extreme points of Ω , $\{(0, 0), (5, 0), (0, 3), (1, 3), (2.5, 2.5)\}$. In this case, the optimal solution is $(1, 3)$ found by using k th best or graphical method.

Example 2. Consider

$$\begin{aligned} & \max_x \frac{3x + 2y}{4x + y + 6}, \\ & \max_y \frac{4x + 3y}{4x + 6y + 3}, \\ & \text{s.t.} \\ & \quad x + y \leq 5, \\ & \quad x + 3y \leq 10, \\ & \quad y \leq 3, \\ & \quad x, y \geq 0, \end{aligned}$$

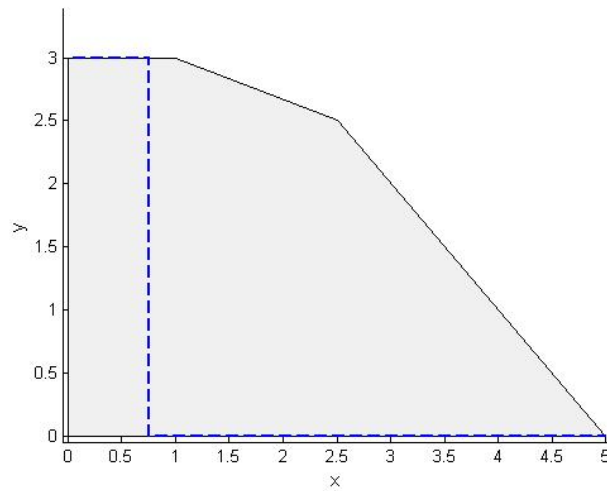


Figure 2: Constraint and inducible regions of Example 2

However, in Example 2, the lower level problem has multiple optimal solutions at $x = 0.75 \in \Omega_1$, which means $R(x)$ of Example 2 is not single-valued for at least one x in Ω_1 . In this case, it can be observed that some elements of \mathcal{R} are in the interior of Ω , and \mathcal{R} is not the union of faces of the polyhedron Ω as shown in Figure 2. The set of extreme points of Ω of Example 2 is $\{(0,0), (5,0), (0,3), (1,3), (2.5,2.5)\}$ whereas the set of extreme points of \mathcal{R} of Example 2 is $\{(5,0), (0,3), (0.75,0), (0.75,3)\}$, which is not a subset of the set of extreme points of Ω . Using k th best method one can obtain $(0,3)$ as the maximum point. However, this point is not the optimal

solution of the problem. The optimal solution is $(0.75, 3)$ found by inspection, and it is not part of the set of extreme points of Ω , rather, it is a boundary point of Ω .

Our main focus in this article is the case where $R(x)$ is a nonsingleton set for some x . To design a solution approach for such cases in general, we need to establish the following preliminary results.

Theorem 2. The optimal solution of the bi-level linear fractional problem (3) occurs generally at the boundary points of its constraint region.

Proof. For a fixed point x , we have $\mathcal{R} = \{(x, y) \in \Omega : y \in R(x)\} \subseteq \{(x, y) : y \in \Omega(x)\} \subseteq \Omega$.

The intersections of the plane that contains x and the constraint set Ω is the set $\{(x, y) : y \in \Omega(x)\}$. The extreme points of the set $\{(x, y) : y \in \Omega(x)\}$ lie on the boundaries of Ω .

Since the objective functions are linear fractional, $R(x)$ is either an extreme point of $\Omega(x)$ (if it is single-valued), or it is a convex combination of some extreme points of $\Omega(x)$ (if it is multi-valued). In both cases, the set of extreme points of $R(x)$ is the subset of extreme points of $\Omega(x)$. Extreme points of \mathcal{R} are extreme points of the set $\{(x, y) \in \Omega : y \in R(x)\}$ and hence the subset of extreme points of $\{(x, y) \in \Omega : y \in \Omega(x)\}$. From these arguments, one can conclude that *extreme points* of \mathcal{R} lie on the boundaries of Ω . \square

When $R(x)$ is nonsingleton, the difficulty in the use of the k th best algorithm (or any of the so far known methods, for that matter), is obtaining the extreme points of \mathcal{R} , which are not part of the extreme points of Ω , but those are boundary points of Ω .

To address this difficulty, we first need to find all feasible variables of the leader that make the optimal reaction set of the follower multi-valued. The following theorem helps us to obtain those points.

Theorem 3. For a fixed \bar{x} in problem (3), if

1. $(D\bar{x} + \beta)_i = 0$ for some i and $(D\bar{x} + \beta)_j \leq 0$ for all $i \neq j$, where

$$D = \begin{pmatrix} d_{12} & d_{22} \end{pmatrix} \begin{pmatrix} d_{21}^T \\ -d_{11}^T \end{pmatrix} \text{ and } \beta = \begin{pmatrix} d_{12} & d_{22} \end{pmatrix} \begin{pmatrix} \alpha_{22} \\ -\alpha_{21} \end{pmatrix},$$

$$2. \bar{\alpha}_{22} = \alpha_{22} + d_{21}^T \bar{x} \neq 0,$$

then the lower level problem has multiple optimal solutions.

Proof. The lower level problem (4) of (3) at a fixed point \bar{x} , can be rewritten as

$$\begin{aligned} \max_y \quad & f(\bar{x}, y) = \frac{d_{11}^T \bar{x} + d_{12}^T y + \alpha_{21}}{d_{21}^T \bar{x} + d_{22}^T y + \alpha_{22}} = \frac{d_{12}^T y + \bar{\alpha}_{21}}{d_{22}^T y + \bar{\alpha}_{22}} \\ \text{s.t.} \quad & y \in \Omega(\bar{x}), \end{aligned}$$

where

$$\bar{\alpha}_{21} = \alpha_{21} + d_{11}^T \bar{x} \text{ and } \bar{\alpha}_{22} = \alpha_{22} + d_{21}^T \bar{x},$$

and the problem is well defined for $\bar{\alpha}_{22} \neq 0$.

Let $(D\bar{x} + \beta)_i = 0$ for some i and $(D\bar{x} + \beta)_j \leq 0$ for all $j \neq i$.

Then

$$\begin{aligned} D\bar{x} + \beta &= \begin{pmatrix} d_{12} & d_{22} \end{pmatrix} \begin{pmatrix} d_{21}^T \\ -d_{11}^T \end{pmatrix} \bar{x} + \begin{pmatrix} d_{12} & d_{22} \end{pmatrix} \begin{pmatrix} \alpha_{22} \\ -\alpha_{21} \end{pmatrix} \\ &= (d_{12}d_{21}^T - d_{22}d_{11}^T) \bar{x} + d_{12}\alpha_{22} - d_{22}\alpha_{21} \\ &= d_{12}(d_{21}^T \bar{x} + \alpha_{22}) - d_{22}(d_{11}^T \bar{x} + \alpha_{21}) \\ &= d_{12}\bar{\alpha}_{22} - d_{22}\bar{\alpha}_{21}. \end{aligned}$$

Since $(D\bar{x} + \beta)_i = 0$, so $(d_{12}\bar{\alpha}_{22} - d_{22}\bar{\alpha}_{21})_i = 0$ for some i . Then by Theorem 1 the lower level problem (4) has multiple optimal solutions. \square

In the following section, we shall formulate a solution procedure for bi-level linear fractional problems with possible multiple optimal reaction values from the lower level, based on the above preliminary results.

3 The proposed solution algorithm

It has been indicated in [5, 7] that bi-level linear fractional problems of type (3) can be solved by using the k th best (or vertex-searching) approach when the reaction set is a singleton for each feasible decision of the upper level. Since optimal solutions of problem (3) occur at the extreme points of the inducible region, the k th best solution approach cannot solve problem (3)

when the rational reaction map is multi-valued for some feasible decision of the upper level. This is due to the fact that if the rational reaction map is multi-valued for some feasible points, then there are some extreme points of the inducible region, which are not part of extreme points of the constraint region, and they cannot be visited by the k th best solution approach. However, if we branch the problem at those feasible points, where the reaction map is multiple-valued, then we can make the k th best method to visit all extreme points of the inducible region. The branching constraints are formulated by using Theorem 3 and then incorporated into the relaxed problem (5).

To this end, let D have n rows. Then by Theorem 3, problem (4) has multiple solutions if $D_i x = -\beta_i$ and $D_j x \leq -\beta_j$ for all $j \neq i$. To get the branching constraint, we consider $D_i x \leq -\beta_i$ and $D_i x \geq -\beta_i$ in place of $D_i x = -\beta_i$. Therefore, for each $i \leq n$, we get two problems:

$$\begin{aligned} \max_{x,y} F(x,y) &= \frac{c_{11}^T x + c_{12}^T y + \alpha_{11}}{c_{21}^T x + c_{22}^T y + \alpha_{12}}, \\ \text{s.t.} \quad & A_1 x + A_2 y \leq b, \\ & D_i x \leq -\beta_i, \\ & D_j x \leq -\beta_j, \\ & x, y \geq 0, \\ & i \neq j; \end{aligned} \tag{7}$$

and

$$\begin{aligned} \max_{x,y} F(x,y) &= \frac{c_{11}^T x + c_{12}^T y + \alpha_{11}}{c_{21}^T x + c_{22}^T y + \alpha_{12}}, \\ \text{s.t.} \quad & A_1 x + A_2 y \leq b, \\ & D_i x \geq -\beta_i, \\ & D_j x \leq -\beta_j, \\ & x, y \geq 0, \\ & i \neq j. \end{aligned} \tag{8}$$

This branching procedure will result in $2 \times n$ problems in total. However, the first part (problem (7)) of the branching appears in all the cases. That means, the same problem is to be repeatedly solved in each case (n times). To avoid this repetition, we first consider

$$\begin{aligned} \max_{x,y} F(x,y) &= \frac{c_{11}^T x + c_{12}^T y + \alpha_{11}}{c_{21}^T x + c_{22}^T y + \alpha_{12}}, \\ \text{s.t.} \quad & \\ & A_1 x + A_2 y \leq b, \\ & Dx \leq -\beta, \\ & x, y \geq 0, \end{aligned} \tag{9}$$

once, and then we solve the next branch for each i . Finally, we only have $n + 1$ problems to be solved all together. In proposing the algorithm, we assume that the inducible region \mathcal{R} is nonempty and an *optimistic* version of the problem is considered.

At each iterations, first, we solve either problem (9) (in the first iteration) or (8) by using the simplex procedure. There may be a solution to each of the branched bi-level problems or not. If we have a solution, then the next step is to find a bi-level solution by using the k th best approach. Indeed the obtained solution could be infeasible, or it may have appeared in one of the previous iterations, or the objective value at this iteration may not be better than those in the other branches.

Now, let us define some sets, which are to be used in Algorithm 1. Let N denote the set of bi-level infeasible points from among the extreme points of the feasible region, let S be the set of bi-level feasible points, let E^i be the set of extreme points, which are candidates of optimal solution at the i th iteration, and let A^i be the set of adjacent extreme points of (x^i, y^i) at the i th iteration. By making each of the nonbasic variables as an entering variable in the tableau corresponding to (x^i, y^i) , we obtain elements of E^i and the set A^i at each iteration i . Let LB be a lower bound of problem (3), and its value can be updated if a bi-level feasible point with a better upper level objective value is obtained.

Algorithm 1 Algorithm for bi-level linear fractional problem with possible multiple optimal responses

- Step 0. $i = 0, N = \emptyset, S = \emptyset, LB = -\infty$, and n is equal to the number of rows of D .
- Step 1. Solve problem (9) by using the simplex method.
- If it has no solution, then go to Step 3.
 - If it has a solution (x^i, y^i) , then set $E^i = \{(x^i, y^i)\}$ and go to Step 2.
- Step 2. Solve the lower-level problem (4) by fixing x^i , using the simplex procedure to get \hat{y} .
- If $\hat{y} = y^i$, then set $LB = F(x^i, y^i)$, $(x^*, y^*) = (x^i, y^i)$, $S = S \cup \{(x^i, y^i)\}$, and go to Step 3.
 - If not, then
 - find the set of adjacent extreme points, A^i , of (x^i, y^i) and $N = N \cup \{(x^i, y^i)\}$, $E^i = (E^i \cup A^i) \setminus N$.
 - solve $\max \{F(x, y) : (x, y) \in E^i\}$ to obtain (\bar{x}^i, \bar{y}^i) and set $(x^i, y^i) = (\bar{x}^i, \bar{y}^i)$.
 - * If $(x^i, y^i) \in S$, then go to Step 3.
 - * Otherwise, repeat Step 2 with the updated values of x^i and y^i .
- Step 3. Set $i = i + 1$
- If $i \leq n$, then go to Step 4.
 - If $i > n$ (all the branching options are already explored), then stop, and set the optimal solution to be (x^*, y^*) .
- Step 4. Solve problem (8) by using the simplex procedure.
- If it has no solution, then go to Step 3.
 - If the problem has a feasible solution, then let (x^i, y^i) be the solution and
 - if $F(x^i, y^i) < LB$ or $(x^i, y^i) \in S$ or $(x^i, y^i) \in N$, then go to Step 3,
 - otherwise let $E^i = \{(x^i, y^i)\}$, and go to Step 2.
-

Theorem 4. The solution procedure described in Algorithm 1 terminates to the solution of problem (3) after finite iterations.

Proof. In Algorithm 1, there are at most $n + 1$ iterations, where n is the dimension of the lower level decision variable vector, and at each iteration the k th best algorithm was used to solve the problem. The convergence of k th best algorithm is proved in [5]. At each iteration if the problem has a solution, then we must check whether we need to further use k th best algorithm or not by using three conditions. The first one is comparing the value of the optimal solution with LB and if it has worst value, then we do not consider it any further. The second condition is existence of the solution in the nonfeasible set N . Again if the solution is in N , then we do not consider it further. The final condition is about the occurrence of the solution in the set S . If the solution is in S , then we do not consider it further as it was already considered in the previous steps and its value was compared with LB . These three conditions remove the unwanted repetition in the algorithm. After completing the $n + 1$ iterations, the point corresponding to the LB becomes the solution of (3).

Since the branching constraints make the boundary points of Ω that coincide with extreme points of \mathcal{R} to be vertices of the branched region, all the feasible extreme points of the inducible region \mathcal{R} are visited by Algorithm 1. Hence the final solution is the global optimal solution of problem (3). \square

4 Illustrative examples

In order to test our proposed Algorithm, we consider some numerical examples below, some taken from literature to check the validity of the output of the algorithm, and others are newly constructed to test for the additional conditions.

Here below, we present the solution of two examples by showing all the detailed procedures to demonstrate how the steps in the proposed algorithm work.

Example 3.

$$\begin{aligned} & \max_x \frac{3x + 2y}{4x + y + 6}, \\ & \max_y \frac{4x + 3y}{4x + 6y + 3}, \\ & \text{s.t.} \\ & \quad x + y \leq 5, \\ & \quad x + 3y \leq 10, \\ & \quad y \leq 3, \\ & \quad x, y \geq 0. \end{aligned}$$

This is the problem presented in Example 2 above and the procedures of the solution are presented in the following detailed steps. Note that, existing methods cannot automatically address such a problem as it has a nonsingleton reaction map.

To check existence of multi-valued reaction, we first formulate

$$D = \begin{pmatrix} 3 & 6 \end{pmatrix} \begin{pmatrix} 4 \\ -4 \end{pmatrix} = -12, \text{ and } \beta = \begin{pmatrix} 3 & 6 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \end{pmatrix} = 9.$$

Then we follow the steps below.

Step 0. $i = 0, N = \emptyset, S = \emptyset, LB = -\infty$ and $n = 1$ (as D has only 1 row).

Step 1. Solve

$$\begin{aligned} & \max_x \frac{3x + 2y}{4x + y + 6}, \\ & \text{s.t.} \\ & \quad x + y \leq 5, \\ & \quad x + 3y \leq 10, \\ & \quad y \leq 3, \\ & \quad -12x \leq -9, \\ & \quad x, y \geq 0. \end{aligned}$$

- When we solve this linear fractional problem using the simplex like method, we obtain a solution $(1, 3)$ with $x^0 = 1, y^0 = 3$. Then we set $E^0 = \{(1, 3)\}$ and go to Step 2.

Step 2. Solve the lower level problem (4) of Example 3 by fixing $x^0 = 1$, to get $\hat{y} = 0$. Then

- $\hat{y} \neq y^0$. Hence we obtain adjacent extreme points of (x^0, y^0) :
- $A^0 = \{(0.75, 3), (2.5, 2.5)\}$,
 $N = N \cup \{(x^0, y^0)\} = \{(1, 3)\}$, $E^0 = (E^0 \cup A^0) \setminus N = A^0$.
- Solve $\max \{F(x, y) : (x, y) \in E^0\}$, to get $(0.75, 3) \notin S$. Then update $x^0 = 0.75, y^0 = 3$ and repeat Step 2.

Solve the lower level problem (4) of Example 3 by fixing $x^0 = 0.75$, to obtain $\hat{y} = 3$.

- Since $\hat{y} = y^0$, set $LB = F(0.75, 3) = 0.6875, S = S \cup \{(0.75, 3)\} = \{(0.75, 3)\}$, $(x^*, y^*) = (0.75, 3)$ and go to Step 3.

Step 3. $i = 0 + 1 = 1$

- Since i satisfies $i \leq n$, go to Step 4.

Step 4. Solve

$$\begin{aligned} \max_x \quad & \frac{3x + 2y}{4x + y + 6}, \\ \text{s.t.} \quad & x + y \leq 5, \\ & x + 3y \leq 10, \\ & y \leq 3, \\ & -12x \geq -9, \\ & x, y \geq 0. \end{aligned}$$

- Then we get a solution: $(0.75, 3)$, with $x^1 = 0.75, y^1 = 3$.
 – Since $(0.75, 3) \in S$, go to Step 3.

Step 3. $i = 1 + 1 = 2$

- Since i does not satisfy $i \leq n$, Stop.

Hence the optimal solution is $(x, y) = (0.75, 3)$ with the upper level optimal value $F = 0.6875$ and the lower level optimal value $f = 0.5$.

Example 4. A newly constructed problem with nonunique reaction set.

$$\begin{aligned} \max_x \quad & \frac{-x_1 + x_2 + 2y_1 - 2y_2 - y_3 - 1}{-x_1 - 2y_1 + y_2 + 5y_3 + 8}, \\ \max_y \quad & \frac{x_1 + x_2 - 2y_1 + y_2 - y_3 - 2}{2x_1 + y_1 + y_2 + 3y_3 + 1}, \\ \text{s.t.} \quad & \\ & -y_1 + y_2 + y_3 \leq 1, \\ & 2x_1 - y_1 + 2y_2 + 0.5y_3 \leq 3, \\ & 2x_2 + 2y_1 - y_2 + 0.5y_3 \leq 9, \\ & x_1, x_2, y_1, y_2, y_3 \geq 0, \end{aligned}$$

with $x = (x_1, x_2), y = (y_1, y_2, y_3)$.

Solution of Example 4: In this case we have

$$D = \begin{pmatrix} -2 & 1 \\ 1 & 1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} -5 & -1 \\ 1 & -1 \\ -5 & -3 \end{pmatrix}, \text{ and } \beta = \begin{pmatrix} -2 & 1 \\ 1 & 1 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \\ 5 \end{pmatrix}.$$

Step 0. $i = 0, N = \emptyset, S = \emptyset, LB = -\infty$, and $n = 3$ (number of rows of D).

Step 1. Solve

$$\begin{aligned} \max_{x,y} \quad & \frac{-x_1 + x_2 + 2y_1 - 2y_2 - y_3 - 1}{-x_1 - 2y_1 + y_2 + 5y_3 + 8}, \\ \text{s.t.} \quad & \\ & -y_1 + y_2 + y_3 \leq 1, \\ & 2x_1 - y_1 + 2y_2 + 0.5y_3 \leq 3, \\ & 2x_2 + 2y_1 - y_2 + 0.5y_3 \leq 9, \\ & -5x_1 - x_2 \leq 0, \\ & x_1 - x_2 \leq -3, \\ & -5x_1 - 3x_2 \leq -5, \\ & x_1, x_2, y_1, y_2, y_3 \geq 0. \end{aligned}$$

- After solving, we obtain a solution: $(0, 3, 1.5, 0, 0)$ with $x^0 = (0, 3), y^0 = (1.5, 0, 0)$. Then set $E^0 = \{(0, 3, 1.5, 0, 0)\}$ and go to Step 2.

Step 2. Solve the lower level problem (4) of Example 4 by fixing $x^0 = (0, 3)$, to get $\hat{y} = (1.5, 0, 0)$.

- Since $\hat{y} = y^0$, set $LB = F(0, 3, 1.5, 0, 0) = 1$, $S = S \cup \{(0, 3, 1.5, 0, 0)\} = \{(0, 3, 1.5, 0, 0)\}$, $(x^*, y^*) = (0, 3, 1.5, 0, 0)$ and go to Step 3.

Step 3. $i = 1$

- Since i satisfies $i \leq n$, go to Step 4.

Step 4. Solve the branched problem (complementing the condition: $-5x_1 - x_2 \leq 0$):

$$\begin{aligned} \max_{x,y} \quad & \frac{-x_1 + x_2 + 2y_1 - 2y_2 - y_3 - 1}{-x_1 - 2y_1 + y_2 + 5y_3 + 8}, \\ \text{s.t.} \quad & \\ & -y_1 + y_2 + y_3 \leq 1, \\ & 2x_1 - y_1 + 2y_2 + 0.5y_3 \leq 3, \\ & 2x_2 + 2y_1 - y_2 + 0.5y_3 \leq 9, \\ & -5x_1 - x_2 \geq 0, \\ & x_1 - x_2 \leq -3, \\ & -5x_1 - 3x_2 \leq -5, \\ & x_1, x_2, y_1, y_2, y_3 \geq 0. \end{aligned}$$

- Since we have no solution for this problem, go to Step 3.

Step 3. $i = 2$

- Since i satisfies $i \leq n$, go to Step 4.

Step 4. Solve the branched problem (complementing the condition that $x_1 - x_2 \leq -3$):

$$\begin{aligned} \max_{x,y} \quad & \frac{-x_1 + x_2 + 2y_1 - 2y_2 - y_3 - 1}{-x_1 - 2y_1 + y_2 + 5y_3 + 8}, \\ \text{s.t.} \quad & \\ & -y_1 + y_2 + y_3 \leq 1 \\ & 2x_1 - y_1 + 2y_2 + 0.5y_3 \leq 3, \\ & 2x_2 + 2y_1 - y_2 + 0.5y_3 \leq 9, \\ & -5x_1 - x_2 \leq 0, \\ & x_1 - x_2 \geq -3, \\ & -5x_1 - 3x_2 \leq -5, \\ & x_1, x_2, y_1, y_2, y_3 \geq 0. \end{aligned}$$

- Then we obtain a solution, $(3.75, 0, 4.5, 0, 0)$ with $x^2 = (3.75, 0), y^2 = (4.5, 0, 0)$.
- Since $F(3.75, 0, 4.5, 0, 0) = -0.89 \leq LB$, go to Step 3.

Step 3. $i = 3$

- Since i satisfies $i \leq n$, go to Step 4.

Step 4. Solve the branched problem (complementing the condition that $-5x_1 - 3x_2 \leq -5$):

$$\begin{aligned} \max_{x,y} \quad & \frac{-x_1 + x_2 + 2y_1 - 2y_2 - y_3 - 1}{-x_1 - 2y_1 + y_2 + 5y_3 + 8}, \\ \text{s.t.} \quad & \\ & -y_1 + y_2 + y_3 \leq 1, \\ & 2x_1 - y_1 + 2y_2 + 0.5y_3 \leq 3, \\ & 2x_2 + 2y_1 - y_2 + 0.5y_3 \leq 9, \\ & -5x_1 - x_2 \leq 0, \\ & x_1 - x_2 \leq -3, \\ & -5x_1 - 3x_2 \geq -5, \\ & x_1, x_2, y_1, y_2, y_3 \geq 0. \end{aligned}$$

- Since this problem has no solution, go to Step 3.

Step 3. $i = 4$

- Since i does not satisfy $i \leq n$, Stop.

Therefore, the optimal solution is $(x_1^*, x_2^*, y_1^*, y_2^*, y_3^*) = (0, 3, 1.5, 0, 0)$ with $F = 1$ is the upper level optimal value and $f = -0.8$ is the lower level optimal value.

The same algorithmic procedure can be used to solve linear fractional bi-level optimization problems with unique optimal response from the follower for each of the choices of variables of the leader. The examples below are taken from literature whose exact solutions were calculated; and we obtain the same result (shown in Table 2) for each one of them as in the references indicated.

Note that the purpose of the examples here below is not to compare the efficiency of the algorithm rather to show that the same exact solution can be obtained using the proposed algorithm as well, while it solve problems with multiple optimal response from the lower level. It is known that the methods given in each of the references for these problems fail to solve if the optimal response from the lower level is nonunique.

Example 5. Consider a bi-level problem from [5]

$$\begin{aligned} \max_x \quad & \frac{-x - 3y - 3}{x + y + 5}, \\ \max_y \quad & \frac{x - 2y - 7}{x + y + 2}, \\ \text{s.t.} \quad & x + 2y \leq 20, \\ & x + y \leq 12, \\ & 2x + y \leq 20, \\ & 3x - 4y \leq 19, \\ & x - 4y \leq 5, \\ & x, y \geq 0, \end{aligned}$$

Example 6. Consider a bi-level problem from [10]

$$\begin{aligned}
 & \max_x \frac{-y+2}{x+y+1}, \\
 & \max_y \frac{-5x-4y-5}{5x+5y+10}, \\
 & \text{s.t.} \quad 3x-2y \geq -5, \\
 & \quad \quad 2x+9y \leq 69, \\
 & \quad \quad 3x-2y \leq 26, \\
 & \quad \quad x-6y \leq -2, \\
 & \quad \quad x+y \geq 5, \\
 & \quad \quad x, y \geq 0.
 \end{aligned}$$

Example 7. Consider a bi-level problem from [6]

$$\begin{aligned}
 & \max_x \frac{-2x-3y_1-y_2-2}{x+6y_2+5}, \\
 & \max_y \frac{-3x-2y_1-y_2}{x+y_1+2y_2+1}, \\
 & \text{s.t.} \quad x+y_2 \leq 1, \\
 & \quad \quad y_1+y_2 \leq 1, \\
 & \quad \quad x, y_1, y_2 \geq 0,
 \end{aligned}$$

$$y = (y_1, y_2).$$

Example 8. Consider a bi-level problem from [5]

$$\begin{aligned}
 & \max_x \frac{-x_1+x_2-2y_2-1}{-x_1-2y_1+y_2+5y_3+8}, \\
 & \max_y \frac{-x_1-x_2-2y_1+y_2-y_3-1}{2x_1+y_1+y_2-3y_3+6}, \\
 & \text{s.t.} \quad -y_1+y_2+y_3 \leq 1, \\
 & \quad \quad 2x_1-y_1+2y_2-0.5y_3 \leq 1, \\
 & \quad \quad 2x_2+2y_1-y_2-0.5y_3 \leq 1, \\
 & \quad \quad x_1, x_2, y_1, y_2, y_3 \geq 0,
 \end{aligned}$$

$$x = (x_1, x_2), y = (y_1, y_2, y_3).$$

Table 2: Summery of solutions for the problems in the examples.

| Examples | optimal solution using the proposed algorithm |
|----------|---|
| 5 | $(1, 0)$ |
| 6 | $(3, 2)$ |
| 7 | $(0.2, 0, 0.8)$ |
| 8 | $(0.75, 0.75, 0, 0, 1)$ |

5 Conclusion

In this paper, we presented a vertex search method to find an exact global optimal solution to the continuous bi-level linear fractional programming problem. Our algorithm is a combination of the k th best method and a branch-and-bound procedure. The existing k th best method is known to find a global optimal solution for bi-level linear fractional problems with single valued reaction set for all upper level decisions. To overcome the limitations of the k th best method when there are nonsingleton optimal reaction sets for some upper level decisions, a new algorithm that combines the k th best method together with a branch-and-bound mechanism is proposed. In this algorithm, iterative solution procedure is applied, where the branch-and-bound method is used to branch the problem into two problems of the same type in each branching step. We implemented the algorithm using the MATLAB software and it can solve the optimistic version of any bi-level linear fractional problem. The algorithm can also be applied for solving bi-level problems when the objective functions are generally quasi-convex and the constraints are polyhedral.

The algorithm performs well in solving linear fractional bi-level programming problems of any kind. However, if the optimal response map of the lower-level is single valued for all feasible upper level variables, then some steps of the algorithm will still run to check if there are possible feasible solutions outside of the vertices of the constraint region. This might create unnecessary delay in the solution process. In the future one may try to develop a mechanism to avoid the process of execution of the unnecessary iterations within the framework of the proposed algorithm.

Acknowledgements

The first author acknowledges the International Science Program (ISP) of Sweden, for providing financial support through a research project at the Department of Mathematics, Addis Ababa University.


References

- [1] Adhami, A. and Kausar, H. *bi-level Multi-Objective Stochastic Linear Fractional Programming with General Form of Distribution*, Stat., Optim. Inf. Comput. 7 (2019), 407–416.
- [2] Bajalinov, E.B. *Linear-Fractional Programming: Theory, Methods, Applications and Software*, Springer New York, USA, 2003.
- [3] Bialas, W.F. and Karwan, M.H. *Mathematical Methods for Multilevel Planning*, Technical report No. 79-2, Department of Industrial Engineering, State University of New York at Buffalo, New York, 1979.
- [4] Calvete, H.I. and Galé, C. *The bi-level linear/linear fractional programming problem*, Eur. J. Oper. Res. 114 (1999), 188–197.
- [5] Calvete, H.I. and Galé, C. *Solving linear fractional bi-level programs*, Oper. Res. Lett. 32 (2004), 143–151.
- [6] Calvete, H.I., Galé, C. and Mateo, P.M. *A genetic algorithm for solving linear fractional bi-level problems*, Ann. Oper. Res. 166 (2009), 39–56.
- [7] Chen, H.J. *A two-level vertex-searching global algorithm framework for bi-level linear fractional programming problems*, Syst. Sci. Control Eng. 8 (2020), 488–499.
- [8] Dempe, S. *Foundations of bi-level programming*, Springer New York, NY, USA, 2002.
- [9] Emmami, M. and Osgooei, E. *An algorithm to solve linear fractional bi-level problems based on Taylor approximation*, J. Ind. Syst. Eng. 14(4) (2022), 279–292.

- [10] Mathur, K. and Puri, M.C. *On bi-level fractional programming*, Optim. 35 (1995) 215–226.
- [11] Mishra, S. *Weighting method for bi-level linear fractional programming problems*, Eur. J. Oper. Res. 183 (2007), 296–302.
- [12] Pollak, E.G., Novaes, G.N. and Frankel, E.G. *On the optimization and integration of shipping ventures*, Int. Shipbuild. Prog. 12 (131) (1965), 267–281.
- [13] Pramanik, S. and Banerjee, D. *Chance Constrained Multi-Objective Linear Plus Linear Fractional Programming Problem Based on Taylor's Series Approximation*, Int. J. Eng. Res. Develop. 1(3) (2012), 55–62.
- [14] Stancu-Minasian, I.M. *Fractional programming: theory, methods and applications*, Vol. 409. Springer Science & Business Media, 2012.
- [15] Toksari, D.M. *Taylor series approach for bi-level linear fractional programming*, Selçuk J Appl Math, 11 (2010), 63–69.
- [16] Wang, G., Ziyou G. and Zhongping W. *A global optimization algorithm for solving the bi-level linear fractional programming problem*, Computers and Industrial Engineering, 63 (2012), 428–432.
- [17] Wen, U.-P. and Bialas, W. F. *The hybrid algorithm for solving the three-level linear programming problem*, Comput. Oper. Res. 13 (1986), 367–377.
- [18] White, D.J. *Penalty function approach to linear trilevel programming*, J. Optim. Theory Appl. 93 (1997), 183–197.



An adaptive scheme for the efficient evaluation of integrals in two-dimensional boundary element method

R. Si Hadj Mohand*, , Y. Belkacemi and S. Rechak

Abstract

An efficient analysis with the boundary element method requires an accurate evaluation of all the boundary integrals. Typically, nonsingular integrals are solved numerically using Gauss quadrature. Therefore, the

*Corresponding author

Received 14 May 2025; revised 6 August 2025; accepted 17 August 2025

Rahim Si Hadj Mohand

Laboratory of Green Mechanics and Development (LGMD), Department of mechanical engineering, Ecole Nationale Polytechnique, Algiers, Algeria. e-mail: rahim.si_hadj_mohand@g.enp.edu.dz

Yacine Belkacemi

Laboratory of Green Mechanics and Development (LGMD), Department of mechanical engineering, Ecole Nationale Polytechnique, Algiers, Algeria. e-mail: yacine.belkacemi@g.enp.edu.dz

Said Rechak

Laboratory of Green Mechanics and Development (LGMD), Department of mechanical engineering, Ecole Nationale Polytechnique, Algiers, Algeria. e-mail: said.rechak@g.enp.edu.dz

How to cite this article

Si Hadj Mohand, R., Belkacemi, Y. and Rechak, S., An adaptive scheme for the efficient evaluation of integrals in two-dimensional boundary element method. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1420-1463. <https://doi.org/10.22067/ijnao.2025.93526.1647>

development of criteria and schemes that determine the appropriate Gauss order while maintaining a balance between accuracy and performance is of great importance.

In the present work, an adaptive integration criterion tailored for two-dimensional elasticity problems is introduced and verified. This criterion is formulated as an empirical formula, incorporating a parameter ranging from zero to unity. This parameter enables control over computational effort, making the criterion very efficient across a wide range of applications, from thick structures to extremely thin ones where near-singularities are pronounced.

The proposed integration criterion is tested on a very thin structure, where it showed a high degree of accuracy and effectiveness in solving problems with a very pronounced boundary layer effect. Additionally, the criterion demonstrated its advantage by reducing and moderating computational overhead in the case of pre-treatment of near-singularities by a semi-analytical technique or a variable transformation technique.

AMS subject classifications (2020): 74S15, 65R20, 65D30.

Keywords: Boundary integrals, Near-singularity, Gauss quadrature, Integration criterion, Thin structures.

1 Introduction

The accurate evaluation of boundary element integrals is of crucial importance in any boundary element method (BEM) analysis. The boundary integrals appearing in the BEM method involve kernel functions with terms of the form $\frac{1}{r^p}$ or $\log\left(\frac{1}{r}\right)$ with r the shortest distance between the source point and the boundary element. This nature causes a singular behavior when source points approach the boundary (r tends to zero). Thus, depending on the ratio $\lambda = \frac{r}{L}$ (with L the element length) integrals are classified into three major categories (Regular, Singular and Near-Singular), as schematized in Figure 1.

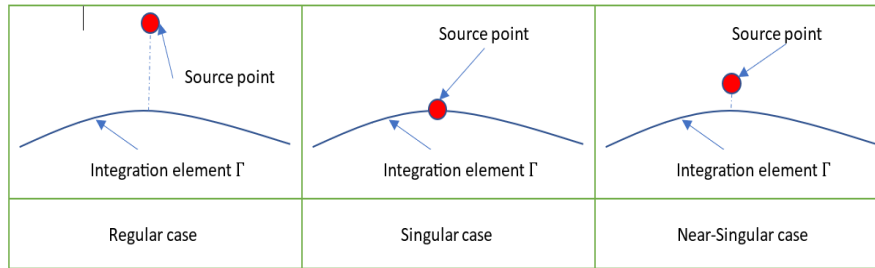


Figure 1: Boundary integral types

When the source point is sufficiently distant from the boundary element, the integral becomes regular. In this case, its numerical evaluation using Gauss quadrature with a relatively small order is sufficient. By contrast, when the source point coincides with the boundary element ($\lambda = 0$), the integral becomes singular. Several techniques have been proposed in the BEM literature to evaluate singular integrals, including analytical integration [32, 43, 44, 31, 47, 33], indirect methods [15, 17, 6, 7, 35], semi-analytical methods [16, 14, 37, 2], coordinate transformation [40, 20], and the use of singularity-reduced kernels [26]. Other methods directly formulate fundamental solutions, as in the novel scaled coordinate transformation BEM (SCTBEM [42, 19]), which converts the domain integral into a boundary integral, leading to the elimination of the low-order singularity.

The third type (near-singular integrals) arises when the source point is very close but not coinciding with the boundary element ($\lambda \approx 0$). While they are regular in nature and do not exhibit mathematical singularities, their evaluation is challenging due to steep variations of the integrand around the projection of the source point. This phenomenon is commonly known as the boundary layer effect, which arises in several applications of the BEM, such as thin-walled structures and thin coatings [3, 47, 37, 12, 46], crack-related problems [37, 5, 26, 30, 2, 33, 34], contact problems [8, 18], and near-boundary field calculations [6, 32, 13]. To deal with the boundary layer effect and near-singular integrations causing it, several techniques have been developed:

Element subdivision: A numerical technique proposed by Lachat and Watson [25], based on subdividing the original element into smaller subelements, thereby concentrating Gauss points around the projection of the source point.

Semi-analytical techniques: Methods such as singularity subtraction, proposed by Cruse and Aithal [9] and Mi and Aliabadi [28], are employed.

Variable transformation: This approach involves applying a nonlinear variable transformation that weakens the near-singular behavior and smooths the sharp peak of the integrand. Several transformations have been proposed, including the polynomial transformation of Telles [40], the optimal transformations of Sladek and Sladek [38], the distance transformation of Ma and Kamiya [27], and the sinh transformation of Johnston and Elliott [21], which was further extended by Gu et al. [12, 13] and Zhang, Gong, and Gao [45].

The use of Gauss quadrature is essential for most of the techniques cited above, either for evaluating the entire integral in numerical techniques or for evaluating the regular part and the transformed integral in semi-analytical techniques and variable transformations, respectively. Thus, determining the number of Gauss points for a given integral is of great importance. The set of rules and guidelines that determines the smallest order of quadrature guaranteeing a specified precision is called an integration criterion. The need to derive effective and precise criteria has led to several publications on this subject, where different integration criteria and upper-bound error formulas have been proposed.

The first work prior to any publications on this subject was the contribution of Stroud and Secrest [39], who proposed a formula for calculating the upper-bound error of Gaussian numerical integration. Based on this formula, Lachat and Watson [25] proposed the first integration criterion for functions of the form $\frac{1}{r^2}$, applicable to three-dimensional (3D) structures, which was further simplified by Mustoe [29] and Gao and Davies [11], who provided simpler approximate formulas for the upper-bound error estimate. Jun and Beer, [22] again used the upper-bound error formula of Stroud and Secrest [39] and proposed a new criterion for functions of the form $(\frac{1}{r^p}, p = 1, 2, 3)$, applicable in both (two-dimensional) 2D and 3D structures, presented in tabular format.

After performing an extensive numerical study on the error distribution around a flat rectangular 3D element, Bu and Davies [4] developed a new integration criterion for 3D problems. This criterion is presented both in tables and as empirical formulas and was further improved by Gao and Davies

[11], who proposed a unified approximation formula based on Bu and Davies' [4] numerical experiments.

Eberwien, Duencer, and Moser [10] followed a similar numerical strategy as Bu and Davies [4] and applied it in the 2D case by considering two reference elements, one flat and the other slightly curved. Unlike the research works cited above, the influence of the shape functions and the Jacobian was taken into consideration. Thus, the considered integrands are of the form

$$f(\xi) = (r(\xi))^p \times \Psi_m(\xi) \times J(\xi), \quad p = 1, 2.$$

The numerical study performed in [10] showed that the errors predicted by the previous publications [25, 39, 22, 4, 29, 11] were underestimated. This is due to the nonconsideration of the shape functions and the Jacobian. The criterion proposed by Eberwien, Duencer, and Moser was tested on a benchmark problem and showed a clear improvement in the results compared to its predecessors.

In 2020, Junhao, Zhipeng, and Yongqiang [23] accomplished an extensive numerical study on integrands of the form $(\frac{1}{r^p}, p = 1, 2, 3)$, and proposed a new upper-bound error estimate formula, which showed a certain gain in efficiency and precision compared to previous formulas. The gain in performance is especially visible at very small λ ratios.

In 2023, Zhou, Yang, and Chen [48] introduced a new adaptive scheme leveraging deep machine learning and AI technologies. Their approach categorizes the ratio λ into three distinct intervals, incorporates the sinh variable transformation for small values of λ , and predicts the required number of Gauss points using a trained neural network or its recorded data. This technique achieves a good level of precision, with significantly fewer Gauss points and reduced CPU time.

In the current study, a new integration criterion is introduced. It is formulated through empirical formulas, which determine the required number of Gauss points to attain one of three precision levels ($\epsilon = 10^{-2}, 10^{-3}, 10^{-4}$), depending on the λ ratio and the nature of the kernel. Specifically designed for 2D linear elasticity problems, this criterion was derived following an extensive numerical testing. Two bounding empirical formulas were established:

The lower-bound formula is deduced by considering integrands of the form

$$f(\xi) = \frac{1}{r(\xi)^p}, \quad p = 1, 2,$$

while the upper-bound one by considering the complete form of the integrands as encountered in 2D-elasticity BEM

$$f(\xi) = F(P(\xi), P_0) \times \Psi_m(\xi) \times J(\xi),$$

where $F(P(\xi), P_0)$ represents one of the kernel functions of 2D-elasticity BEM, $P(\xi)$ and P_0 denote the field and source points, respectively, and $\Psi_m(\xi)$ and $J(\xi)$ denote the shape function and the Jacobian. Finally, a third formula is defined by the combination of the two bounding ones, using a real parameter $\alpha \in [0, 1]$, which allows control over the computational effort and facilitates its adjustment depending on application requirements.

Tested on an extremely thin structure, the new criterion demonstrated better and more stable precision compared to existing methods, even in regions of extreme thinness where the achieved relative error is lower than the target upper bound. The results highlight an underestimation of errors by existing criteria and error-bound formulas, due to the nonconsideration of the complete form of the kernels. Another advantage observed is a performance enhancement for moderate values of the λ ratio compared to the formulas outlined in [23]. These moderate values of λ commonly arise when employing the element subdivision technique, which is preferred in BEM applications over the use of high-order quadratures, since very high quadrature orders lead to floating-point round-off errors. Additionally, Gauss–Legendre abscissas and weights are not computed dynamically at runtime; instead, a limited set of predefined abscissas and corresponding weights is typically available and prescribed in the program.

Finally, the proposed criterion incorporates a parameter that can effectively reduce computational effort, especially in cases where integrands are treated using coordinate transformation or other semi-analytical techniques that mitigate or eliminate the near-singular behavior.

2 A brief review of the BEM for elasticity problems

Solving elasticity problems using the BEM is achieved by interpreting the partial differential equations that govern the problem in the form of integral equations.

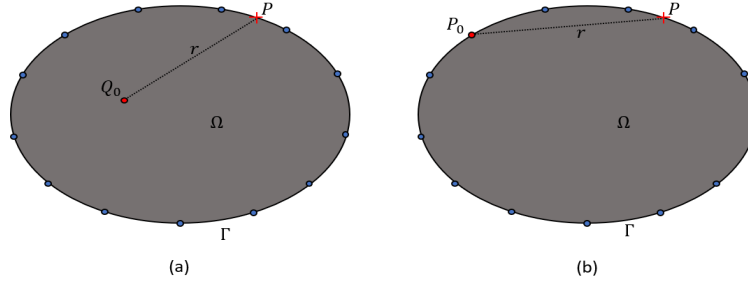


Figure 2: BEM Modeling of an elastic domain - (a) Source point inside the domain (b) source point on the boundary

Let us consider a linear elastic domain Ω with boundary $\Gamma = \partial\Omega$ (Figure 2). There are generally six integral equations that arise in any boundary element analysis. By neglecting the body forces, these equations are defined as [1, 6, 14]:

$$u_i(Q_0) = \int_{\Gamma} U_{ij}(Q_0, P) t_j(P) d\Gamma - \int_{\Gamma} T_{ij}(Q_0, P) u_j(P) d\Gamma, \quad (1)$$

$$\sigma_{ij}(Q_0) = \int_{\Gamma} D_{ijk}(Q_0, P) t_k(P) d\Gamma - \int_{\Gamma} S_{ijk}(Q_0, P) u_k(P) d\Gamma, \quad (2)$$

$$u_{i,k}(Q_0) = \int_{\Gamma} W_{ijk}(Q_0, P) t_j(P) d\Gamma - \int_{\Gamma} V_{ijk}(Q_0, P) u_j(P) d\Gamma, \quad (3)$$

$$C_{ij}(P_0) u_j(P_0) = \int_{\Gamma} U_{ij}(P_0, P) t_j(P) d\Gamma - \int_{\Gamma} T_{ij}(P_0, P) u_j(P) d\Gamma, \quad (4)$$

$$\frac{1}{2} \sigma_{ij}(P_0) = \int_{\Gamma} D_{ijk}(P_0, P) t_k(P) d\Gamma - \int_{\Gamma} S_{ijk}(P_0, P) u_k(P) d\Gamma, \quad (5)$$

$$C_{ikjl} u_{j,l}(P_0) = \int_{\Gamma} W_{ijk}(P_0, P) t_j(P) d\Gamma - \int_{\Gamma} V_{ijk}(P_0, P) u_j(P) d\Gamma, \quad (6)$$

where

- Q_0 and P_0 : the source points inside the domain and at the boundary, respectively ($Q_0 \in \Omega$, $P_0 \in \Gamma$);
- P : a field point located at the boundary ($P \in \Gamma$);
- $u_j(P)$, $t_j(P)$, $\sigma_{ij}(P)$ and $u_{i,k}(P)$: represent, respectively, the displacement, traction, stress tensor, and displacement derivative components at a point P ;
- $U_{ij}(P_0, P)$, $T_{ij}(P_0, P)$, $D_{ijk}(P_0, P)$, $S_{ijk}(P_0, P)$, $W_{ijk}(P_0, P)$, $V_{ijk}(P_0, P)$: the fundamental solutions or the kernels;
- C_{ij} and C_{ijkl} : Free terms that depend on the nature of the boundary. For a smooth boundary they are given by

$$C_{ij} = \begin{cases} 1/2 & i = j, \\ 0 & i \neq j, \end{cases} \quad \text{and} \quad C_{ijkl} = \begin{cases} 1/2 & ik = jl, \\ 0 & ik \neq jl. \end{cases}$$

Equations (1), (2), (3) give the displacement, the stress and the displacement derivative at an internal source point $Q_0 \in \Omega$, respectively. They are expressed in terms of the boundary variables $u_j(P)$ and $t_j(P)$ at a field point $P \in \Gamma$.

Equations (4), (5) and (6) are called boundary integral equations (BIEs). For a source point located at the boundary $P_0 \in \Gamma$, they give the same quantities as (1), (2), (3), respectively. To solve (1)–(6), the boundary Γ is discretized into a finite number N of iso-parametric elements. Each element Γ_n is composed of M nodes. Equations (1)–(6) can then be written in their discrete forms as

$$\begin{aligned} u_i(Q_0) = & \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [U_{ij}(Q_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad t_j^{nm} \\ & - \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [T_{ij}(Q_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad u_j^{nm}, \end{aligned} \quad (7)$$

$$\begin{aligned}\sigma_{ij}(Q_0) &= \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [D_{ijk}(Q_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad t_k^{nm} \\ &\quad - \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [S_{ijk}(Q_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad u_k^{nm},\end{aligned}\quad (8)$$

$$\begin{aligned}u_{i,k}(Q_0) &= \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [W_{ijk}(Q_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad t_j^{nm} \\ &\quad - \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [V_{ijk}(Q_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad u_j^{nm},\end{aligned}\quad (9)$$

$$\begin{aligned}C_{ij}(P_0)u_j(P_0) &+ \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [T_{ij}(P_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad u_j^{nm} \\ &= \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [U_{ij}(P_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad t_j^{nm},\end{aligned}\quad (10)$$

$$\begin{aligned}\frac{1}{2}\sigma_{ij}(P_0) &= \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [D_{ijk}(P_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad t_k^{nm} \\ &\quad - \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [S_{ijk}(P_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad u_k^{nm},\end{aligned}\quad (11)$$

$$\begin{aligned}C_{ikjl}u_{j,l}(P_0) &= \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [W_{ijk}(P_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad t_j^{nm} \\ &\quad - \sum_{n=1}^N \sum_{m=1}^M \int_{-1}^1 [V_{ijk}(P_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi \quad u_j^{nm}.\end{aligned}\quad (12)$$

From (7)–(12), we deduce that the boundary integrals involved in a BEM analysis have the general form:

$$I = \int_{-1}^1 [F(P_0, P(\xi)) \psi_m(\xi) |J_n(\xi)|] d\xi, \quad (13)$$

where

- $F(P_0, P(\xi))$: One of the fundamental solutions (kernels)
 $\{U_{ij}, T_{ij}, D_{ijk}, S_{ijk}, W_{ijk}, V_{ijk}\};$

- $\psi_m(\xi)$: The linear shape function corresponding to the node of index m ;
- $|J_n(\xi)|$: The Jacobian of the coordinate transformation from $d\Gamma$ to $d\xi$, corresponding to the element of index n .

3 The proposed integration criterion

Most of the existing integration criteria are developed under the assumption of simplified forms of integrands, such as $\left\{f(\xi) = \frac{1}{r(\xi)^p} \quad p = 1, 2, 3\right\}$ and $\left\{f(\xi) = \log\left(\frac{1}{r(\xi)}\right)\right\}$. Some criteria also incorporate shape functions and the Jacobian with $\left\{f(\xi) = \frac{1}{r(\xi)^p} \cdot \psi_m(\xi) \cdot J_n(\xi) \quad p = 1, 2\right\}$, as done by Eberwien, Duencer, and Moser [10]. These simplified assumptions yield satisfactory results in structures that are not extremely thin, especially in scalar problems like electrostatics, where kernel functions are typically simple. However, elasticity problems are vectorial in nature, involving heavy kernel functions that include directional and normal derivatives of the distance r . These complexities can exacerbate singularities and amplify the boundary layer effect in extremely thin structures. As a result, we are motivated to investigate the actual distribution of integration errors by considering the complete form of integrands encountered in planar elastic boundary element analysis and comparing them with simplified integrands. Subsequently, we propose a numerical scheme with a new criterion aimed at meeting high precision requirements, even in extremely thin structures, while optimizing computational resources and CPU time.

3.1 The numerical testing

Due to the absence of mathematical tools and techniques to define analytically error bounds for integrands in their complete form as specified in (13), numerical experimentation remains the primary recourse. The numerical testing methodology followed by Eberwien, Duencer, and Moser [10] is

adopted. It involves the utilization of a rectangular region encompassing a reference quadratic iso-parametric element. Within this region, a highly dense grid of points is established, creating the illusion of a continuous area. Two iso-parametric elements are considered, one flat and the other slightly curved (Figure 3). The relative error under consideration is determined as the maximum value obtained from these two reference elements.



Figure 3: The two reference boundary elements

The numerical experimentation strategy involves evaluating the boundary integrals at each point of the dense grid twice. Initially, the first integral, I_N , is computed using a prescribed Gauss order N . Subsequently, a reference integral, I_{100} , is calculated with a fixed 100 Gauss points. Next, for each point P , the relative error is determined as $\epsilon(P) \approx \left| \frac{I_N(P) - I_{100}(P)}{I_{100}(P)} \right|$. Once the distribution of the relative errors is established, an iso-error curve, or an error contour curve (as named in [4]), is plotted for a target maximum relative error, ϵ_0 . From this curve, the minimum allowable ratio $\lambda_0 = \frac{r_0}{L}$ is derived, ensuring a relative error $\epsilon < \epsilon_0$ for $\lambda = \frac{r}{L} > \lambda_0$.

By repeating the steps described above for various Gauss orders, a curve $N = f(\lambda)$ is constructed. By utilizing curve fitting techniques, an empirical formula is derived to express the number of Gauss points N as a function of the λ ratio. This process is conducted for various orders of singularity, including $\{\log(\frac{1}{r}), \frac{1}{r}, \frac{1}{r^2}\}$, and for three target maximal relative errors $\epsilon_0 = 10^{-2}, 10^{-3}, 10^{-4}$.

Numerical tests are conducted on two cases. The first case represents the lower limit, where errors are underestimated by assuming simplified integrands that only include terms responsible for the singularity. In contrast, the second case provides a realistic estimation of the error by considering the complete expressions of the kernels, shape functions, and the Jacobian, as outlined in (13).

3.1.1 Case one: Simplified integrands

The following integrals are considered:

$$\begin{cases} I_1 = \int_{-1}^1 \log\left(\frac{1}{r(\xi)}\right) d\xi & \Rightarrow O\left(\log\left(\frac{1}{r}\right)\right), \\ I_2 = \int_{-1}^1 \frac{1}{r(\xi)} d\xi & \Rightarrow O\left(\frac{1}{r}\right), \\ I_3 = \int_{-1}^1 \frac{1}{r(\xi)^2} d\xi & \Rightarrow O\left(\frac{1}{r^2}\right). \end{cases}$$

For each integral, the procedure detailed in section 3.1 is executed.

The resulting iso-error curves exhibit a closed form characterized by multiple lobes, akin to those observed in the work [10]. These distinctive curves, commonly referred to as butterfly curves, are depicted in Figures 4 and 5. The dataset in Table 1 defines, for the lower limit case, the minimum allowable ratios λ_0 for quadrature orders ranging from 2 to 35. These results are computed for singularity orders of $\{\log(\frac{1}{r}), \frac{1}{r}, \frac{1}{r^2}\}$, and precisions $\epsilon_0 = 10^{-2}, 10^{-3}, 10^{-4}$.

Table 1 shows that the minimum allowable ratio λ_0 decreases with increasing Gauss order N , as fewer Gauss points lead to higher errors, requiring greater distances from the boundary to stay below the error threshold. This effect intensifies with stronger singularities, especially for $O(\frac{1}{r^2})$, and as the error threshold ϵ_0 becomes stricter, with the highest λ_0 observed at $\epsilon_0 = 10^{-4}$.

3.1.2 Case two: Full form integrands

In this case the full form of integrals given in (13) is considered:

$$I = \int_{-1}^1 F(P_0, P(\xi)) \cdot \psi_m(\xi) \cdot J(\xi) d\xi,$$

where $F(P_0, P(\xi))$ is one of the fundamental solutions (kernels) $\{U_{ij}, T_{ij}, D_{ijk}, S_{ijk}, W_{ijk}, V_{ijk}\}$

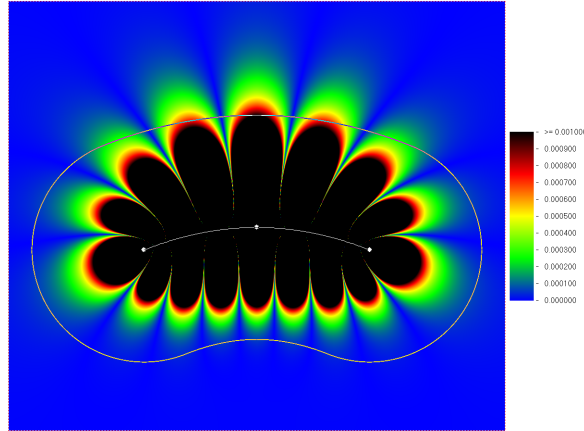


Figure 4: Relative error distribution around the curved reference element, for integral I_2 with 4 Gauss points

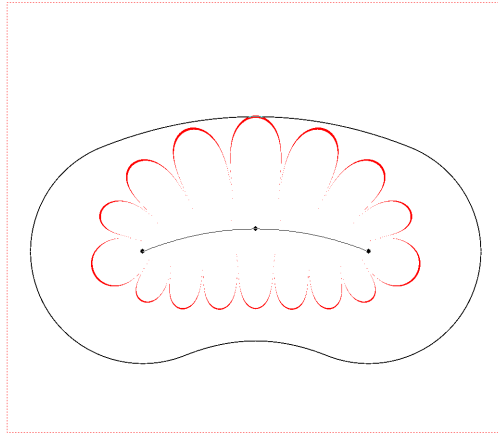


Figure 5: Iso-error curve and envelope, corresponding to integral I_2 with 4 Gauss points and a relative error 10^{-3}

$$\begin{cases} F(P_0, P(\xi)) = U_{ij} & \Rightarrow O\left(\log\left(\frac{1}{r}\right)\right), \\ F(P_0, P(\xi)) = T_{ij}, D_{ijk}, W_{ijk} & \Rightarrow O\left(\frac{1}{r}\right), \\ F(P_0, P(\xi)) = S_{ijk}, V_{ijk} & \Rightarrow O\left(\frac{1}{r^2}\right). \end{cases}$$

The relative error is computed for each function by varying the indices $i = 1, 2; j = 1, 2; k = 1, 2; m = 1, 2, 3$ to encompass all possible combinations, from which the maximum error is identified. Subsequently, the maximum

Table 1: The numerical results obtained for the lower limit case

| Precision | $\epsilon_0 = 10^{-2}$ | | | $\epsilon_0 = 10^{-3}$ | | | $\epsilon_0 = 10^{-4}$ | | |
|-----------|------------------------|------------------------|--------------------|------------------------|------------------|--------------------|------------------------|------------------|--------------------|
| | N | $O(\log(\frac{1}{r}))$ | $O(\frac{1}{r^2})$ | $O(\log(\frac{1}{r}))$ | $O(\frac{1}{r})$ | $O(\frac{1}{r^2})$ | $O(\log(\frac{1}{r}))$ | $O(\frac{1}{r})$ | $O(\frac{1}{r^2})$ |
| 2 | 1.125718 | 0.732523 | 1.053462 | 1.722896 | 1.69977 | 2.476304 | 3.119036 | 4.06014 | 6.231677 |
| 3 | 0.378903 | 0.39962 | 0.554979 | 1.047436 | 0.749395 | 1.00197 | 1.396343 | 1.318512 | 1.759226 |
| 4 | 0.223531 | 0.274538 | 0.378153 | 0.472037 | 0.483714 | 0.628713 | 1.010303 | 0.76963 | 0.981932 |
| 5 | 0.154594 | 0.21098 | 0.290135 | 0.310726 | 0.357336 | 0.460149 | 0.535374 | 0.540789 | 0.676164 |
| 6 | 0.11511 | 0.169262 | 0.235735 | 0.234901 | 0.28358 | 0.364319 | 0.381201 | 0.419096 | 0.517973 |
| 7 | 0.093778 | 0.142997 | 0.19878 | 0.188934 | 0.235656 | 0.301954 | 0.300095 | 0.342807 | 0.422116 |
| 8 | 0.076123 | 0.122481 | 0.171704 | 0.156941 | 0.201674 | 0.258723 | 0.248345 | 0.290531 | 0.356786 |
| 9 | 0.065062 | 0.107721 | 0.151201 | 0.133973 | 0.175814 | 0.225891 | 0.211074 | 0.251334 | 0.307793 |
| 10 | 0.055238 | 0.09541 | 0.134793 | 0.115931 | 0.156121 | 0.200444 | 0.183201 | 0.221721 | 0.272248 |
| 11 | 0.048691 | 0.086535 | 0.121919 | 0.103183 | 0.140165 | 0.180581 | 0.161941 | 0.198604 | 0.24436 |
| 12 | 0.042454 | 0.077927 | 0.111508 | 0.091398 | 0.126997 | 0.164631 | 0.145019 | 0.179886 | 0.220847 |
| 13 | 0.038029 | 0.071969 | 0.102493 | 0.083067 | 0.116594 | 0.15077 | 0.131184 | 0.164631 | 0.201681 |
| 14 | 0.033912 | 0.065875 | 0.094861 | 0.074748 | 0.107349 | 0.138989 | 0.119406 | 0.15146 | 0.186122 |
| 15 | 0.03105 | 0.061564 | 0.088616 | 0.068554 | 0.099716 | 0.129108 | 0.11012 | 0.140369 | 0.172947 |
| 16 | 0.027884 | 0.057352 | 0.082592 | 0.063127 | 0.092762 | 0.12079 | 0.101328 | 0.130294 | 0.161163 |
| 17 | 0.025693 | 0.054003 | 0.077653 | 0.058508 | 0.086813 | 0.11317 | 0.09407 | 0.122375 | 0.150767 |
| 18 | 0.023196 | 0.050654 | 0.07316 | 0.05407 | 0.081657 | 0.10693 | 0.087638 | 0.114613 | 0.141575 |
| 19 | 0.021674 | 0.047976 | 0.069224 | 0.050654 | 0.077048 | 0.100757 | 0.082193 | 0.107954 | 0.133874 |
| 20 | 0.019848 | 0.045297 | 0.065766 | 0.047306 | 0.073029 | 0.095438 | 0.076952 | 0.101978 | 0.126629 |
| 21 | 0.018508 | 0.043105 | 0.062709 | 0.044644 | 0.069224 | 0.090771 | 0.07297 | 0.096695 | 0.120459 |
| 22 | 0.017054 | 0.040698 | 0.05969 | 0.042248 | 0.065891 | 0.08681 | 0.068605 | 0.092073 | 0.114522 |
| 23 | 0.015891 | 0.039147 | 0.056977 | 0.039922 | 0.062791 | 0.082871 | 0.065434 | 0.08811 | 0.109204 |
| 24 | 0.015116 | 0.037209 | 0.054651 | 0.037597 | 0.060078 | 0.079265 | 0.061669 | 0.08415 | 0.104181 |
| 25 | 0.013953 | 0.035659 | 0.052713 | 0.036047 | 0.057364 | 0.076128 | 0.059154 | 0.080194 | 0.099794 |
| 26 | 0.013178 | 0.034109 | 0.050388 | 0.034109 | 0.055201 | 0.072991 | 0.05637 | 0.076993 | 0.096029 |
| 27 | 0.012403 | 0.032946 | 0.048837 | 0.032558 | 0.053101 | 0.070205 | 0.053876 | 0.074054 | 0.092266 |
| 28 | 0.011598 | 0.031701 | 0.046796 | 0.030928 | 0.051179 | 0.067698 | 0.051619 | 0.071368 | 0.089129 |
| 29 | 0.011082 | 0.03067 | 0.045249 | 0.029639 | 0.049374 | 0.065206 | 0.049548 | 0.068712 | 0.085678 |
| 30 | 0.010567 | 0.029381 | 0.043702 | 0.028351 | 0.047569 | 0.062952 | 0.047484 | 0.066279 | 0.082934 |
| 31 | 0.010052 | 0.028608 | 0.042413 | 0.02732 | 0.046022 | 0.060976 | 0.045935 | 0.064 | 0.080128 |
| 32 | 0.009278 | 0.027577 | 0.041123 | 0.026031 | 0.044475 | 0.059171 | 0.044072 | 0.061952 | 0.077605 |
| 33 | 0.009021 | 0.026546 | 0.039834 | 0.025258 | 0.043186 | 0.057366 | 0.042526 | 0.060129 | 0.075173 |
| 34 | 0.008505 | 0.025773 | 0.038545 | 0.024227 | 0.041639 | 0.055562 | 0.040979 | 0.058065 | 0.072906 |
| 35 | 0.008247 | 0.025 | 0.037514 | 0.023196 | 0.040608 | 0.054015 | 0.039691 | 0.056516 | 0.070641 |

error value is selected among functions sharing the same singularity order, as follows:

$$\begin{cases} \epsilon_{\max} \left(O \left(\log \left(\frac{1}{r} \right) \right) \right) &= \epsilon_{\max} (U_{ij}), \\ \epsilon_{\max} \left(O \left(\frac{1}{r} \right) \right) &= \max (\epsilon_{\max} (T_{ij}), \epsilon_{\max} (D_{ijk}), \epsilon_{\max} (W_{ijk})), \\ \epsilon_{\max} \left(O \left(\frac{1}{r^2} \right) \right) &= \max (\epsilon_{\max} (S_{ijk}), \epsilon_{\max} (V_{ijk})), \end{cases}$$

In this case, the iso-error curves obtained do not exhibit closed contours akin to those described in section 3.1.1. Rather, they feature sharp filaments that become thinner as they extend further from the element (Figures 6–7). This characteristic is particularly noticeable when lower Gaussian or-

ders are employed. Generating envelopes for the iso-error curves is not as straightforward as in the previous case. Therefore, the adopted solution is to solely consider the points located within denser regions of the iso-error curve. Specifically, a point is considered valid only if it has at least three adjacent points that have an error greater than or equal to the target error. This automated approach yields satisfactory results, as depicted in Figures 6 and 7. Another notable observation is that numerical testing with fewer than 4 Gauss points results in significantly large errors and exceedingly high λ_0 ratios. Consequently, quadrature orders of 2 and 3 cannot be considered in this case.

The dataset in Table 2 defines, for the upper limit case, the minimum allowable ratios λ_0 for quadrature orders ranging from 4 to 35. These results are computed for singularity orders of $\{\log(\frac{1}{r}), \frac{1}{r}, \frac{1}{r^2}\}$, and precisions $\epsilon_0 = 10^{-2}, 10^{-3}, 10^{-4}$.

The same patterns observed in Table 1 regarding the variation of λ_0 with respect to Gauss order N , singularity order, and the imposed error threshold are also evident in Table 2. Notably, the λ_0 values in Table 2 are significantly higher than their counterparts in Table 1, which can be attributed to the consideration of the full form of the integrals.

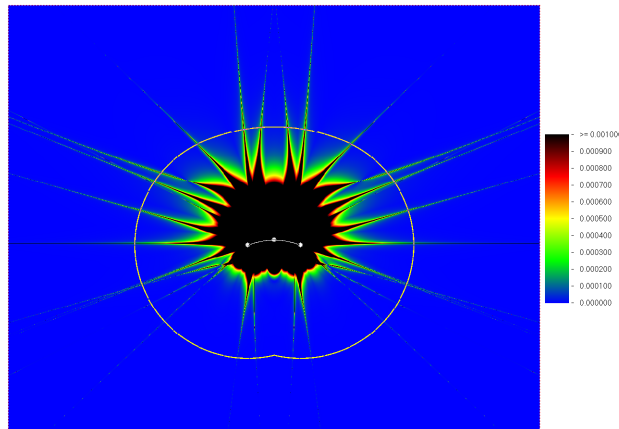
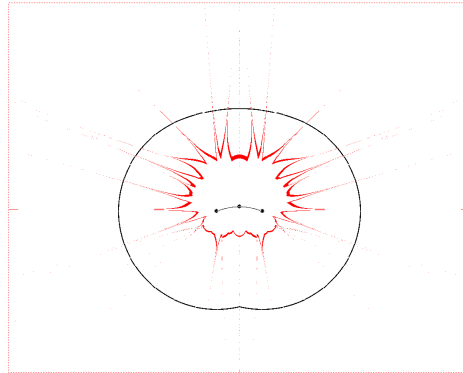


Figure 6: Relative error distribution around the reference curved element, for integral with T_{ij} kernel with 5 Gauss points

Table 2: The numerical results obtained for the upper limit case

| Precision | $\epsilon_0 = 10^{-2}$ | | | $\epsilon_0 = 10^{-3}$ | | | $\epsilon_0 = 10^{-4}$ | | |
|-----------|-------------------------------|-------------------------|---------------------------|-------------------------------|-------------------------|---------------------------|-------------------------------|-------------------------|---------------------------|
| | $O(\log(\frac{1}{\epsilon}))$ | $O(\frac{1}{\epsilon})$ | $O(\frac{1}{\epsilon^2})$ | $O(\log(\frac{1}{\epsilon}))$ | $O(\frac{1}{\epsilon})$ | $O(\frac{1}{\epsilon^2})$ | $O(\log(\frac{1}{\epsilon}))$ | $O(\frac{1}{\epsilon})$ | $O(\frac{1}{\epsilon^2})$ |
| 4 | 1.731452 | 3.466213 | 4.605432 | 2.26537 | 5.621703 | 9.956321 | | | |
| 5 | 0.990553 | 1.789411 | 2.234695 | 1.549063 | 2.258132 | 2.916322 | 1.916653 | 4.349683 | 5.034239 |
| 6 | 0.641329 | 1.210843 | 1.446829 | 0.847523 | 1.485321 | 1.71883 | 1.291281 | 2.211769 | 2.869445 |
| 7 | 0.506101 | 0.869918 | 0.948868 | 0.644162 | 1.147801 | 1.28719 | 0.938622 | 1.470089 | 1.717305 |
| 8 | 0.40158 | 0.594745 | 0.703197 | 0.483972 | 0.798056 | 0.936451 | 0.659695 | 1.09948 | 1.179583 |
| 9 | 0.339498 | 0.509217 | 0.621206 | 0.418563 | 0.610964 | 0.764019 | 0.567532 | 0.786614 | 0.98676 |
| 10 | 0.287768 | 0.441216 | 0.576349 | 0.346524 | 0.529581 | 0.670176 | 0.463009 | 0.662586 | 0.795725 |
| 11 | 0.24508 | 0.392913 | 0.524175 | 0.307237 | 0.458307 | 0.577628 | 0.411636 | 0.551755 | 0.6866 |
| 12 | 0.217618 | 0.361741 | 0.44923 | 0.265209 | 0.419566 | 0.533782 | 0.363731 | 0.494877 | 0.613864 |
| 13 | 0.184143 | 0.329125 | 0.404135 | 0.232998 | 0.384247 | 0.462339 | 0.316576 | 0.437666 | 0.561505 |
| 14 | 0.172671 | 0.302964 | 0.371398 | 0.219531 | 0.360468 | 0.438523 | 0.288139 | 0.40124 | 0.527596 |
| 15 | 0.153868 | 0.296416 | 0.349031 | 0.201693 | 0.337231 | 0.401847 | 0.255058 | 0.36921 | 0.451162 |
| 16 | 0.14172 | 0.258965 | 0.319079 | 0.183379 | 0.308434 | 0.361102 | 0.228336 | 0.342827 | 0.40318 |
| 17 | 0.13214 | 0.22594 | 0.309555 | 0.168858 | 0.295729 | 0.343361 | 0.211628 | 0.31895 | 0.365151 |
| 18 | 0.127365 | 0.209414 | 0.276741 | 0.163691 | 0.261573 | 0.323737 | 0.198633 | 0.307648 | 0.346978 |
| 19 | 0.119226 | 0.197478 | 0.261351 | 0.15 | 0.244915 | 0.315091 | 0.187495 | 0.300695 | 0.334115 |
| 20 | 0.111972 | 0.188242 | 0.232066 | 0.142254 | 0.230969 | 0.30471 | 0.178216 | 0.272005 | 0.323738 |
| 21 | 0.108583 | 0.178362 | 0.211824 | 0.136444 | 0.216996 | 0.273097 | 0.168004 | 0.243959 | 0.315953 |
| 22 | 0.102817 | 0.169428 | 0.202378 | 0.130282 | 0.199078 | 0.262495 | 0.160862 | 0.237681 | 0.31163 |
| 23 | 0.098592 | 0.166204 | 0.198193 | 0.125306 | 0.189353 | 0.226737 | 0.155008 | 0.225242 | 0.271844 |
| 24 | 0.09507 | 0.165334 | 0.186731 | 0.119718 | 0.185464 | 0.216276 | 0.147583 | 0.222462 | 0.245897 |
| 25 | 0.091549 | 0.164062 | 0.175876 | 0.115096 | 0.170425 | 0.208829 | 0.142943 | 0.20568 | 0.233789 |
| 26 | 0.088028 | 0.163364 | 0.172232 | 0.111383 | 0.162875 | 0.191682 | 0.136445 | 0.185374 | 0.221683 |
| 27 | 0.085211 | 0.145735 | 0.165063 | 0.107042 | 0.158314 | 0.186731 | 0.130875 | 0.184582 | 0.213032 |
| 28 | 0.082394 | 0.136129 | 0.158037 | 0.103957 | 0.154121 | 0.181879 | 0.128091 | 0.172242 | 0.202653 |
| 29 | 0.079577 | 0.134008 | 0.155645 | 0.100245 | 0.14795 | 0.177025 | 0.123084 | 0.167684 | 0.194871 |
| 30 | 0.077465 | 0.132594 | 0.151986 | 0.097183 | 0.144613 | 0.167816 | 0.117881 | 0.163207 | 0.188816 |
| 31 | 0.075184 | 0.130938 | 0.144066 | 0.093747 | 0.14207 | 0.166256 | 0.115096 | 0.158548 | 0.180169 |
| 32 | 0.072535 | 0.129645 | 0.14263 | 0.091549 | 0.138743 | 0.16298 | 0.111384 | 0.154713 | 0.174119 |
| 33 | 0.070423 | 0.128353 | 0.134518 | 0.089106 | 0.136108 | 0.161409 | 0.107671 | 0.15079 | 0.16897 |
| 34 | 0.069014 | 0.127706 | 0.131838 | 0.086322 | 0.134169 | 0.151348 | 0.103958 | 0.146912 | 0.164346 |
| 35 | 0.066901 | 0.124931 | 0.128298 | 0.083803 | 0.13223 | 0.149121 | 0.101984 | 0.145145 | 0.159715 |

Figure 7: Iso-error curve and envelope, corresponding to integral with T_{ij} kernel with 5 Gauss points and a target relative error 10^{-3}

3.2 Determination of the empirical formulas for the two limit cases

Tables 1 and 2 present the numerically obtained results for the lower and upper limit cases, respectively. These data are then translated into two empirical formulas aimed at determining the Gauss order N necessary to attain a desired precision (10^{-2} , 10^{-3} and 10^{-4}) for a given λ ratio. The empirical formulas are deduced using curve-fitting techniques, assuming an exponential decay profile that closely mirrors the relationship between N and the λ ratio.

3.2.1 The lower limit curve

Based on the numerical results in Table 1, Figure 8 presents the variation curves of the required Gauss order N versus the minimum allowable ratio λ_0 for three singularity orders: $O\left(\log\left(\frac{1}{r}\right)\right)$, $O\left(\frac{1}{r}\right)$ and $O\left(\frac{1}{r^2}\right)$, with a target error of $\epsilon_0 = 10^{-3}$.

A closer examination of the three graphs shown in Figure 8 reveals a clear exponential decay pattern in the relationship between the Gauss order N and the ratio λ_0 . Specifically, the curves decrease from infinity and approach a horizontal asymptote at $N = 2$. This behavior is consistently observed for the other error thresholds as well, namely $\epsilon_0 = 10^{-2}$ and $\epsilon_0 = 10^{-4}$.

Consequently, we propose the following formula:

$$N_l(\lambda) = 2 + A_l \times \lambda^{B_l}, \quad (14)$$

where A_l and B_l are constants, with $A_l > 0$, $B_l < 0$ and $\lambda = \frac{r}{L}$.

N.B. The subscript “l” stands for **L**ower limit curve.

$$\begin{cases} \lim_{\lambda \rightarrow 0} N_l(\lambda) = +\infty, \\ \lim_{\lambda \rightarrow +\infty} N_l(\lambda) = 2. \end{cases} \quad (15)$$

The coefficients A_l and B_l are derived through exponential curve fitting. The

results are outlined in Table 3. Figure 9 displays the variation curves of the required Gauss order N needed to achieve a target precision of $\epsilon_0 = 10^{-3}$ for a singularity of order $O\left(\frac{1}{r}\right)$, based on both the empirical formula in (14) and the numerical results from Table 1.

A closer examination of the two graphs in Figure 9 reveals strong agreement between the numerical data and its empirical representation by 14. This consistency also holds for the other singularity orders $\{O\left(\log\left(\frac{1}{r}\right)\right), O\left(\frac{1}{r^2}\right)\}$ and error thresholds $\{10^{-2}, 10^{-4}\}$.

Table 3: A_l and B_l coefficients for the lower limit curve

| | $\epsilon_0 = 10^{-2}$ | | $\epsilon_0 = 10^{-3}$ | | $\epsilon_0 = 10^{-4}$ | |
|--|------------------------|---------|------------------------|---------|------------------------|---------|
| | A_l | B_l | A_l | B_l | A_l | B_l |
| $O\left(\log\left(\frac{1}{r}\right)\right)$ | 0.9923 | -0.7332 | 1.226 | -0.8785 | 1.693 | -0.9214 |
| $O\left(\frac{1}{r}\right)$ | 0.6712 | -1.059 | 1.133 | -1.054 | 1.695 | -1.034 |
| $O\left(\frac{1}{r^2}\right)$ | 0.8708 | -1.11 | 1.419 | -1.08 | 2.052 | -1.049 |

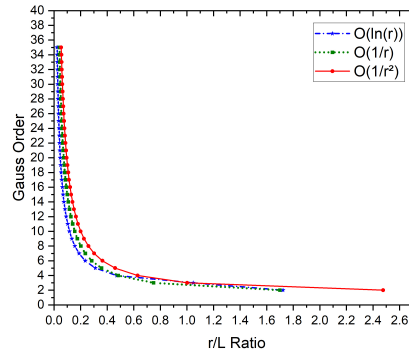


Figure 8: Numerical results of variation of N as function of λ ratio for lower limit case with a precision 10^{-3}

3.2.2 The upper limit curve

Similar to Figure 8, Figure 10 illustrates the variation of N with respect to the minimum allowable ratio λ_0 , based on the numerical results of the full-form

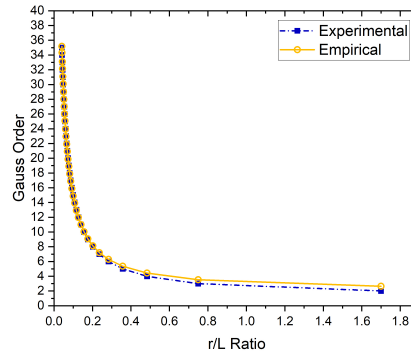


Figure 9: Comparison between empirical formula of (14) and numerical results of the lower limit case with singularity $O\left(\frac{1}{r}\right)$ and precision 10^{-3}

integrands, as recorded in Table 2, for a target error bound of $\epsilon_0 = 10^{-3}$.

An analysis of the curves in Figure 10 reveals a similar exponential decay pattern; however, in this case, the horizontal asymptote is located at $N = 4$ instead of $N = 2$.

Accordingly, we propose another empirical formula analogous to that previously introduced in (14).

$$N_u(\lambda) = 4 + A_u \times \lambda^{B_u}, \quad (16)$$

where A_u and B_u are constants, with $A_u > 0, B_u < 0$ and $\lambda = \frac{r}{L}$.

N.B. The subscript “u” stands for Upper limit curve.

$$\begin{cases} \lim_{\lambda \rightarrow 0} N_u(\lambda) = +\infty, \\ \lim_{\lambda \rightarrow +\infty} N_u(\lambda) = 4. \end{cases} \quad (17)$$

Similar to section 3.2.1, the coefficients A_u and B_u are derived through exponential curve fitting techniques. The results are outlined in Table 4 To validate the consistency of the empirical formula in (16) with the numerical results in Table 2, Figure 11 illustrates the variation of the Gauss order N as a function of λ , required to achieve a target accuracy of $\epsilon_0 = 10^{-3}$ for a singularity of order $O\left(\frac{1}{r}\right)$. The figure compares the predictions of the empirical

model with the corresponding numerical data.

A detailed comparison of the two curves in Figure 11 reveals excellent agreement between the empirical and numerical results. This consistency also extends to rest cases involving different singularity orders $\{O(\log(\frac{1}{r})), O(\frac{1}{r^2})\}$ and alternative error thresholds, including $\{10^{-2}, 10^{-4}\}$.

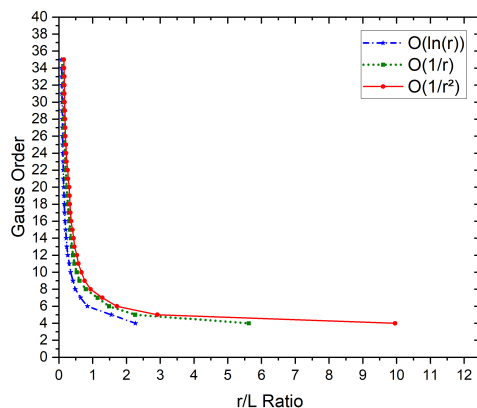


Figure 10: Experimental upper limit curves of variation of N as function of λ ratio in case of precision 10^{-3}

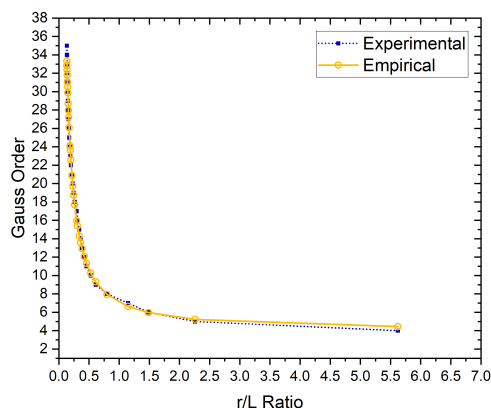


Figure 11: Comparison between empirical formula and experimental results for the case of the upper limit curve $O(\frac{1}{r})$ and precision 10^{-3}

Table 4: A_u and B_u coefficients for the upper limit curve

| | $\epsilon_0 = 10^{-2}$ | | $\epsilon_0 = 10^{-3}$ | | $\epsilon_0 = 10^{-4}$ | |
|--|------------------------|--------|------------------------|--------|------------------------|---------|
| | A_u | B_u | A_u | B_u | A_u | B_u |
| $O\left(\log\left(\frac{1}{r}\right)\right)$ | 1.169 | -1.21 | 1.596 | -1.195 | 2.376 | -1.12 |
| $O\left(\frac{1}{r}\right)$ | 2.002 | -1.283 | 3.065 | -1.116 | 3.953 | -1.048 |
| $O\left(\frac{1}{r^2}\right)$ | 3.323 | -1.082 | 4.2 | -1.038 | 5.16 | -0.9721 |

3.3 Determination of the generalized empirical formula of the proposed integration criterion

While the lower limit formula of (14) offers improved performance by reducing computational time, it tends to yield less accurate results. Conversely, the upper limit formula of (16) achieves high accuracy but at the expense of significantly slower performance. Therefore, an optimal approach requires striking a balance between performance and accuracy. This is accomplished by introducing a formula that describes a curve positioned between the two defined extremes, leaning toward one limit or the other depending on the structure type or the method employed for near-singular treatment. This approach can be particularly useful in the case of variable transformation techniques and semi-analytical algorithms that reduce or cancel the boundary layer effect, where unnecessary computational overhead can be avoided to take full advantage of these techniques.

For instance, in the case of a thin structure analyzed solely through the element subdivision method with a straightforward application of the Gaussian integration technique, a curve closer to the upper limit is preferable. However, when employing a nonlinear transformation technique to address the same problem, the curve can be shifted toward the lower limit, thereby conserving computational resources. A parameter $\alpha \in [0, 1]$ is introduced to control the curve position, such that $\alpha = 0$ corresponds to the lower curve and $\alpha = 1$ to the upper curve. Any value of α between 0 and 1 produces an intermediate curve.

Equations (14) and (16), are written in the new following form:

$$\begin{cases} N_l(\lambda) = N_{0_l} + A_l \times \lambda^{B_l}, \\ N_u(\lambda) = N_{0_u} + A_u \times \lambda^{B_u}, \end{cases} \quad (18)$$

where $N_{0_l} = 2, N_{0_u} = 4, (A_l, B_l)$ are given in Table 3 and (A_u, B_u) is given in Table 4.

To maintain the same exponential profile observed in the lower and upper limit curves, we propose a formula similar to (14) and (16).

$$N(\lambda, \alpha) = N_0(\alpha) + A(\alpha) \times \lambda^{B(\alpha)}, \quad (19)$$

where $N_0(\alpha), A(\alpha)$ and $B(\alpha)$ are given as functions of the parameter α deduced by linear combinations of $(N_{0_l}, N_{0_u}), (A_l, A_u)$ and (B_l, B_u) , respectively.

Let $\{\Delta N_0 = N_{0_u} - N_{0_l} = 2\}, \{\Delta A = A_u - A_l\}$ and $\{\Delta B = B_u - B_l\}$.

The linear combinations will have the following form:

$$\begin{cases} N_0(\alpha) = N_{0_l} + \alpha \times \Delta N_0 = 2.(\alpha + 1), \\ A(\alpha) = A_l + \alpha \times \Delta A, \\ B(\alpha) = B_l + \alpha \times \Delta B. \end{cases} \quad (20)$$

Thus, (19) becomes

$$N(\lambda, \alpha) = 2.(\alpha + 1) + A(\alpha) \times \lambda^{B(\alpha)} \quad (21)$$

with the functions $A(\alpha)$ and $B(\alpha)$ given in Table 5.

Table 5: $A(\alpha)$ and $B(\alpha)$ functions for our final empirical formula

| | $\epsilon_0 = 10^{-2}$ | | $\epsilon_0 = 10^{-3}$ | | $\epsilon_0 = 10^{-4}$ | |
|-------------------------------|-------------------------|--------------------------|------------------------|--------------------------|------------------------|--------------------------|
| | $A(\alpha)$ | $B(\alpha)$ | $A(\alpha)$ | $B(\alpha)$ | $A(\alpha)$ | $B(\alpha)$ |
| $O(\log(\frac{1}{\epsilon}))$ | $0.9923 + 0.1767\alpha$ | $-0.7332 - 0.4768\alpha$ | $1.226 + 0.37\alpha$ | $-0.8785 - 0.3165\alpha$ | $1.693 + 0.683\alpha$ | $-0.9214 - 0.1986\alpha$ |
| $O(\frac{1}{\epsilon})$ | $0.6712 + 1.3308\alpha$ | $-1.059 - 0.224\alpha$ | $1.133 + 1.932\alpha$ | $-1.054 - 0.062\alpha$ | $1.695 + 2.258\alpha$ | $-1.034 - 0.014\alpha$ |
| $O(\frac{1}{\epsilon^2})$ | $0.8708 + 2.4522\alpha$ | $-1.11 + 0.028\alpha$ | $1.419 + 2.781\alpha$ | $-1.08 + 0.042\alpha$ | $2.052 + 3.108\alpha$ | $-1.049 + 0.0769\alpha$ |

We propose a new integration criterion for the 2D elasticity BEM, defined by the empirical formula in (21). The corresponding linear functions $A(\alpha)$ and $B(\alpha)$ are listed in Table 5. The parameter α is application-dependent

and may vary according to factors such as the thinness of the structure or the use of near-singular treatment techniques. Figure 12 presents a series of curves generated with the proposed empirical formula for a singularity of order $O(\frac{1}{r})$, with a target accuracy of 10^{-3} . These curves lie consistently between the two limiting curves and display a uniform exponential behavior. The proposed empirical formula of (21), like existing ones, follows an exponential form that naturally leads to very high Gauss quadrature orders for small λ ratios. This behavior is consistent with the formulations in (15) and (17). However, excessively high quadrature orders may introduce floating-point round-off errors. Moreover, computing or retrieving the corresponding Gauss–Legendre nodes and weights at runtime can create significant performance bottlenecks.

To overcome this issue, most BEM software relies on precomputed tables of Gauss–Legendre nodes and weights. Consequently, when (21) prescribes a Gauss order exceeding the available predefined values, the element must be subdivided. To address this, we provide an additional formula to determine the required number of subelements:

$$M = \frac{1}{\lambda} \times \left(\frac{N_{avl} - 2(1 + \alpha)}{A(\alpha)} \right)^{\frac{1}{B(\alpha)}} + 1, \quad (22)$$

where M is the number of subelements and N_{avl} is the maximum Gauss order available in the program.

3.4 Guidelines on the choice of the α parameter

An optimal selection of the parameter α would require a comprehensive optimization study that considers multiple factors, including the geometry of the structure, the value of the ratio λ , and the type of variable transformation technique if any is employed. Such an in-depth investigation is not addressed in the current manuscript.

Nevertheless, preliminary guidance for choosing α can be provided based on practical experience and supported by the results from the validation example presented in Section 4. Table 6 summarizes these guidelines with respect to

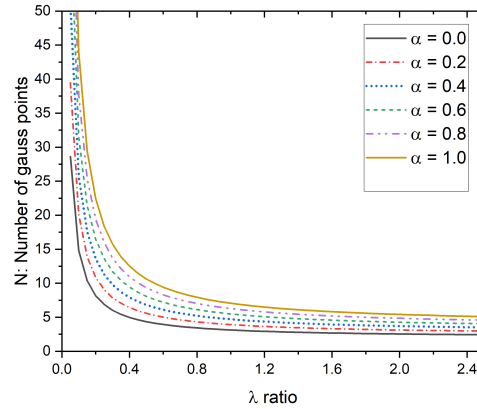


Figure 12: Limit and intermediate curves of the new criterion, singularity order $O(\frac{1}{r})$ and precision 10^{-3}

structural thinness and the presence or absence of near-singularity treatment techniques.

Table 6: Guidelines on the choice of the α parameter

| | Nonthin structures | Moderately thin structures | Extremely thin structures |
|-----------------------------------|---------------------------|-----------------------------------|----------------------------------|
| Without semi-analytical treatment | $\alpha = 0$ to 0.3 | $\alpha = 0.3$ to 0.7 | $\alpha = 0.7$ to 1 |
| With semi-analytical treatment | N.A | $\alpha = 0$ to 0.3 | $\alpha = 0.3$ to 0.6 |

3.5 Numerical implementation

One of the main advantages of the empirical formula approach is its simplicity and ease of numerical implementation. The proposed expressions can be directly coded as a function that receives the relevant input arguments and returns the required Gauss order. This function is then invoked within the subroutine that performs the numerical integration. If needed, the subroutine can subdivide the boundary element into smaller subelements and re-evaluate the corresponding Gauss order for each by calling the same func-

tion.

The implementation details of the algorithm used to perform the BEM numerical integrations, corresponding to the general form of (13), are summarized as follows: **Step 0:** To evaluate the integral I over the boundary element E with

$$I = \int_{-1}^1 f(\xi) d\xi$$

such that

$$f(\xi) = F(P^*, P(\xi)) \psi(\xi) J(\xi).$$

Gather all the necessary input data: {Source point P^* , boundary element E (of length L), the singularity order of the integration kernel $F(P^*, P(\xi))$, the selected value of α , the required precision (10^{-2} , 10^{-3} or 10^{-4}) }.

Step 1: Evaluate r : the shortest distance between the source point P^* and the boundary element E .

Step 2: Calculate the λ ratio with $\lambda = \frac{r}{L}$.

Step 3: Calculate N : the number of Gauss points necessary to obtain the target precision by using (21), which returns a real number that is rounded to the nearest integer.

Step 4: Verify if the obtained Gauss order N is less than or equal N_{avl} , the maximum available order in the program ($N \leq N_{avl}$).

Step 5: IF TRUE, evaluate the integral using N Gauss points, with

$$I = \int_{-1}^1 f(\xi) d\xi \approx \sum_{i=1}^N f(\xi_i) w_i. \quad (23)$$

Return I and terminate the algorithm.

Step 6: ELSE, the element E has to be subdivided into M subelements, with M determined through (22).

Step 7: For each subelement E_k , $\{k = 1, 2, \dots, M\}$ calculate the necessary number of Gauss points by repeating the steps from **Step 0** to **Step 3**, then calculate the integral I_k by using the following formula [1]:

$$I_k = \int_{-1}^1 f(\eta) d\eta \approx \sum_{i=1}^N f(\bar{\eta}_i) w_i \quad (24)$$

with $\bar{\eta}_i = \frac{1}{M}(M - 2k + 1 + \eta_i)$.

Step 8: Calculate the global integral over the boundary element E with

$$I \approx \frac{1}{M} \sum_{k=1}^M I_k. \quad (25)$$

Return I and terminate the algorithm.

3.6 Comparison between the proposed criterion and the existing ones

In order to compare the existing criteria with the proposed one, the corresponding functions $N(\lambda)$ are plotted on a log-log graph (Figure 13). A closer examination of the curves in Figure 13 reveals the following observations:

The formula of Lachat and Watson suggests relatively low Gauss orders for large values of the λ ratio. However, the required order increases drastically for small λ values, tending to infinity as λ approaches $\frac{1}{4}$. This behavior is confirmed by the approximation formula proposed by Gao and Davies [11], which appears in (26)–(27)

$$N(\lambda, \epsilon_0) = \frac{p' \log\left(\frac{\epsilon_0}{2}\right)}{2 \log\left(\frac{1}{4\lambda}\right)}, \quad (26)$$

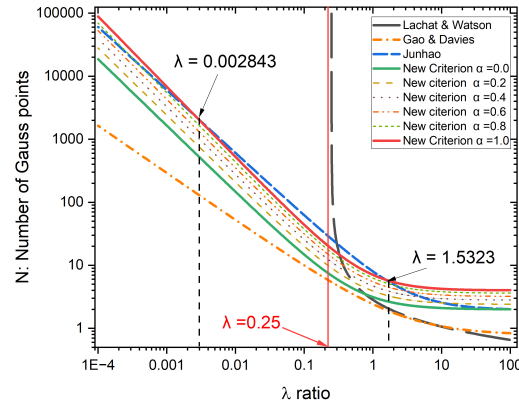
$$\lim_{\lambda \rightarrow \frac{1}{4}} N(\lambda, \epsilon_0) = \lim_{\lambda \rightarrow \frac{1}{4}} \frac{p' \log\left(\frac{\epsilon_0}{2}\right)}{2 \log\left(\frac{1}{4\lambda}\right)} = +\infty. \quad (27)$$

This divergence of the Gauss order at $\lambda \approx \frac{1}{4}$ makes the Lachat and Watson criterion very resource-consuming, introducing unnecessary computational overhead, especially in the case of thin structures. A closer look at the other curves shows that the Gao and Davies formula suggests the lowest N , making it less resource-consuming but at the expense of reduced accuracy, which will be demonstrated further in the validation example of Section 4. The newly proposed criterion, with its different values of the parameter α , appears in the log-log graph as a band of nearly parallel lines, particularly for $\lambda < 0.5$. The upper-limit curve ($\alpha = 1$) intersects with the Junhao curve at $\lambda = 0.002843$ and $\lambda = 1.5323$. Within the interval $\lambda \in [0.002843, 1.5323]$, the proposed criterion with $\alpha = 1$ suggests fewer Gauss points than the formula of [23]. For values of $\alpha < 1$, this interval becomes wider. From these observations, we note that the proposed formula produces fewer Gauss points at moderate λ values, which are predominant when element subdivision is applied, as in most BEM applications where only a limited number of Gaussian weights and abscissas are available in the program. Furthermore, it is worth noting that the numerical implementation of empirical criteria is simpler compared to

tabular criteria. In the tabular case, the programmer must check the value of the λ ratio and extract the corresponding Gauss order based on the interval in which it falls. In contrast, with the empirical approach, one only needs to pass the value of λ as an argument to the formula's function, which directly returns the appropriate Gauss order as an integer. Finally, Table 7 presents a comparative summary that situates the proposed criterion within the broader framework of existing methods while emphasizing their key distinctions.

Table 7: Recapitulative table for the comparison between the different criteria

| Criterion | Form | Mathematical error estimation | Numerical testing | Intended for | Further improvements | Disadvantages |
|-------------------------------------|---------------------|---|---|--|--|---|
| Lachat and Watson [25] | Empirical formulas | YES: Based on Stroud and Secrest [39] for $(\frac{1}{r^2})$ | NO | 3D structures, but usable in 2D | - Simplified by Mustoe [29] - Simplified by Gao and Davies [11] | - Less accurate in thin structures; - Diverges at $\lambda = \frac{1}{4}$, gives very high Gauss orders for small λ . |
| Jun and Beer [22] | Table format | YES: Based on Stroud and Secrest [39] for $(\frac{1}{r}, \frac{1}{r^2}, \frac{1}{r^3})$ | NO | Both 3D and 2D | | - Less accurate in thin structures; - Table format only. |
| Bu and Davies [4] | Empirical + Table | NO | YES: Testing on $(\frac{1}{r}, \frac{1}{r^2}, \frac{1}{r^3})$ | 3D structures, usable in 2D | Simplified by Gao and Davies [11] | Less accurate in thin structures. |
| Eberwien, Duencer, and Moser [10] | Table format | NO | YES: Testing on $f(\xi) = \frac{1}{r^p} \Psi(\xi) J(\xi)$, $p \in \{1, 2\}$ | 2D structures | | Table format only. |
| Junhao, Zhipeng, and Yongqiang [23] | Empirical formulas | NO | YES: Testing on $(\ln(\frac{1}{r}), \frac{1}{r}, \frac{1}{r^2})$ | 2D structures | | Computational gain only for very small λ (rare in element subdivision). |
| Zhou, Yang, and Chen [48] | Neural network (AI) | NO | YES: NN trained to propose Gauss order (depends on source point, geometry, kernel, ...) | 2D structures | | Needs a trained neural network. |
| Proposed criterion | Empirical formulas | NO | YES: Testing on full kernel functions of 2D elasticity BEM | Tailored for 2D elasticity (extendable to other 2D apps) | | Requires choice of control parameter α . |

Figure 13: Variation of N versus λ for the new criterion and existing ones

4 Validation example

Let us consider a long and very thin fin, characterized by a length $L = 10$ cm and a maximum height $H = 5$ mm at its left end. The fin is uniformly subjected to a normal load $P = -2$ bar on its upper surface and is fully constrained at its left end, as illustrated in Figure 14.

The fin is made of steel with $E = 200$ GPa and $\nu = 0.3$, and follows a plane stress state.

The classical problem of an infinite elastic wedge, solved by Timoshenko [41, 24], serves as the benchmark solution. This problem is modeled in a polar coordinate system (r, θ) , where $r = 0$ is located at the apex of the wedge. The angle θ ranges from 0 to β , with $\beta = 0.05$ rad being the wedge angle. The upper surface of the wedge is loaded at $\theta = 0$, as illustrated in Figure 15.

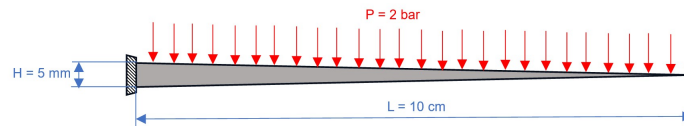


Figure 14: Very thin fin subjected to a bending effort

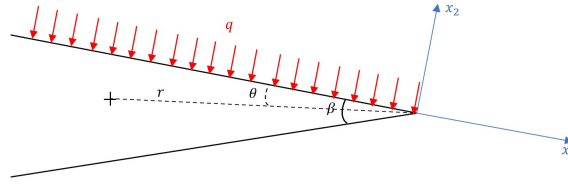


Figure 15: Model of an infinite elastic wedge subjected to a uniform pressure on its upper surface

According to [41, 24], the stress components depend only on θ and are given by

$$\begin{cases} \sigma_{rr}(\theta) = \frac{q}{k} \left(-k + \frac{1}{2} \tan \beta - \theta + \frac{1}{2} \tan \beta \cos 2\theta - \frac{1}{2} \sin 2\theta \right), \\ \sigma_{\theta\theta}(\theta) = \frac{q}{k} \left(-k + \frac{1}{2} \tan \beta - \theta - \frac{1}{2} \tan \beta \cos 2\theta + \frac{1}{2} \sin 2\theta \right), \\ \sigma_{r\theta}(\theta) = \frac{q}{2k} (1 - \tan \beta \sin 2\theta - \cos 2\theta), \end{cases} \quad (28)$$

where $k = \tan \beta - \beta$. The displacement field is obtained by integrating the strains, which are derived from the linear-elastic constitutive law (Hooke's law), while satisfying the compatibility equations. The resulting integration constants are determined by imposing displacement boundary conditions at $r = L$ (L being the length of the fin), as follows:

$$\begin{cases} u_r(L, 0) = u_r\left(L, \frac{\beta}{2}\right) = 0, \\ u_\theta\left(L, \frac{\beta}{2}\right) = 0. \end{cases} \quad (29)$$

The resulting displacement field is

$$\begin{cases} u_r(r, \theta) = \frac{q}{Ek} \left[(1 - \nu) \left(\frac{1}{2} \tan \beta - (k + \theta) \right) + (1 + \nu) \frac{\sin(\beta - 2\theta)}{2 \cos \beta} \right] r \\ \quad + C_1 \cos \theta + C_2 \sin \theta, \\ u_\theta(r, \theta) = \frac{q}{Ek} \left[2 \log \frac{r}{L} + \frac{1 + \nu}{2 \cos \beta} (1 - \cos(\beta - 2\theta)) \right] r \\ \quad + C_1 \left(\sin \left(\frac{\beta}{2} \right) \frac{r}{L} - \sin \theta \right) + C_2 \left(\cos \theta - \cos \left(\frac{\beta}{2} \right) \frac{r}{L} \right), \end{cases} \quad (30)$$

with the constants C_1 and C_2 given by

$$\begin{cases} C_1 = \frac{qL}{Ek} ((1-\nu)k - \tan \beta), \\ C_2 = \frac{qL}{Ek} \left(\frac{\tan \beta}{\tan \frac{\beta}{2}} + (1-\nu)k \frac{1 - 2 \cos \frac{\beta}{2}}{2 \sin \frac{\beta}{2}} \right). \end{cases} \quad (31)$$

To perform the computational tasks, a BEM code was developed, using the C++ programming language. The problem is then solved using this code, employing both existing integration criteria and the newly proposed criterion.

The analytically obtained results for the infinite wedge should align with the numerical results except at locations near the fixed base, where the resulting errors are not accounted for in the evaluation of the different criteria. More specifically, the obtained results confirmed that the analytical solution aligns well with the numerical results at $x > 3cm$.

The relative errors are evaluated by

$$\epsilon = \left| \frac{Res_{analytical} - Res_{BEM}}{Res_{analytical}} \right|. \quad (32)$$

The effectiveness of the proposed criterion is assessed across various values of α aiming for a target precision of 10^{-3} .

4.1 The displacement solution

To analyze the displacement solution, a series of 40 probe points are placed along the horizontal direction of the fin at $\theta = \frac{\beta}{2}$. The displacement magnitude $U = \sqrt{U_1^2 + U_2^2}$ is evaluated at each point within this set.

The curves presented in Figure 16 illustrate the variation of the displacement magnitude along the length of the fin. Whereas, Figure 17 illustrates the variation of the relative error versus the horizontal location. Upon analysis of the results depicted in Figure 16, a notable correspondence between the displacement magnitudes obtained from the analytical solution and the numerical (BEM) solutions when employing the criterion of Junhao and the new criterion with α values of 0.8 and 1 is observed. However, utilizing

the remaining existing criteria (Lachat and Watson, Jun and Beer, Gao and Davies and Eberwien) or the new criterion with low α values leads to fluctuating results, particularly near the fin tip, resulting in higher errors.

Additionally, Figure 17 illustrates how the relative error tends to escalate in thinner regions when employing either existing criteria or the new criterion with lower α values. However, when using the new integration criterion with α values close to 1, the error begins to decrease and remains relatively stable, even in extremely thin regions. Specifically, for $\alpha = 1$, the error decreased slightly below the target error upper bound of 10^{-3} . Moreover, although the Junhao criterion produced acceptable and good quality results, the resulting relative error remained above the required upper bound. In contrast, the new criterion yielded better results even with $\alpha = 0.8$, with a net gain in efficiency as stated previously in section 3.6 and illustrated in Figure 13, where the new criterion's curves are situated below Junhao's curve in the case of moderate values of the λ ratio. Furthermore, this favored interval grows wider as α is decreased. As, for $\alpha = 0.8$, this interval becomes $[0.000381564, 2.09121]$. Based on the preceding analysis, the new criterion demonstrated a high level of accuracy in displacement results, offering improved and more consistent precision, even in regions of extreme thinness. In contrast, the existing criteria were unable to satisfy the required upper error bound, largely due to the strong influence of boundary layer effect, especially in the thinner sections of the fin.

4.2 The stress solution

4.2.1 Stress assessment in the horizontal direction

In contrast to the displacements, which exhibit significant variations along the horizontal direction, the major stress variations occur along the vertical direction of the fin, as it is confirmed by the stress analytical solution of the infinite wedge problem, which varies only in terms of θ (see (28)). Consequently, the previously utilized probe points are not used for stress assessment. Given that stress reaches its maximum values at the boundary

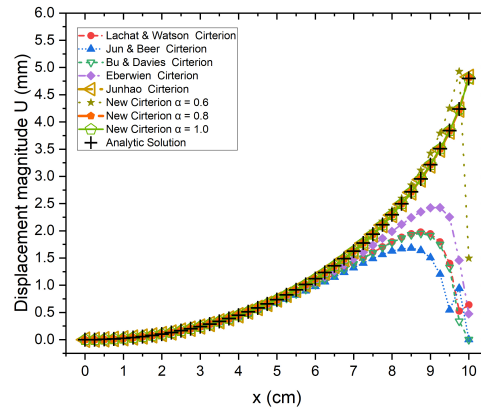


Figure 16: Variation of the displacement magnitude along the horizontal direction of the fin

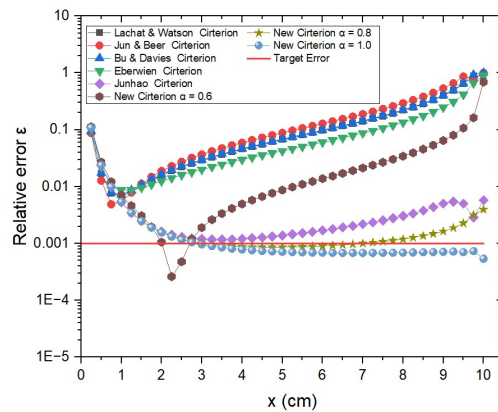


Figure 17: The variation of the relative error for displacement magnitude along the horizontal direction of the fin

of the fin, the initial analysis of the stress solution involves probing 40 points positioned on the top side of the fin.

The curves in Figure 18 illustrate the variation of von Mises equivalent stress along the upper surface of the fin. Whereas, Figure 19 depicts the variation of the corresponding relative error with respect to the horizontal location.

A detailed examination of the curves in Figure 18 reveals significant fluctuations and the presence of very important errors in stress solution, particularly

when utilizing existing criteria or the new criterion with relatively low values for the α coefficient. These fluctuations and errors increase drastically as the fin tip is approached. Notably, these errors are more pronounced compared to the displacement solution, attributable to the higher singularity order $O\left(\frac{1}{r^2}\right)$ of the kernels employed in the stress solution.

However, with the adoption of the Junhao criterion and the new integration criterion using α values of 0.8 and 1, satisfactory results are obtained, even in extremely thin sections very close to the fin tip, without necessitating any analytical or semi-analytical treatment.

Figure 19 depicts, akin to Figure 17, the significant increase of errors in thinner regions, with values reaching approximately a twenty (20) times higher than displacement errors in some cases. Nonetheless, the proposed criterion with $\alpha = 1$ yields acceptable errors, below the target upper bound of 10^{-3} . Moreover, although the Junhao criterion produced acceptable and good quality results, the resulting relative error remained above the required upper bound. While, the proposed criterion yielded better results even with $\alpha = 0.8$ and remain stable even in thinner sections, with a gain in efficiency, as seen in section 4.1.

From the results above, it is evident that the new criterion was capable of producing very accurate results in stress response, with improved and more stable precision in regions of extreme thinness, despite the augmented order of the near-singularities appearing in stress kernel functions. In contrast, the existing criteria resulted in excessively high errors, exceeding those reported in section 4.1 by approximately a factor of twenty.

4.2.2 Stress assessment in the vertical direction

To further assess the stress solution, especially the variation in terms of θ , three cross sections are considered, having these respective distances from the apex $\{r = 50 \text{ mm}, 20 \text{ mm}, 5 \text{ mm}\}$ corresponding to $\{x = 5 \text{ cm}, 8 \text{ cm}, 9.5 \text{ cm}\}$, respectively.

The three graphs in Figures 20, 21, and 22 illustrate the variation of the relative errors for the Von-Mises stress along these 3 delineated sections.

Upon comparing the results presented in Figures 20–22, it is evident that the

von Mises stress computed using the various existing criteria (Lachat and Watson, Jun and Beer, Bu and Davies, and Eberwien) exhibits significantly high errors, all exceeding the required upper bound. These errors become more pronounced near the boundaries, specifically at $\theta \approx 0$ and $\theta \approx \beta = 0.05$ rad, as clearly illustrated in Figures 20–22 when approaching the left and right ends.

Furthermore, these errors increase drastically as the cross-section approaches the apex. Thus, errors illustrated in Figure 22 at $r = 5$ mm are the most pronounced.

Nevertheless, by using our new integration criterion with $\alpha = 1$, we obtained a highly stable precision that meets the target upper bound of $\epsilon = 10^{-3}$, even in the extremely thin cross section at $r = 5$ mm and at points very close to the boundary ($\theta \approx 0$ and $\theta \approx \beta = 0.05$ rad).

Once again, the proposed criterion demonstrated its ability to accurately represent the stress field distribution, even in extremely thin cross-sections and at points very close to the boundary, while maintaining a highly stable relative error with respect to the imposed error threshold.

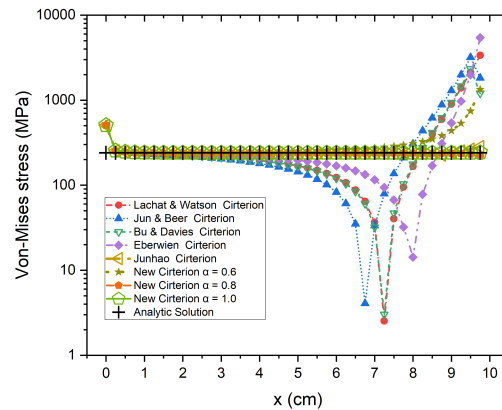


Figure 18: Variation of the von Mises equivalent stress along the fin

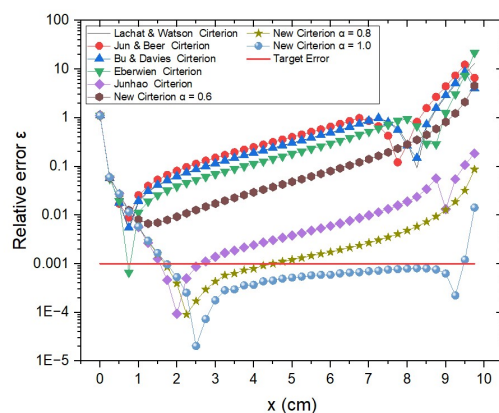


Figure 19: The variation of the relative error of von Mises stress along the horizontal direction of the fin

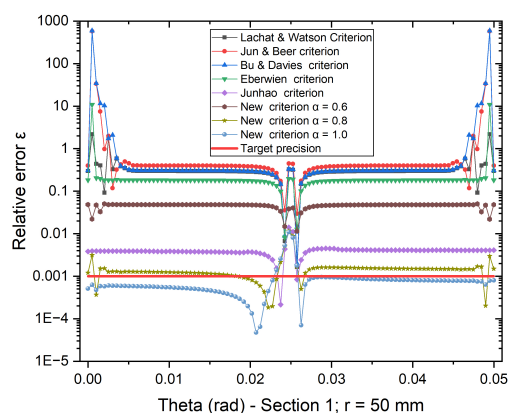


Figure 20: Relative error for the von Mises stress in terms of θ across the the section at ($r=50$ mm)

4.3 Further computational improvement

As stated in the introduction, analytical and semi-analytical techniques, particularly nonlinear variable transformations, are employed to treat near-singularities. Most of these methods aim to mitigate or eliminate the effects of near singularities, resulting in a less pronounced singular behavior. How-

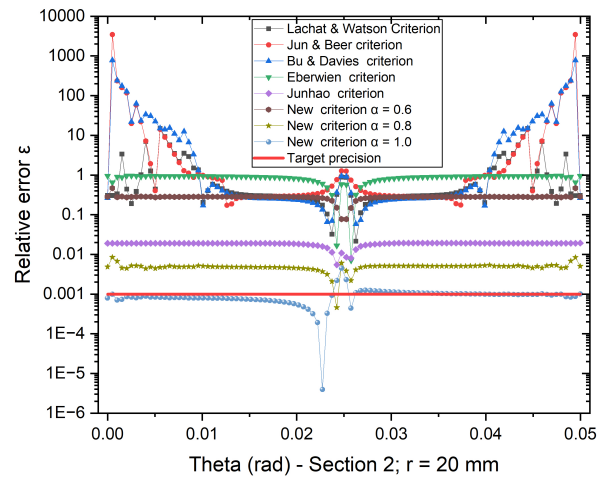


Figure 21: Relative error for the von Mises stress in terms of θ across the the section at ($r=20$ mm)

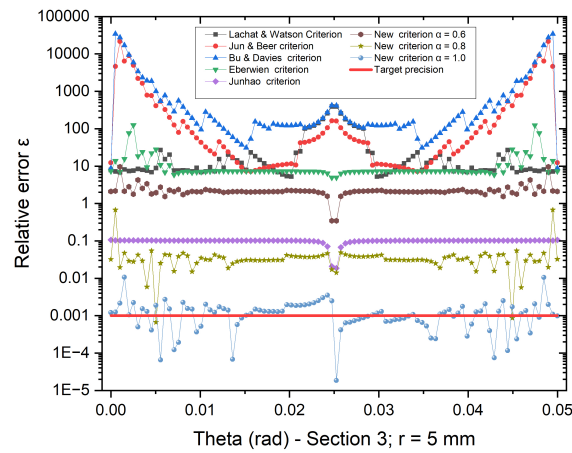


Figure 22: Relative error for the von Mises stress in terms of θ across the the section at ($r=5$ mm)

ever, in the case of very thin bodies, despite reducing the boundary layer effect, its influence remains significant.

This will be demonstrated in the current section using the sinh variable transformation technique in combination with different integration criteria to solve the previously discussed example. One of the major advantages of the proposed criterion will also be illustrated, namely, its ability to reduce computational effort by lowering the parameter α , taking advantage of the dampened singularity.

The same 40 probe points used in sections 4.1 and 4.2.1 are employed here to evaluate the displacement magnitude and Von Mises stress, respectively. The results are presented in Figures 23 and 24, showing the relative error of both quantities.

Figures 23 and 24 reveal that, despite applying the sinh transformation, the existing criteria (Lachat and Watson, Jun and Beer, Gao and Davies, and Eberwien) still yield poor accuracy, with relative errors exceeding the desired error threshold.

In contrast, the proposed criterion combined with the sinh transformation and a reduced value of $\alpha = 0.6$ delivers excellent results, comparable to those obtained with $\alpha = 1$ without variable transformation.

The combination of the Junhao criterion and the sinh transformation also produced acceptable accuracy, though not as high as that achieved by the proposed criterion.

Furthermore, as discussed in section 3.6 and shown in the log-log plot of Figure 13, the Junhao criterion's error upper-bound formula leads to more Gauss points for moderate values of λ . In this case, with $\alpha = 0.6$, the interval over which the proposed criterion outperforms the Junhao criterion is wider than when $\alpha = 1$, becoming $\lambda \in [2.36258 \times 10^{-5}, 3.19886]$.

In conclusion, the application of the sinh variable transformation allowed further computational improvement with accurate results that satisfy the prescribed precision requirements while reducing computational cost through lower values of α . Additionally, in the case of extremely thin structures, although the combination of existing criteria with the sinh transformation showed some improvement, the results remained unsatisfactory, with relative errors still exceeding the required precision threshold.

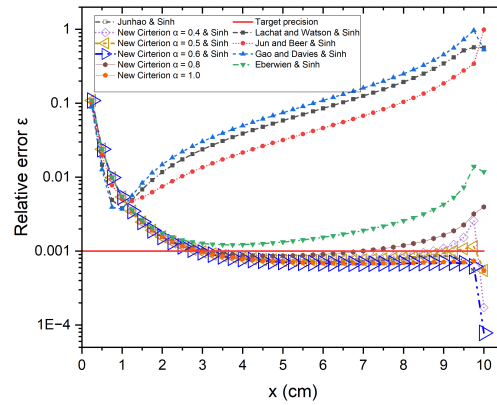


Figure 23: Relative error for the displacement magnitude when utilizing the sinh variable transformation

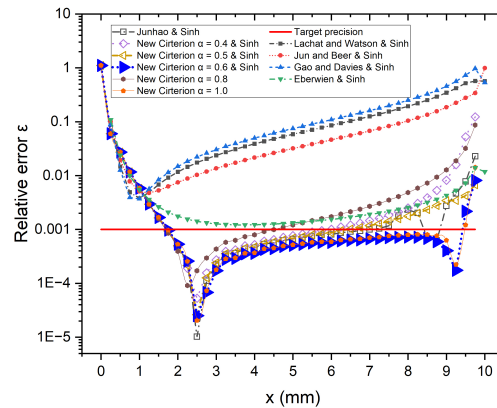


Figure 24: Relative error for the von Mises stress when utilizing the sinh variable transformation

5 Conclusion

The utilization of Gaussian quadrature for numerical integration within the BEM is practically indispensable. Consequently, the choice of quadrature order significantly impacts the accuracy and efficiency of computational codes and programs based on this method. Various criteria and error bound for-

mulas existing in BEM literature, were used for the purpose of an optimal selection of the Gauss order. The majority of these criteria and upper bound error formulas were developed under the assumption of a simplified form for the integration kernels.

In order to develop a criterion that meets high precision requirements in extremely thin bodies, a numerical testing procedure was performed to estimate the errors due to numerical integration using Gauss-Legendre quadrature, but in contrast to previous research works the complete form of the integrands is considered. From the numerically obtained results, a new integration criterion is proposed, formulated as empirical formulas with a unified structure incorporating a parameter named α . This parameter facilitates the adjustment of computational efforts, by enabling its reduction in case of nonthin bodies or after the use of semi-analytical algorithms and variable transformation techniques. Simulations are conducted on an extremely thin structure (a thin wedge), and the results are compared to the analytical solution for an infinite wedge. The comparison showed that results align well at locations relatively far from the fixed base and demonstrated the criterion's ability to achieve highly accurate results. For the optimal parameter value $\alpha = 1$, the proposed criterion outperformed existing criteria, delivering results with stable precision that satisfied the upper bound error requirement of $\epsilon = 10^{-3}$, even in regions close to the wedge apex. Furthermore, the proposed criterion showed its advantage in controlling computational efforts, which is demonstrated through the application of the sinh transformation technique, where additional computational time is reduced by lowering the value of α to 0.6.

Acknowledgements

The support from Directorate-General for Scientific Research and Technological Development (DG-RSDT) of Algerian government in the form of research grant is gratefully acknowledged. The Laboratory of Green and Mechanical Development (LGMD) of National Polytechnic School (ENP) is also gratefully acknowledged for the resources and support.

References

- [1] Aliabadi, M.H., *The boundary element method – applications in solids and structures*, Wiley, Volume 2. UK, Chichester, 2002.
- [2] Aliabadi, M.H., Hall, W.S., and Phemister, T.G., *Taylor expansions for singular kernels in the boundary element method*, Int. J. Numer. Methods Eng. 21 (1985), 2221–2236.
- [3] Araújo, F.C., and Gray, L.J., *Analysis of thin-walled structural elements via 3D standard BEM with generic substructuring*, Comput. Mech. 41 (2008), 633–645.
- [4] Bu, S., and Davies, T.G., *Effective evaluation of non-singular integrals in 3D BEM*, Adv. Eng. Softw. 23 (2) (1995), 121–128.
- [5] Chang, C., and Mear, M.E., *A boundary element method for two-dimensional linear elastic fracture analysis*, Int. J. Fract. 74 (1995), 219–25.
- [6] Chen, H.B., Lu, P., Huang, M.G., and Williams, F.W., *An effective method for finding values on and near boundaries in the elastic BEM*, Comput. Struct. 69 (1998), 421–431.
- [7] Chen, H.B., Lu, P., and Schnack, E., *Regularized algorithms for the calculation of values on and near boundaries in 2D elastic BEM*, Eng. Anal. Bound. Elem. 25 (10) (2001), 851–876.
- [8] Chernov, A., and Stephan, E.P., *Adaptive BEM for Contact Problems with Friction*, Peter Wriggers and Udo Nackenhorst (eds.), In IUTAM Symposium on Computational Methods in Contact Mechanics: Proceedings of the IUTAM Symposium held in Hannover, Germany, November 5–8, 2006, pp. 113–122. Dordrecht: Springer Netherlands, 2007.
- [9] Cruse, T.A., and Aithal, R., *Non-singular boundary integral equation implementation*, Int. J. Numer. Method. Eng. 36 (1993), 237–254.
- [10] Eberwien, U., Duencer, C., and Moser, W., *Efficient calculation of internal results in 2D BEM*, Eng. Anal. Bound. Elem. 29 (2005), 447–453.

- [11] Gao, X.W., and Davies, T.G., *Adaptive integration in elasto-plastic boundary element analysis*, J. Chin. Inst. Eng. 23 (3) (2000), 349–356.
- [12] Gu, Y., Chen, W., and Zhang, C., *Stress analysis for thin multilayered coating systems using a sinh transformed boundary element method*, Int. J. Solids Struct. 50 (20-21) (2013), 3460–3471.
- [13] Gu, Y., Zhang, C., Qu, W., and Ding, J., *Investigation on near-boundary solutions for three-dimensional elasticity problems by an advanced BEM*, Int. J. Mech. Sci. 142–143 (2018), 269–275.
- [14] Guiggiani, M., *Hypersingular formulation for boundary stress evaluation*, Eng. Anal. Bound. Elem. 13 (1994), 169–179.
- [15] Guiggiani, M., *The evaluation of Cauchy principal value integrals in the boundary element method – A review*, Math. Comput. Model. 15(3-5) (1991), 175–184.
- [16] Guiggiani, M., and Casalini, P., *Direct computation of Cauchy principal value integrals in advanced boundary elements*, Int. J. Numer. Methods Eng. 24 (1987), 1711–1720.
- [17] Guiggiani, M., and Casalini, P., *Rigid body translation with curved boundary elements*, Appl. Math. Model. 13(6) (1989), 365–368.
- [18] Gwinner, J. and Stephan, E.P., *A boundary element procedure for contact problems in plane linear elastostatics*, ESAIM: Math. Model. Numer. Anal. 27 (4) (1994), 457–480.
- [19] Jing, R., Yu, B., Ren, S., and Yao, W., *A novel SCTBEM with inversion-free Padé series expansion for 3D transient heat transfer analysis in FGMs*, Comput. Method. Appl. Mech. Eng. 433 (2025), 117546.
- [20] Johnston, P.R., and Elliott, D., *Transformations for evaluating singular boundary element integrals*, J. Comput. Appl. Math. 146 (2) (2002), 231–251.
- [21] Johnston, P.R., and Elliott, D., *A sinh transformation for evaluating nearly singular boundary element integrals*, Int. J. Numer. Method. Eng. 62 (2005), 564–578.

- [22] Jun, L., Beer, G., and Meek, J.L., *Efficient evaluation of integrals of order $1/r$, $1/r^2$, $1/r^3$ using Gauss quadrature*, Eng. Anal. 2 (3) (1985), 118–123.
- [23] Junhao, H., Zhipeng, W., and Yongqiang, C., *A new error upper bound formula for Gaussian integration in boundary integral equations*, Eng. Anal. Bound. Elem. 112 (2020), 39–45.
- [24] Kachanov, M., Shafiro, B., and Tsukrov, I., *Handbook of elasticity solutions*, Springer Science+Business Media Dordrecht, B.V, 2003.
- [25] Lachat, J.C., and Watson, J.O., *Effective numerical treatment of boundary integral equations: A formulation for three dimensional elastostatics*, Int. J. Numer. Method. Eng. 10 (1976), 991–1005.
- [26] Li, S., and Mear, M.E., *Singularity-reduced integral equations for displacement discontinuities in three-dimensional linear elastic media*, Int. J. Fract. 93 (1998), 87–114.
- [27] Ma, H., and Kamiya, N., *Distance transformation for the numerical evaluation of near singular boundary integrals with various kernels in boundary element method*, Eng. Anal. Bound. Elem. 26 (2002), 329–339.
- [28] Mi, Y., and Aliabadi, M.H., *A Taylor expansion algorithm for integration of 3D near-singular integrals*, Commun. Appl. Numer. Method. Eng. 12 (1996), 51–62.
- [29] Mustoe, G.G.W., *Advanced integration schemes over boundary elements and volume cells for two-and three-dimensional non-linear analysis*, Develop. Bound. Elem. Method. 3 (1984), 213–270.
- [30] Newman-Jr, J.C., Mear, J.M., and Raju, I.S., *Computer simulation of elastic stress analyses of two-dimensional multiple-cracked bodies*, 37th Structure, Structural Dynamics and Materials Conference, AIAA Meeting Papers on Disc 1996, 869–876.
- [31] Padhi, G.S., Shenoi, R.A., Moy, S.S.J., and McCarthy, M.A., *Analytic integration of kernel shape function product integrals in the boundary element method*, Comput. Struct. 79 (14) (2001), 1325–1333.

- [32] Paulsen, K.D., and Lynch, D.R., *Calculation of interior values by the boundary element method*, Commun. Appl. Numer. Method. 5 (1989), 7–14.
- [33] Portella, A., Aliabadi, M.H., and Rooke, D.P., *The dual boundary element method: effective implementation for crack problems*, Int. J. Numer. Method. Eng. 33 (1992), 1269–1287.
- [34] Portella, A., Aliabadi, M.H., and Rooke, D.P., *Dual boundary element incremental analysis of crack propagation*, Comput. Struct. 46 (02) (1993), 237–247.
- [35] Rudolphi, T.J., *The use of simple solutions in the regularization of hypersingular boundary integral equations*, Math. Comput. Model. 15 (3–5) (1991), 269–278.
- [36] Salvadori, A., *Analytical integrations in 2D BEM elasticity*, Int. J. Numer. Method. Eng. 53 (2002), 1695–1719.
- [37] Sladek, V., Sladek, J., and Tanaka, M., *Nonsingular BEM formulations for thin-walled structures and elastostatic crack problems*, Acta Mech. 99 (1993), 173–190.
- [38] Sladek, V., Sladek, J., and Tanaka, M., *Optimal transformations of the integration variables in computation of singular integrals in BEM*, Int. J. Numer. Method. Eng. 47 (2000), 1263–1283.
- [39] Stroud, A.H., and Secrest, D., *Gaussian quadrature formulas*, Prentice-Hall Inc, Englewood Cliffs, N.J. 1966.
- [40] Telles, J.C.F., *A self-adaptive coordinate transformation for efficient numerical evaluation of general boundary element integrals*, Int. J. Numer. Method. Eng. 24 (1987), 959–973.
- [41] Timoshenko, S., and Goodier, J.N., *Theory of elasticity*, McGraw-Hill book company, New York, 1951.
- [42] Yu, B., and Jing, R., *SCTBEM: A scaled coordinate transformation boundary element method with 99-line MATLAB code for solving Poisson's equation*, Comput. Phys. Commun. 300 (2024), 109185.

- [43] Zhang, X. and Zhang, X. *Exact integration in the boundary element method for two-dimensional elastostatic problems*, Eng. Anal. Bound. Elem. 27 (2003), 987–997.
- [44] Zhang, X. and Zhang, X. *Exact integration for stress evaluation in the boundary element analysis of two-dimensional elastostatics*, Eng. Anal. Bound. Elem. 28 (2004), 997–1004.
- [45] Zhang, Y., Gong, Y., and Gao, X., *Calculation of 2D nearly singular integrals over high-order geometry elements using the sinh transformation*, Eng. Anal. Bound. Elem. 60 (2015), 144–153.
- [46] Zhang, Y., and Gu, Y., *An effective method in BEM for potential problems of thin bodies*, J. Mar. Sci. Technol. 18 (2010), 137–144.
- [47] Zhang, Y., Gu, Y., and Chen, J.T., *Analysis of 2D thin-walled structures in BEM with high-order geometry elements using exact integration*, Comput. Model. Eng. Sci. 50 (2009), 1–20.
- [48] Zhou, W., Yang, X., and Chen, Y., *Adaptive sinh transformation Gaussian quadrature for 2D potential problems using deep learning*, Eng. Anal. Bound. Elem. 155 (2023), 197–211.



Numerical solution of nonlinear diffusion-reaction in porous catalysts using quantum spectral successive linearization method

S. Abbasbandy 

Abstract

Significant advances in quantum computing science have been achieved through the development of general-purpose quantum solvers for linear differential equations in several studies. This study employs a hybrid quantum-spectral approach to analyze nonlinear reaction-diffusion dynamics in porous catalysts. Two distinct forms of nonlinearity in the model are examined. The successive linearization method is applied to linearize the governing equations. Within an iterative framework, the final quantum state is constructed by progressively integrating the quantum state from each iteration, enabled by an efficient quantum algorithm. Numerical simulations validate the method's efficacy, showing favorable agreement with existing literature. Moreover, the approach delivers accurate solutions even for large Thiele modulus values.

Received 9 July 2025; revised 16 August 2025; accepted 19 August 2025

Saeid Abbasbandy

Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin, Iran. e-mail: abbasbandy@yahoo.com, abbasbandy@sci.ikiu.ac.ir.

How to cite this article

Abbasbandy, S., Numerical solution of nonlinear diffusion-reaction in porous catalysts using quantum spectral successive linearization method. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1464-1481. <https://doi.org/10.22067/ijnao.2025.94329.1677>

AMS subject classifications (2020): 65L05, 68Q09, 81P68.

Keywords: Catalyst pellet; Diffusion and reaction; Thiele modulus; Non-linear reactive transport model.

1 Introduction

Nonlinear phenomena are essential, and many aspects of diffusion and reaction in porous catalysts have been thoroughly investigated. As we know, finding the closed form solution to a problem is very difficult; in many cases, it is impossible to find the solution. In this case, finding semi-analytical or numerical solutions is significant, for example, perturbation methods [34, 38], nonperturbation methods [30], Adomian decomposition method [3, 32], δ -expansion method [23], and homotopy analysis method (HAM) [1, 9, 18, 28].

In recent years, artificial intelligence and deep learning methods have been increasingly applied to compute approximate analytical and semi-analytical solutions for differential equations. Deep learning is a subset of machine learning that utilizes multi-layered artificial neural networks to model complex patterns in data. By automatically extracting hierarchical features—from low-level details to high-level abstractions—deep learning has revolutionized fields such as computer vision, natural language processing, and scientific research. Architectures like convolutional neural networks, recurrent neural networks, and transformers power applications ranging from image recognition and autonomous driving to language translation and drug discovery. While its success relies on large datasets and significant computational resources, advancements in efficiency (e.g., lightweight models, federated learning) and interpretability continue to expand its potential. Despite challenges like overfitting and ethical concerns, deep learning remains at the forefront of artificial intelligence, driving innovations that reshape industries and everyday life, [16, 17, 15].

The study of coupled diffusion and reaction in porous catalysts is an important topic in chemical engineering, as it involves a nonlinear problem. Thiele [39] derived an analytical solution for first-order reactions; for higher-order reactions and further details, see [5, 33, 35]. The HAM was employed

by Abbasbandy [1] to investigate the nonlinear coupled diffusion-reaction dynamics in porous catalysts. The steady-state reactive transport model (RTM) was recently studied by Ganie et al. [19], generalizing nonlinear reaction-diffusion dynamics in porous catalysts for micro-vessel applications. We study their examples in this article. While many numerical methods address nonlinear problems in heat transfer, fluid dynamics, and chemical reactors, their accuracy often hinges on reaction rates and initial conditions, precluding the derivation of series solutions.

Several studies have designed general-purpose quantum solvers for linear differential equations, achieving notable advances in quantum oracle complexity. For global numerical differentiation methods, in [2, 11], it was introduced a spectral-based solver by using the Chebyshev polynomials of the first kind. As established by the Remez algorithm [161,162], Chebyshev polynomials provide an optimal basis for L_∞ -norm approximation [26].

We present a quantum algorithm for solving linear ordinary differential equations (ODEs) using spectral methods—a global approximation technique that contrasts with finite difference approaches. Our method efficiently handles time-dependent initial and boundary value problems with favorable computational complexity. Quantum computing has seen growing interest in solving differential equations. For instance, Leyton and Osborne [27] introduced a quantum Euler method for nonlinear ODEs with polynomial nonlinearities, achieving logarithmic complexity in system dimension but exhibiting exponential scaling in evolution time (a fundamental limitation for general nonlinear ODEs). Other advances focus on partial differential equations (PDEs). Clader, Jacobs, and Sprouse [12] employed the quantum linear systems algorithm (QLSA) in a finite element method for Maxwell's equations, while Costa, Jordan, and Ostrander [13] leveraged Hamiltonian simulation for finite difference approximations of the wave equation. More recently, Arrazola et al. [6] proposed a continuous-variable quantum algorithm for nonhomogeneous linear PDEs in initial value problems (IVPs).

This article presents a quantum pseudo-spectral method (QPSM) for solving the nonlinear diffusion-reaction model in porous catalysts [1, 19]. The governing equations consist of nonlinear second-order differential equations subject to boundary value problems (BVPs). The application of QPSM to

linear IVPs via the QLSA was first presented in [11]. Subsequent work in [2] has demonstrated its effectiveness for Lane–Emden type equations. The QLSA for a linear sparse system of d equations produces a quantum state proportional to the solution system in time $\text{poly}(\log d)$, and it is improved in [10]. Berry [7] developed a quantum algorithm employing finite difference schemes to solve general linear ODEs, demonstrating $\text{poly}(1/\epsilon)$ complexity relative to solution accuracy ϵ . Later, the complexity is improved to $\text{poly}(\log(1/\epsilon))$ in [8]. The quantum algorithm in [11] is based on a pseudo-spectral method for time-dependent IVP and BVP with the complexity $\text{poly}(\log d, \log(1/\epsilon))$. Newly, by using Carleman linearization, the nonlinear quadratic ODEs are considered by a quantum algorithm with complexity $E^2 q \text{poly}(\log E; \log d; \log 1/\epsilon)/\epsilon$, where E is the evolution time and q measures decay of the solution, [14, 25, 29].

Our analysis focuses on solving the governing equation for a one-dimensional steady-state reactive model:

$$\frac{d^2 u}{dx^2} - \varphi^2 u^N = 0, \quad x \in [0, 1], \quad (1)$$

and the boundary conditions

$$\left. \frac{du}{dx} \right|_{x=0} = 0, \quad u(1) = 1, \quad (2)$$

where φ is Thiele modulus [31, 39] and N is the order reaction. The successive linearization method (SLM) is first employed, as QPSM demonstrates particular efficiency for linear systems. Hence we have a hybrid quantum spectral successive linearization method (QSSLM). We consider an expansion of the unknown function $u(\cdot)$ in the form:

$$u(\tau) = U_i(\tau) + \sum_{k=0}^{i-1} u_k(\tau), \quad i = 1, 2, 3, \dots \quad (3)$$

Let $U_i(\cdot)$ denote the unknown functions and let $u_k(\cdot)$ be their successive approximations. The latter are obtained via QPSM, which solves the linearized version of the original equation—obtained by inserting (3) into (1)—in a recursive manner. Applying (3) in (1) by considering the Binomial theorem,

we have

$$U_i'' - \varphi^2 \sum_{l=0}^N \binom{N}{l} U_i^{N-l} \left(\sum_{j=0}^{i-1} u_j \right)^l = - \sum_{j=0}^{i-1} u_j''. \quad (4)$$

We choose the constant function $u_0(\cdot) = 1$ as our initial approximation, which satisfies the boundary conditions specified in (2). Also, we assume that $\lim_{i \rightarrow \infty} U_i = 0$. For $i \geq 1$, each $u_i(\cdot)$ is determined by solving the linear IVP resulting from the linearization of (4) for $i \in [M] = \{1, 2, 3, \dots, M\}$:

$$u_i'' - \varphi^2 N b_{i-1}^{N-1} u_i = \mu_{i-1}, \quad (5)$$

where

$$b_{i-1}(\cdot) = \sum_{j=0}^{i-1} u_j(\cdot),$$

$$\mu_{i-1}(\cdot) = -b_{i-1}''(\cdot) + \varphi^2 b_{i-1}^N(\cdot).$$

The solutions $u_{i \geq 1}(\cdot)$ satisfy the initial conditions $u_i(0) = 0$ and $u_i'(0) = 0$, where M denotes the number of iterations. Prior to applying the quantum spectral method, we first transform the IVP (5) into an equivalent system of first-order ODEs:

$$y_{i,0}'(x) = y_{i,1}(x), \quad (6)$$

$$y_{i,1}'(x) = \mu_{i-1}(x) + \varphi^2 N b_{i-1}^{N-1}(x) y_{i,0}(x).$$

This system satisfies $y_{i,0}(0) = y_{i,1}(0) = 0$, where $y_{i,0} = u_i$ and $y_{i,1} = u_i'$. Following (6), consider the coefficient matrix as follows:

$$A_i(\cdot) = \begin{bmatrix} 0 & 1 \\ \varphi^2 N b_{i-1}^{N-1}(\cdot) & 0 \end{bmatrix}.$$

The spectral approach represents solutions as linear combinations of orthogonal polynomials (typically Chebyshev), with differential equations transformed to algebraic equations through pseudo-spectral discretization at the Gauss-Lobatto nodes [37]. Applying this framework, we employ a truncated Chebyshev expansion within the Chebyshev pseudo-spectral method [22] to

(5), yielding:

$$u_i(\cdot) = \sum_{j=0}^n c_{i,j} T_j(\cdot), \quad (7)$$

for any $n \in \mathbb{Z}^+$ and Chebyshev polynomials, $T_j(\cdot)$. Consistent with Chebyshev polynomial theory, we work on $[-1, 1]$ in (7) and accordingly rescale (1). The numerical solution utilizes Chebyshev–Gauss–Lobatto collocation points $\xi_l = \cos(\frac{l\pi}{n})$ for $l \in [n+1]_0 = \{0, 1, \dots, n\}$ to discretize the linear system obtained from (5). For computing $c_{i,k}$, we have

$$\frac{du_i(\xi)}{d\xi} = \sum_{k=0}^n \sum_{j=0}^n [D_n]_{k,j} c_{i,j} T_k(\xi),$$

where the matrix D_n is defined in [40]. Using (6) and (7), for $i \in [M]$ and $l \in [n+1]_0$, we have

$$\begin{aligned} \sum_{j=0}^n T_j(\xi_l) c'_{i,0,j} &= \sum_{j=0}^n T_j(\xi_l) c_{i,1,j}, \\ \sum_{j=0}^n T_j(\xi_l) c'_{i,1,j} &= \mu_{i-1} + \varphi^2 N b_{i-1}^{N-1}(\xi_l) \sum_{j=0}^n c_{i,0,j} T_j(\xi_l), \end{aligned} \quad (8)$$

where $c_{i,l,\cdot}$ and $c'_{i,l,\cdot}$ are Chebyshev coefficients related to functions $y_{i,l}(\cdot)$ and $y'_{i,l}(\cdot)$ for $l = 0, 1$, respectively. The linear system (8) is solved by the QLSA [10], which achieves a complexity of $\text{poly}(\log 2)$. The convergence properties of this approach are supported by theoretical results from [11, 20]. Specifically, we have the following properties:

- If $u \in C^{r+1}(-1, 1)$, then the error norm scales as $\mathcal{O}(n^{2-r})$.
- For $u \in C^\infty(-1, 1)$, then the error norm exhibits spectral convergence, scaling as $\sqrt{\frac{2}{\pi}} \left(\frac{e}{2n}\right)^n$.

In the latter case, selecting $n = \text{poly}(\log(1/\epsilon))$ ensures an ϵ -approximation of the solution. In solving IVPs, to improve the accuracy, we usually divide the region including the independent variable into some subintervals. Indeed here, we want to solve a BVP, and hence, we cannot do this. Hence, by increasing the values of n and M , we improve the accuracy. For using the Chebyshev polynomials, we should convert $[0, 1]$ in (1) to $[-1, 1]$. After this,

$x \in [0, 1]$ converts to $\xi \in [-1, 1]$ by $\xi = K(x) = 1 - 2x$, and conversely, $x = IK(\xi) = \frac{(1-\xi)}{2}$. Hence, (6) converts to

$$\begin{aligned} z'_{i,0}(\xi) &= z_{i,1}(\xi), \\ z'_{i,1}(\xi) &= \eta_{i-1}(\xi) + \varphi^2 \frac{N}{D^2} b_{i-1}^{N-1}(\xi) z_{i,0}(\xi), \end{aligned} \quad (9)$$

where

$$\begin{aligned} b_{i-1}(\cdot) &= \sum_{j=0}^{i-1} z_{j,0}(\cdot), \\ \eta_{i-1}(\cdot) &= -\left(b''_{i-1}(\cdot) - \varphi^2 \frac{1}{D^2} b_{i-1}^N(\cdot)\right), \end{aligned}$$

and $D = \frac{d\xi}{dx}$ and $z_{i,v}(\xi) = y_{i,v}(\xi) = y_{i,v}(IK_h(\xi))$ for $\xi \in [0, 1]$, $i \in [M]$ and $v \in \{0, 1\}$. For convenience, we set $z_{0,0} = 1$ and consequently $z_{0,1} = 0$. Following (9), we have

$$A_i(\xi) = \begin{bmatrix} 0 & 1 \\ \frac{N\varphi^2}{D^2} b_{i-1}^{N-1}(\xi) & 0 \end{bmatrix}.$$

2 Quantum implementation

For a better understanding of quantum computing symbols, see [41]. Here, we analyze the linear system

$$L_i |X_i\rangle = |B_i\rangle, \quad (10)$$

derived from the previous section. Using quantum computations applied to (9), we solve this system to obtain approximate solutions at the terminal point $x = 1$ for each iteration $i \in [M]$. The vector $|X_i\rangle \in \mathbb{C}^{p+2} \otimes \mathbb{C}^2 \otimes \mathbb{C}^{n+1}$ describes the solution by

$$|X_i\rangle = \sum_{v=0}^1 \sum_{j=0}^n c_{i,v,j} |0vj\rangle + \sum_{h=1}^{p+1} \sum_{v=0}^1 \sum_{j=0}^n x_{iv} |hvj\rangle. \quad (11)$$

The coefficient $c_{i,v,l}$ expands $z_{i,v}(K(1))$ in Chebyshev series, while $x_{iv} = z_{i,v}(K(0))$ gives the final state. The padding parameter p [21, 8] amplifies the solution for reliable quantum measurement. Thus, the solution at $x = 1$ (the final state) is given by $1 + \sum_{i=1}^M x_{i1}$, where the constant term accounts for the initial approximation $u_0(\cdot) = 1$.

In (1),

$$|B_i\rangle = \sum_{v=0}^1 0|v0\rangle + \sum_{j=1}^n 0|0j\rangle + \sum_{j=1}^n \eta_{i-1}(\xi_j)|1j\rangle,$$

and

$$\begin{aligned} L_i &= |0\rangle\langle 0| \otimes (L1 + L2(A_i)) + |1\rangle\langle 0| \otimes L3 \\ &\quad + \sum_{h=1}^{p+1} |h\rangle\langle h| \otimes L4 + \sum_{h=1}^{p+1} |h\rangle\langle h-1| \otimes L5. \end{aligned}$$

The matrix $L1$ is a discrete representation for derivative process; that is,

$$|0\rangle\langle 0| \otimes L1|X_i\rangle = \sum_{v=0}^1 \sum_{k=0}^n T_k(\xi_0) c_{i,v,k} |0v0\rangle + \sum_{v=0}^1 \sum_{j=1, k, r=0}^n T_k(\xi_j) [D_n]_{k,r} c_{i,v,r} |0vj\rangle,$$

and hence

$$\begin{aligned} L1 &= \sum_{v=0}^1 \sum_{k=0}^n T_k(\xi_0) |v0\rangle\langle vk| + \sum_{v=0}^1 \sum_{j=1, k, r=0}^n \cos \frac{kj\pi}{n} [D_n]_{k,r} |vj\rangle\langle vr| \\ &= I_2 \otimes \left(|0\rangle\langle 0| P_n + \sum_{j=1}^n |j\rangle\langle j| P_n D_n \right), \end{aligned}$$

where

$$P_n = \sum_{j,k=0}^n \cos \frac{kj\pi}{n} |j\rangle\langle k|.$$

The matrix $L2(A_i)$ discretizes $A_i(\xi)$ as follows:

$$|1\rangle\langle 1| \otimes L2(A_i)|X_i\rangle = - \sum_{v,\varrho=0}^1 \sum_{j=1, k=0}^n A_i(\xi_j)_{v,\varrho} T_k(\xi_j) c_{i,\varrho,k} |1vj\rangle,$$

and then

$$L2(A_i) = - \sum_{v,\varrho=0}^1 \sum_{j=1,k=0}^n A_i(\xi_j)_{v,\varrho} \cos \frac{kj\pi}{n} |vj\rangle \langle \varrho k| = - \sum_{j=1}^n A_i(\xi_j) \otimes |j\rangle \langle j| P_n.$$

In practice, the Chebyshev coefficients in $L3$ are chosen to ensure patchwise continuity; that is, the endpoint of one subinterval $z_{i,h}(-1)$ aligns with the start of the next $z_{i,h+1}(1)$,

$$L3 = - \sum_{k=0}^n (-1)^k |0\rangle \langle k|.$$

The matrices $L3$ and $L4$ optimize the success probability of the final quantum measurement. Specifically, $L4$ processes the output of $L3$ (for $l = 0$) to construct x_{iv} , iterating this operation n times across all $l \in [n]$. Hence $L3$ and $L4$ repeat x_{iv} , $(n+1)p$ times for $l \in [n]$, and then for an artificial parameter h , we have

$$L3 = - \sum_{k=0}^n (-1)^k |0\rangle \langle k|,$$

$$L4 = - \sum_{v=0}^1 \sum_{j=1}^n |vj\rangle \langle v(j-1)| + \sum_{v=0}^1 \sum_{j=0}^n |vj\rangle \langle vj|,$$

and

$$L5 = - \sum_{v=0}^1 |v0\rangle \langle vn|.$$

Remark 2.1. Because of the boundary condition $u(1) = 1$ in (2), we have $\xi_0 \leftrightarrow \xi_n$ in constructing $L1$.

3 Numerical results

Here, we examine several examples with varying values of φ and N in (1) to demonstrate the method's accuracy and compare it with other approaches. All computations are performed using Python 3.12.4 on a DESKTOP with

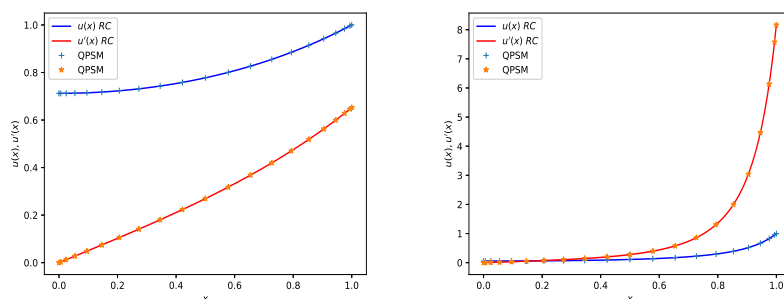


Figure 1: Comparison for $N = 2$, $\varphi = 1$ (Left), $\varphi = 10$ (Right): ($n = 20, M = 10$).

Intel(R) Core(TM) i7-7700 CPU with 8.00 GB RAM, and for simplicity, we set $p = 1$ in every case. We consider only the nonlinear case in the following examples in [1] and illustrate the results for $N = 0.5, 2, 4$ for various φ . For comparison, we consider the numerical method based on residual control (RC) [24] and HAM [1] and the finite difference method (FDM) based on the three-stage Lobatto formula with fourth-order accuracy for C^1 -continuous solution [36]. Figures 1, 2, and 3 show $u(\cdot)$ and $u'(\cdot)$ for QPSM (present method) and others. Figure 4 shows the residual errors in (1), which show the efficiency of QPSM. Only in Figure 3, the obtained results by RC are not fine, and for this reason, we compare QPSM with the HAM solution. Table 1 shows a comparison between QPSM with $n = 20$ and $M = 10$ (for $\varphi \geq 10$, we put $n = 30$) and other methods. In this table, the results for $\varphi = 0.5$ in RC are not fine. The last column of Table 1 presents the CPU time in seconds, which is remarkably small and noteworthy.

4 The reactive transport model

We will consider the RTM (reactive transport model) dynamics with the variance in the half-saturation concentration, α , and the characteristic reaction rate, β , as follows [4, 19]:

$$\frac{d^2 u}{dx^2} - \frac{\beta u(x)}{\alpha + u(x)} = 0, \quad x \in [0, 1], \quad (12)$$

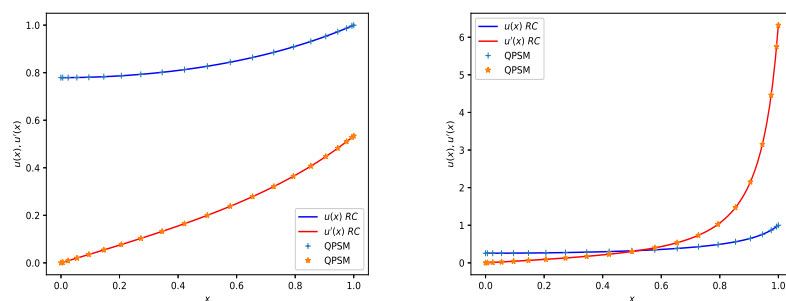


Figure 2: Comparison for $N = 4$, $\varphi = 1$ (Left), $\varphi = 10$ (Right): ($n = 20, M = 10$).

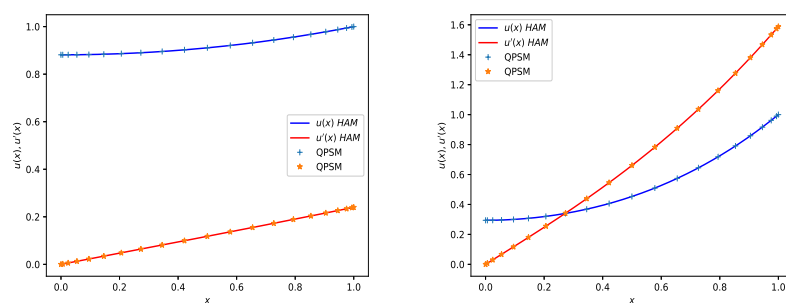


Figure 3: The Comparison for $N = 0.5$, $\varphi = 0.5$ (Left), $\varphi = 1.5$ (Right): ($n = 20, M = 10$).

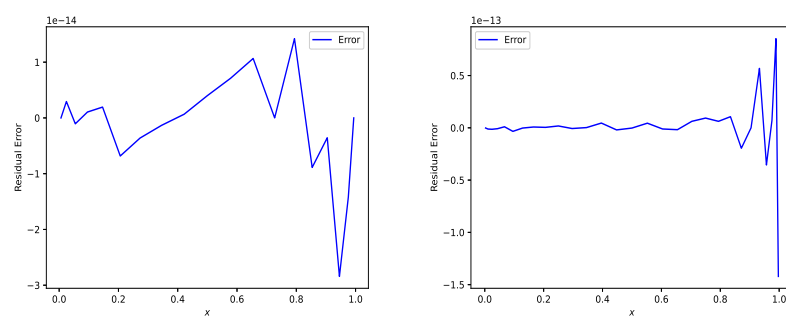


Figure 4: Residual error for $N = 2$, (Left), $N = 4$ (Right): ($\varphi = 10, n = 20, M = 10$).

Table 1: Results for $u(0)$ at various values of N and φ .

| N | φ | QPSM | HAM | FDM | RC | CPU Time (QPSM) |
|-----|-----------|-------------|-----------|-----------|-----------|-----------------|
| 0.5 | 0.5 | 0.88135825 | 0.881358 | 0.881359 | — | 0.43 |
| 0.5 | 1 | 0.59444614 | 0.594446 | 0.594447 | — | 0.43 |
| 0.5 | 1.5 | 0.29428991 | 0.294290 | 0.294290 | — | 0.43 |
| 2 | 1 | 0.71225634 | 0.712256 | 0.712257 | 0.712256 | 0.30 |
| 2 | 2 | 0.44372272 | 0.443723 | 0.443723 | 0.443723 | 0.29 |
| 2 | 4 | 0.21259027 | 0.212590 | 0.212591 | 0.212590 | 0.29 |
| 2 | 10 | 0.057084208 | 0.0570842 | 0.0570843 | 0.0570842 | 0.42 |
| 2 | 20 | 0.017555306 | 0.0175553 | 0.0175615 | 0.0175553 | 0.29 |
| 4 | 1 | 0.779145162 | 0.779145 | 0.779148 | 0.779146 | 0.75 |
| 4 | 10 | 0.257002936 | 0.257003 | 0.257005 | 0.257002 | 0.75 |

and the boundary conditions (2); that is, $u'(0) = 0$, $u(1) = 1$.

In this case after linearization, (9) converts to

$$\begin{aligned} z'_{i,0}(\xi) &= z_{i,1}(\xi), \\ z'_{i,1}(\xi) &= \eta_{i-1}(\xi) + \frac{1}{D^2} \frac{\beta\alpha}{(\alpha + b_{i-1}(\xi))^2} z_{i,0}(\xi), \end{aligned} \quad (13)$$

where

$$\begin{aligned} b_{i-1}(\cdot) &= \sum_{j=0}^{i-1} z_{j,0}(\cdot), \\ \eta_{i-1}(\cdot) &= -\left(b''_{i-1}(\cdot) - \frac{1}{D^2} \frac{\beta b_{i-1}(\cdot)}{\alpha + b_{i-1}(\cdot)}\right), \end{aligned}$$

and hence the matrix of coefficients is

$$A_i(\xi) = \begin{bmatrix} 0 & 1 \\ \frac{1}{D^2} \frac{\beta\alpha}{(\alpha + b_{i-1}(\xi))^2} & 0 \end{bmatrix}.$$

Figure 5 shows a comparison between QPSM and RC for $n = 20$ and $M = 5$ for different values of α and β . These values for α and β are chosen from [19] for better comparison, the system of (13) is independent of the values of α and β . In Table 2, QPSM is compared with the Runge–Kutta method (RK4), particle swarm optimization (PSO), and a hybrid of PSO-sequential quadratic programming (PSO-SQP) [19]. In this table, we report only the values of $u(0)$ because it is a missed value and important.

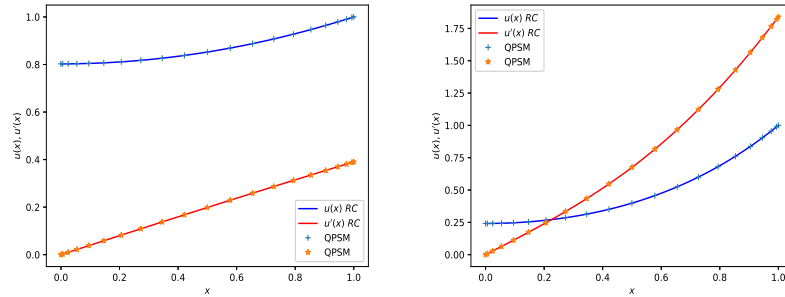


Figure 5: Comparison for $\alpha = -0.2, \beta = 0.3$ (Left), $\alpha = 1, \beta = 6$ (Right): ($n = 20, M = 5$).

Table 2: Results for $u(0)$ at various values of α and β .

| α | β | QPSM | RK4 | PSO | PSO-SQP |
|----------|---------|--------------|-------------|-------------|-------------|
| 0.2 | 0.5 | 0.7985197734 | 0.798520 | 0.798521 | 0.798520 |
| -0.2 | 0.3 | 0.8026489484 | 0.802648942 | 0.802647404 | 0.802648945 |
| 1 | 6 | 0.2412718196 | 0.241272 | 0.241277 | 0.241272 |

5 Conclusions

This paper introduced a new method for solving the nonlinear diffusion-reaction model in porous catalysts, with potential applications in soft tissues and microvessels, using a QPSM. These highly nonlinear models were treated using the SLM to convert the governing equations into a linearized form. Comparative studies with numerical and semi-analytical methods demonstrated the efficiency of the proposed approach. By selecting appropriate parameter values, the method yields highly accurate results. Results showed consistent advantages in speed and accuracy over prior work. Although the initial guess represents a critical step in the linearization method for solving nonlinear problems. Future research will explore extensions to high-order fractional differential equations and alternative QPSM collocation schemes.

Acknowledgements: We thank the anonymous reviewers for helpful comments, which lead to definite improvement in the manuscript.

Declarations

Conflict of Interest: The author declares that has no conflict of interest.

Funding: The author declares that this research received no grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing interests: The author declares that has no conflict of interest.

Data availability: Our manuscript has no associated data.

References

- [1] Abbasbandy, S., *Approximate solution for the nonlinear model of diffusion and reaction in porous catalysts by means of the homotopy analysis method*, Chem. Eng. J., 136 (2008) 144–150.
- [2] Abbasbandy, S., *Series and rational solutions of the second kind Painlevé equations by using quantum pseudo-spectral method*, Int. J. Math. Math. Sci. 2025 (2025) 9705701.
- [3] Adomian G., *Solving frontier problems of physics: The decomposition method*, Kluwer Academic, Dordrecht, 1994.
- [4] Ahmad, I., Ilyas, H., Urooj, A., Aslam, M.S., Shoaib, M. and Raja, M.A.Z., *Novel applications of intelligent computing paradigms for the analysis of nonlinear reactive transport model of the fluid in soft tissues and microvessels*, Neural Comput. Appl., 31 (2019) 9041–9059.

- [5] Aris, R., *Mathematical theory of diffusion and reaction in permeable catalyst*, Oxford University Press, London, 1975.
- [6] Arrazola, J.M., Kalajdzievski, T., Weedbrook, C. and Lloyd, S., *Quantum algorithm for nonhomogeneous linear partial differential equations*, Phys. Rev. A 100 (2019) 032306.
- [7] Berry, D.W., *High-order quantum algorithm for solving linear differential equations*, J. Phys. A 47(10) (2014) 105301.
- [8] Berry, D.W., Childs, A.M., Ostrander, A. and Wang, G., *Quantum algorithm for linear differential equations with exponentially improved dependence on precision*, Commun. Math. Phys. 356(3) (2017) 1057–1081.
- [9] Biazar, J., Dehghan, M., Houlari, T., *Using homotopy analysis method to find the eigenvalues of higher order fractional Sturm–Liouville problems*, Iran. J. Numer. Anal. Optim. 10(1) (2020) 49–62.
- [10] Childs, A.M., Kothari, R. and Somma, R.D., *Quantum linear systems algorithm with exponentially improved dependence on precision*, SIAM J. Comput. 46 (2017) 1920–1950.
- [11] Childs, A.M. and Liu, J.P., *Quantum spectral methods for differential equations*, Commun. Math. Phys. 375 (2020) 1427–1457.
- [12] Clader, D.B., Jacobs, B.C. and Sprouse, C.R., *Preconditioned quantum linear system algorithm*, Phys. Rev. Lett. 110 (2013) 250504.
- [13] Costa, P., Jordan, S. and Ostrander, A., *Quantum algorithm for simulating the wave equation*, Phys. Rev. A 99 (2019) 012323.
- [14] Costa, P.C.S., Schleich, P., Morales, M.E.S. and Berry, D.W., *Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling*, Npj Quantum Inf. 11 (1) (2025) 141.
- [15] Dana Mazraeh, H. and Parand, K., *GEPINN: An innovative hybrid method for a symbolic solution to the Lane–Emden type equation based on grammatical evolution and physics-informed neural networks*, Astron. Comput. 48 (2024) 100846.

- [16] Dana Mazraeh, H. and Parand, K., *A three-stage framework combining neural networks and Monte Carlo tree search for approximating analytical solutions to the Thomas–Fermi equation*, J. Comput. Sci. 87 (2025) 102582.
- [17] Dana Mazraeh, H. and Parand, K., *An innovative combination of deep Q-networks and context-free grammars for symbolic solutions to differential equations*, Eng. Appl. Artif. Intell. 142 (2025) 109733.
- [18] Derakhshan, M. and Aminataei, A., *Comparison of homotopy perturbation transform method and fractional Adams–Bashforth method for the Caputo–Prabhakar nonlinear fractional differential equations*, Iran. J. Numer. Anal. Optim. 10(2) (2020) 63–85.
- [19] Ganie, A.H., Rahman, I.U., Sulaiman, M. and Nonlaopon, K., *Solution of nonlinear reaction-diffusion model in porous catalysts arising in microvessel and soft tissue using a metaheuristic*, IEEE Access, 10 (2022) 41813–41827.
- [20] Gheorghiu, C.I., *Spectral methods for differential problems*, Casa Cartii de Stiinta Publishing House, Cluj-Napoca, 2007.
- [21] Harrow, A.W., Hassidim, A. and Lloyd, S., *Quantum algorithm for linear systems of equations*, Phys. Rev. Lett. 103 (2009) 150502.
- [22] Hosseini, M.M., *A modified pseudospectral method for numerical solution of ordinary differential equations systems*, Appl. Math. Comput. 176(2) (2006) 470–475.
- [23] Karmishin, A.V., Zhukov, A.I. and Kolosov, V.G., *Methods of dynamics calculation and testing for thin-walled structures*, Mashinostroyenie, Moscow, 1990.
- [24] Kierzenka, J. and Shampine, L.F., *A BVP solver based on residual control and the MATLAB PSE*, ACM Trans. Math. Software. 27 (2001) 299–316.
- [25] Krovi, H., *Improved quantum algorithms for linear and nonlinear differential equations*, Quantum 7 (2023) 913.

- [26] Kyriienko, O., Paine, A.E. and Elfving, V.E., *Solving nonlinear differential equations with differentiable quantum circuits*, Phys. Rev. A, 103(5) (2021) 052416.
- [27] Leyton, S.K. and Osborne, T.J., *A quantum algorithm to solve nonlinear differential equations*, arXiv:0812.4423 (2008).
- [28] Liao, S.J., *Beyond perturbation: Introduction to the homotopy analysis method*, Chapman and Hall/CRC Press, Boca Raton, 2003.
- [29] Liu, J.P., An, D., Fang, D., Wang, J., Low, G.H. and Jordan, S., *Efficient quantum algorithm for nonlinear reaction-diffusion equations and energy estimation*, Commun. Math. Phys. 404 (2023) 963–1020.
- [30] Lyapunov, A.M., *General problem on stability of motion* (English translation), Taylor and Francis, London, 1992.
- [31] Magyari, E., *Exact analytical solution of a nonlinear reaction-diffusion model in porous catalysts*, Chem. Eng. J. 143 (2008) 167–171.
- [32] Mirhosseini-Alizamini, S., *Solving linear optimal control problems of the time-delayed systems by Adomian decomposition method*, Iran. J. Numer. Anal. Optim. 9(2) (2019) 165–183.
- [33] Moitsheki, R.J., Hayat, T., Malik, M.Y. and Mahomed, F.M., *Symmetry analysis for the nonlinear model of diffusion and reaction in porous catalysts*, Nonlinear Anal. Real World Appl. 11 (2010) 3031–3036.
- [34] Nayfeh, A.H., *Perturbation methods*, John Wiley and Sons, New York, 2000.
- [35] Satterfield, C.N., *Mass transfer in heterogeneous catalysis*, MIT Press, Cambridge, 1970.
- [36] Shampine, L.F., Reichelt, M.W. and Kierzenka, J., *Solving boundary value problems for ordinary differential equations in MATLAB with bvp4c*, MATLAB File Exchange, 2004.
- [37] Shen, J., Tang, T., Wang, L.L., *Spectral methods: Algorithms, analysis and applications*, Springer, Berlin, 2011.

- [38] Srinivas, E., Lalu, M. and Phaneendra, K., *A numerical approach for singular perturbation problems with an interior layer using an adaptive spline*, Iran. J. Numer. Anal. Optim. 12(2) (2022) 355–370.
- [39] Thiele, E.W., *Relation between catalytic activity and size of particle*, Ind. Eng. Chem. 31 (1939) 916–920.
- [40] Trefethen, L.N., *Spectral methods in MATLAB*, Society for Industrial and Applied Mathematics (SIAM), 2000.
- [41] Zygelman, B., *A first introduction to quantum computing and information*, Springer Nature Switzerland AG, 2018.



Solving Bratu equations using Bell polynomials and successive differentiation

N.A. Gezer*, 

Abstract

This paper uses transformations and recursive algebraic equations to obtain series expansions, utilizing Bell polynomials, to solve the one-dimensional Bratu problem and several Bratu-type equations. The central aim of this work is to compare this approach with the successive differentiation method (SDM) by using computer routines for the computation of Bell polynomials. The series expansion method is applied to these nonlinear ordinary differential equations, and the various aspects of computation are compared with those obtained by the SDM. The former method is effective in handling nonlinearity, especially those arising from exponential terms, and the complexity of computations involving exponentials is handled by readily available computer routines for Bell polynomials. On the other hand, the SDM needs to handle these complexities with each differentiation.

AMS subject classifications (2020): Primary 65L05; Secondary 65L10.

*Corresponding author

Received 8 May 2025; revised 21 August 2025; accepted 26 August 2025

Niyazi Anıl Gezer

Department of Mathematics, Faculty of Arts and Sciences, TED University, 06420, Ankara, Turkey. e-mail: anilgezer@gmail.com

How to cite this article

Gezer, N.A., Solving Bratu equations using Bell polynomials and successive differentiation. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1482–1497.
<https://doi.org/10.22067/ijnao.2025.93423.1644>

Keywords: Successive differentiation method; Bratu equation; Computational analysis.

1 Introduction

Wazwaz [16] introduced the successive differentiation method (SDM) for solving various types of ordinary differential equations (ODEs). To obtain the series solution of an initial value problem by using the SDM, one differentiates the associated ODE and evaluates the obtained derivatives by using the initial values. In the case of boundary value problems, one further uses the boundary values to determine a series solution for the problem. In [16], SDM was used to obtain series solutions for the Bratu equation; see [3], and a variety of Bratu-type equations; see [16, 14].

The Bratu boundary value problem, which is a one-dimensional version of the classical Bratu problem, see [3, 12, 10], is given by

$$\begin{aligned} u'' + \lambda e^u &= 0, & 0 < x < 1, \\ u(0) = u(1) &= 0, \end{aligned} \tag{1}$$

where λ is a parameter. The Bratu problem appears in the mathematical models of certain engineering problems including the fuel ignition model of thermal combustion, and radiative heat transfer; see [9, 12, 10, 13] and the references therein.

Following [16], we are further interested in Bratu type-I, type-II, and type-III equations. These boundary value problems can be expressed as

$$\begin{aligned} u'' - \pi^2 e^u &= 0, & 0 < x < 1, \\ u(0) = u(1) &= 0, \end{aligned} \tag{2}$$

$$\begin{aligned} u'' + \pi^2 e^{-u} &= 0, & 0 < x < 1, \\ u(0) = u(1) &= 0, \end{aligned} \tag{3}$$

and

$$\begin{aligned} u'' - e^u &= 0, & 0 < x < 1, \\ u(0) &= u(1) = 0, \end{aligned} \tag{4}$$

respectively. We refer the reader to [16] for detailed discussions of these equations.

A method for finding series solutions to differential equations was used by Zhou [18] in 1986 to solve linear and non-linear initial value problems in electric circuits. In 1996, Chen and Ho [5, 6] solved eigenvalue problems for the free and transverse vibration problems of a rotating twisted Timoshenko beam under axial loading. Many researchers have used series expansions to investigate various problems.

Several techniques have been used to solve the Bratu problem. In [15], an Adomain decomposition method (ADM) was introduced in a framework to determine the exact solutions of Bratu-type equations. In [4], the series expansions were used to solve a particular case of a Bratu-type equation. In [10], similar methods were used for the Bratu boundary value problem. The method of weighted residual was used by Aregbesola to show the existence and multiplicity of solutions to the Bratu problem [1]. A method combining the Adomain decomposition method and the Laplace transforms was used by Syam and Hamdan to solve the Bratu equation [13]. In [9], high-order compact finite difference methods were used to numerically solve one-dimensional Bratu-type equations, and the analysis of convergence and their numerical rate of convergence were given. For the results related to other methods, see [2, 11, 12, 13].

Some of these methods can be further used to obtain series solutions of certain types of differential-algebraic systems whose solution is a vector-valued function admitting an analytical expansion with respect to a real variable. In this direction, we refer the reader to [8].

The structure of the present paper is similar to that of [16] and [4]. However, we use the notations of [17]. In Section 2, we present a lemma that allows us to compute transformations of nonlinear terms using Bell polynomials. The main tool, Lemma 1, utilizes Bell polynomials for this computational purpose. The use of Faà di Bruno's formula and Bell polynomials provides a method for the series expansion of composite nonlinear functions, as seen

in [7], and will be applied throughout the subsequent sections. In Subsections 2.1 and 2.2, we focus on obtaining transformations for general first-order and second-order ODEs. In section 3, we address the Bratu boundary value problem, where a computational comparison with the SDM is provided to analyze the efficiency of the proposed method. Finally, in sections 4, 5, and 6, we respectively study the Bratu type-I, type-II, and type-III equations.

2 Series expansion using bell polynomials

In this section, we follow the conventions given in [17]. A general theory that is very close to the present discussion can be found in [4].

Let $u(x)$ be an analytic function in a domain D containing zero. The analytic expansion of the function $u(x)$ about an ordinary point x_i is of the form

$$u(x) = \sum_{s=0}^{\infty} \frac{(x - x_i)^s}{s!} \left(\frac{d^s u(x)}{dx^s} \right)_{x=x_i} \quad (5)$$

for x belonging to the domain D . Following [18, 17, 4], we put

$$U(s) = \frac{\mathcal{H}^s}{s!} \left(\frac{d^s u(x)}{dx^s} \right)_{x=0} \quad (6)$$

for $s \geq 0$, where $\mathcal{H} \neq 0$ is a constant. To simplify equations, we also write subscripted U_s for $U(s)$.

As an operator, the transformation mapping an analytic function $u(x)$ to $U(s)$ is linear, but certainly not invertible on the space of real sequences. An inversion formula transforming such $U(s)$ to an analytic function $u(x)$ is given by

$$u(x) = \sum_{s=0}^{\infty} \left(\frac{x}{\mathcal{H}} \right)^s U(s), \quad (7)$$

which allows the reconstruction of the original function $u(x)$ from $U(s)$ for $x \in D$.

In the following lemma, instead of $\exp(u(x))$, we write $e^{u(x)}$. It describes the transformation of the exponential $e^{u(x)}$ of an analytic function $u(x)$ on D , in terms of the transformation of the function $u(x)$ itself, using Bell polynomials.

Bell polynomials, denoted as $B_{s,k}(x_1, x_2, \dots, x_{s-k+1})$, are a family of polynomials that appear in various combinatorial problems. These polynomials provide a way to express the derivative of a composite function in terms of the derivatives of the individual functions; see [7].

Lemma 1. The transformation of $e^{u(x)}$ is given by

$$N(s) = \frac{\mathcal{H}^s e^{U_0}}{s!} \sum_{k=1}^s B_{s,k} \left(\frac{U_1}{\mathcal{H}}, \frac{2U_2}{\mathcal{H}^2}, \frac{6U_3}{\mathcal{H}^3}, \dots, \frac{(s-k+1)!U_{s-k+1}}{\mathcal{H}^{s-k+1}} \right) \quad (8)$$

for $s \geq 0$, where $B_{s,k}(x_1, x_2, \dots, x_{s-k+1})$ denotes the Bell polynomials.

The proof of Lemma (1) follows from Faà di Bruno's formula, which gives a formula for the n -th derivative of the composition of two functions. Given the fact that there are well-developed computer routines available for the computation of Bell polynomials, Lemma (1) is very useful when solving nonlinear differential equations, particularly those involving exponential terms, such as the Bratu equation and Bratu-type equations. We remark that when $s = 0$ the equality (8) appearing in Lemma 1 should be understood as $N(0) = e^{U_0}$.

In the view of Lemma 1, the first few terms of the transformation of $e^{u(x)}$ can be written in terms of U_s as

$$\begin{aligned} N(0) &= e^{U_0}, \\ N(1) &= e^{U_0} U_1, \\ N(2) &= \frac{e^{U_0}}{2!} (U_1^2 + 2U_2), \\ N(3) &= \frac{e^{U_0}}{3!} (U_1^3 + 6U_1 U_2 + 6U_3), \\ N(4) &= \frac{e^{U_0}}{4!} (U_1^4 + 12U_1^2 U_2 + 24U_1 U_3 + 12U_2^2 + 24U_4), \\ N(5) &= \frac{e^{U_0}}{5!} (U_1^5 + 20U_1^3 U_2 + 60U_1^2 U_3 + 60U_1 U_2^2 + 120U_1 U_4 + 120U_2 U_3 + 120U_5), \end{aligned}$$

which demonstrates the nonlinear dependence of $N(s)$ on U_0, U_1, \dots, U_s . Furthermore, the equations for $N(s)$ show a recursive pattern; also, see [4, Eq.11]. Each subsequent term $N(s)$ in the sequence depends on U_0, U_1, \dots, U_n .

2.1 The first-order ODEs

Following [16], we start by investigating the first order ODE

$$u'(x) - f(x)u(x) = g(x), u(0) = \alpha_0, \quad (9)$$

where the functions $f(x)$ and $g(x)$ are analytic on a domain containing zero. Equation (9) represents a general first-order linear ODE. We denote by $U(s)$, $F(s)$ and $G(s)$ the transformations of the functions $u(x)$, $f(x)$ and $g(x)$, respectively. The transformation of (9) can be written as

$$\frac{s+1}{\mathcal{H}}U(s+1) - \frac{1}{\mathcal{H}^s} \sum_{k=0}^s F(s)U(s-k) = G(s) \quad (10)$$

for $s \geq 0$ with $U(0) = \alpha_0$. This transformation converts the differential equation into an algebraic equation. Rewriting (10) in the form

$$U(s+1) = \frac{\mathcal{H}}{s+1} \left[G(s) + \frac{1}{\mathcal{H}^s} \sum_{k=0}^s F(k)U(s-k) \right], \quad (11)$$

we obtain a recursive formula for $U(s+1)$. All values of $U(s+1)$ for $s \geq 0$ are completely determined by $U(0) = \alpha_0$. For instance, $U(1) = \mathcal{H}G(0) + \mathcal{H}F(0)\alpha_0$.

When we compare the SDM, see [16, Sec. 2.1], with the above result we see that the required number of differentiations and derivative evaluations are asymptotically equal to each other in both methods. The SDM involves differentiating the ODE and evaluating the derivatives at $x = 0$ to obtain a series solution. In the above, we use a transformation to convert the ODE into an algebraic equation which can then be solved recursively. We note that formula (11) is expressed in a concise mathematical form. The compactness of formula (11) simplifies the implementation of the method for computational purposes, as the number of steps needed to arrive at a solution is reduced. Therefore, the compact formula (11) can be further utilized for computational purposes. In addition, it is effective in handling nonlinearity, especially those arising from exponential terms, and the complexity of computations involving exponentials is handled by readily available com-

puter routines for Bell polynomials. On the other hand, the SDM needs to handle these complexities with each differentiation.

2.2 The second order ODEs

We investigate the second-order ODE

$$u''(x) - f(x)u'(x) - h(x)u(x) = g(x), u(0) = \alpha_0, u'(0) = \alpha_1, \quad (12)$$

where the function $h(x)$ is analytic on a domain containing zero. We denote by $H(s)$ the transformation of the function $h(x)$. Similar to the previous analysis, the transformation of (12) is the algebraic equation

$$\begin{aligned} \frac{(s+1)(s+2)}{\mathcal{H}^2} U(s+2) - \frac{1}{\mathcal{H}^{s+1}} \sum_{k=0}^s (s-k+1) F(k) U(s-k+1) \\ - \frac{1}{\mathcal{H}^s} \sum_{k=0}^s H(k) U(s-k) = G(s) \end{aligned} \quad (13)$$

for $s > 0$ with $U(0) = \alpha_0$ and $U(1) = \mathcal{H}\alpha_1$. Equation (13) can be rearranged to find a recursive formula for U_{s+2} as

$$U_{s+2} = \frac{\mathcal{H}^2}{(s+1)(s+2)} \left[G_s + \frac{1}{\mathcal{H}^s} \sum_{k=0}^s H_k U_{s-k} + \frac{1}{\mathcal{H}^{s+1}} \sum_{k=0}^s (s-k+1) F_k U_{s-k+1} \right] \quad (14)$$

for $s \geq 0$. When $s = 0$, this equality reduces to

$$U(2) = \frac{\mathcal{H}^2}{2} G(0) + \frac{\mathcal{H}}{2} H(0)^2 U(0)^2,$$

and hence, $U(0)$ and $U(1)$ determine $U(s+2)$ for $s \geq 0$ in the sense that (14) allows the computation of all $U(s)$ values for $s \geq 0$ using the initial values.

A series approximation to the solution of (12) can be obtained from the inversion formula (7) after evaluating $U(s+2)$ from (14). When we compare [16, Eq. 17] with (14) we see that the required number of derivatives and derivative evaluations are asymptotically equal to each other in both methods. However, the above method uses a recursive formula and avoids

repeated differentiation of the original equation, which can be an advantage. Equation (14) is compact and avoids repeated differentiation of the original equation. In addition, the complexity of computations involving exponentials is handled by readily available computer routines for Bell polynomials.

3 The Bratu boundary value problem

The Bratu boundary value problem is given as

$$\begin{aligned} u'' + \lambda e^u &= 0, & 0 < x < 1, \\ u(0) &= u(1) = 0 \end{aligned} \quad (15)$$

where the parameter λ is a constant. It follows from (15) and Lemma 1 that

$$U_{s+2} = \frac{-\lambda \mathcal{H}^{s+2} e^{U_0}}{(s+2)!} \sum_{k=1}^s B_{s,k} \left(\frac{U_1}{\mathcal{H}}, \frac{2U_2}{\mathcal{H}^2}, \frac{6U_3}{\mathcal{H}^3}, \dots, \frac{(s-k+1)! U_{s-k+1}}{\mathcal{H}^{s-k+1}} \right) \quad (16)$$

for $s \geq 0$ together with $U(0) = 0$. Hence, the transformation yields a recursive formula for U_{s+2} that involves the parameter λ , the constant \mathcal{H} , and the Bell polynomials. We note that Bell polynomials help in handling the nonlinearity introduced by the exponential term e^u .

Now, let us compute the first few terms of $U(s)$. For $s = 0$, we have

$$U(2) = \frac{-\lambda \mathcal{H}^2 e^{U_0}}{2!} = \frac{-\lambda \mathcal{H}^2}{2!}.$$

From the previous analysis we know that the values of $U(s)$ for $s \geq 1$ depend further on $U(1)$. Hence we have

$$\begin{aligned} U_3 &= -\frac{1}{6} \mathcal{H}^2 \lambda e^{U_0} U_1 = -\frac{1}{6} \mathcal{H}^2 \lambda U_1, \\ U_4 &= -\frac{1}{24} \mathcal{H}^4 \lambda e^{U_0} \left(\frac{U_1^2}{\mathcal{H}^2} + \frac{2U_2}{\mathcal{H}^2} \right) = \frac{1}{24} \mathcal{H}^2 \lambda (\mathcal{H}^2 \lambda - U_1^2), \\ U_5 &= -\frac{1}{120} \mathcal{H}^5 \lambda e^{U_0} \left(\frac{U_1^3}{\mathcal{H}^3} + \frac{6U_2 U_1}{\mathcal{H}^3} + \frac{6U_3}{\mathcal{H}^3} \right) = \frac{1}{120} \mathcal{H}^2 \lambda (4\mathcal{H}^2 \lambda U_1 - U_1^3). \end{aligned}$$

The inversion formula (7) can be used to approximate the solution $u(x)$ of (15). To see this, we first obtain

$$u^{(s)}(0) = \frac{s!U(s)}{\mathcal{H}^s}$$

from formula (7). Let $\mathcal{H} = 1$ and $u'(0) = \alpha$; see [16, Sec. 3]. Hence, we obtain

$$\begin{aligned} u(0) &= 0, \\ u'(0) &= \frac{U(1)}{\mathcal{H}} = \alpha, \\ u''(0) &= \frac{-\lambda\mathcal{H}^2}{2!} = \frac{-\lambda}{2!}, \\ u'''(0) &= \frac{3!U(3)}{\mathcal{H}^3} = -\alpha\lambda, \\ u^{(4)}(0) &= \frac{4!U(4)}{\mathcal{H}^4} = \lambda^2 - \lambda\alpha^2, \\ u^{(5)}(0) &= \frac{4!U(4)}{\mathcal{H}^4} = 4\lambda^2\alpha - \lambda\alpha^3, \end{aligned}$$

for the values of derivatives of $u(x)$ at $x = 0$. The resulting series approximation to $u(x)$ is

$$u(x) = \alpha x - \frac{\lambda}{2!}x^2 - \frac{\alpha\lambda}{3!}x^3 - \frac{\alpha^2\lambda - \lambda^2}{4!}x^4 - \frac{\alpha^3\lambda - 4\lambda^2\alpha}{5!}x^5 + \dots,$$

which is equal to the series approximation given in [16, Eq. 21]. The above series approximation matches the one obtained by SDM. Therefore the obtained result is consistent with approximate solutions of the Bratu boundary value problem.

The computation of α can be found in [16]. We note that neither the SDM nor the above method obtains the value of α from the boundary condition $u(1) = 0$ in a direct manner. Furthermore, neither of these methods distinguishes the critical value of λ for which the Bratu problem has no solution. Both methods generate a series solution, but the solution may be valid for a limited range of λ , or may not converge if λ is too large.

4 The Bratu-type equation I

The Bratu-type I equation is

$$\begin{aligned} u'' - \pi^2 e^u &= 0, & 0 < x < 1, \\ u(0) &= u(1) = 0. \end{aligned} \quad (17)$$

This equation is a variation of the standard Bratu problem, and it is one of three Bratu-type equations that have been examined in the literature.

In this case, $U(0) = 0$. For $s \geq 3$, $U(s)$ can be expressed in terms of $U(1)$, for instance,

$$\begin{aligned} U_0 &= 0, \\ U_1 &= \alpha \mathcal{H}, \\ U_2 &= \frac{\pi^2 \mathcal{H}^2}{2!}, \\ U_3 &= \frac{\pi^2 \mathcal{H}^2}{3!} U_1 = \frac{\pi^2 \mathcal{H}^3}{3!} \alpha, \\ U_4 &= \frac{\pi^2 \mathcal{H}^2}{4!} (U_1^2 + 2U_2) = \frac{\pi^2 \mathcal{H}^4}{4!} (\pi^2 + \alpha^2), \\ U_5 &= \frac{\pi^2 \mathcal{H}^2}{5!} (U_1^3 + 6U_1 U_2 + 6U_3) = \frac{\pi^2 \mathcal{H}^5}{5!} (4\pi^2 + \alpha^2) \alpha. \end{aligned} \quad (18)$$

All the remaining terms can be obtained from the formula (16). Higher-order terms $U(s)$ for $s \geq 3$ are expressed in terms of previous terms, most importantly in terms of $U(1)$. Now, let us compute the values of the derivative of $u(x)$ at zero. We have

$$\begin{aligned} u'(0) &= \frac{U_1}{\mathcal{H}} = \alpha, \\ u''(0) &= \frac{2!U_2}{\mathcal{H}^2} = \pi^2, \\ u'''(0) &= \frac{3!U_3}{\mathcal{H}^3} = \pi^2 \alpha, \\ u^{(4)}(0) &= \frac{4!U_4}{\mathcal{H}^4} = \pi^2 (\pi^2 + \alpha^2), \\ u^{(5)}(0) &= \frac{5!U_5}{\mathcal{H}^5} = \pi^2 (4\pi^2 + \alpha^2) \alpha, \end{aligned}$$

all of which follow from (18). These derivatives are consistent with those derived using the SDM. Hence, the series approximation of $u(x)$ is given by

$$u(x) = \alpha x + \frac{\pi^2}{2!}x^2 + \frac{\pi^2\alpha}{3!}x^3 + \frac{\pi^2(\pi^2 + \alpha)}{4!}x^4 + \frac{\pi^2(4\alpha\pi^2 + \alpha^3)}{5!}x^5 + \dots,$$

which is equal to the series approximation obtained in [16, Eq. 27]. We note that the computation of α can be found in [16]. The value of α which is equal to $u'(0)$, is not directly computed by either of the methods but can be found by applying the boundary condition $u(1) = 0$ to the series approximation.

5 The Bratu-type equation II

The Bratu-type II equation is given by

$$\begin{aligned} u'' + \pi^2 e^{-u} &= 0, & 0 < x < 1, \\ u(0) &= u(1) = 0. \end{aligned} \tag{19}$$

Observe that $-u(x)$ appears as an exponent. We can still use Lemma 1 to find the transformation of $e^{-u(x)}$. Indeed, let $w(x) = e^{-u(x)}$ and denote by $W(s)$ the transformation of $w(x)$. It follows that

$$W(s) = \frac{\mathcal{H}^s e^{-U_0}}{s!} \sum_{k=1}^s B_{s,k} \left(-\frac{U_1}{\mathcal{H}}, -\frac{2U_2}{\mathcal{H}^2}, -\frac{6U_3}{\mathcal{H}^3}, \dots, -\frac{(s-k+1)!U_{s-k+1}}{\mathcal{H}^{s-k+1}} \right), \tag{20}$$

which is expressed in terms of the values of U_s . In other words, we can obtain the transformation of $e^{-u(x)}$ from the formula given in Lemma 1 by replacing every occurrence of U_s with its negative. It follows that for (19) we have

$$U_{s+2} = \frac{-\pi^2 \mathcal{H}^{s+2} e^{-U_0}}{(s+2)!} \sum_{k=1}^s B_{s,k} \left(-\frac{U_1}{\mathcal{H}}, -\frac{2U_2}{\mathcal{H}^2}, -\frac{6U_3}{\mathcal{H}^3}, \dots, -\frac{(s-k+1)!U_{s-k+1}}{\mathcal{H}^{s-k+1}} \right) \tag{21}$$

for $s \geq 0$.

At this point, an informal remark regarding the sensitivity to changes in the ODE is being addressed. When we compare with the SDM of [16], we may conclude that the above method has the disadvantage that the transformation

of an ODE may change drastically under slight changes of the ODE whereas SDM of [16] provides a uniform scheme in such situations. By the uniform scheme, we mean that SDM's approach to solving ODEs is more consistent and less prone to drastic changes in the solution process when the ODE is altered. However, if s is large enough, the computer routines related to Bell polynomials may balance the computation.

It follows from (21) that

$$\begin{aligned} U_0 &= 0, \\ U_1 &= \alpha\mathcal{H}, \\ U_2 &= \frac{-\pi^2\mathcal{H}^2}{2!}, \\ U_3 &= \frac{\pi^2\mathcal{H}^2}{3!}U_1 = \frac{\pi^2\mathcal{H}^3}{3!}\alpha, \\ U_4 &= -\frac{\pi^2\mathcal{H}^2}{4!}(U_1^2 - 2U_2) = -\frac{\pi^2\mathcal{H}^4}{4!}(\pi^2 + \alpha^2), \\ U_5 &= \frac{\pi^2\mathcal{H}^2}{5!}(U_1^3 - 6U_1U_2 + 6U_3) = \frac{\pi^2\mathcal{H}^5}{5!}(4\pi^2 + \alpha^2)\alpha. \end{aligned}$$

By using these, we obtain the values of the derivative of $u(x)$ at zero. In detail, we have

$$\begin{aligned} u'(0) &= \frac{U_1}{\mathcal{H}} = \alpha, \\ u''(0) &= \frac{2!U_2}{\mathcal{H}^2} = -\pi^2, \\ u'''(0) &= \frac{3!U_3}{\mathcal{H}^3} = \pi^2\alpha, \\ u^{(4)}(0) &= \frac{4!U_4}{\mathcal{H}^4} = -\pi^2(\pi^2 + \alpha^2), \\ u^{(5)}(0) &= \frac{5!U_5}{\mathcal{H}^5} = \pi^2(4\pi^2 + \alpha^2)\alpha. \end{aligned}$$

Hence, the series approximation of $u(x)$ is given by

$$u(x) = \alpha x - \frac{\pi^2}{2!}x^2 + \frac{\pi^2\alpha}{3!}x^3 - \frac{\pi^2(\pi^2 + \alpha)}{4!}x^4 + \frac{\pi^2(4\alpha\pi^2 + \alpha^3)}{5!}x^5 + \dots,$$

which is identical to the series approximation obtained using the series approximation obtained in [16, Eq. 34]. When we compare the SDM with the

above result we see that the required number of derivatives and derivative evaluations are asymptotically equal to each other in both methods.

6 The Bratu-type equation III

The Bratu-type III equation is given by

$$\begin{aligned} u'' - e^u &= 0, & 0 < x < 1, \\ u(0) &= u(1) = 0. \end{aligned} \tag{22}$$

Since (16) depends linearly on λ , the computation of $U(s)$ for (22) is similar to that of the Bratu type-I equation. In detail, values of $U(s)$ for the Bratu type-III equation can be obtained from (18) by multiplying both sides by $1/\pi^2$. Hence, in the present case, we have

$$\begin{aligned} U_0 &= 0, \\ U_1 &= \alpha \mathcal{H}, \\ U_2 &= \frac{\mathcal{H}^2}{2!}, \\ U_3 &= \frac{\mathcal{H}^2}{3!} U_1 = \frac{\mathcal{H}^3}{3!} \alpha, \\ U_4 &= \frac{\mathcal{H}^2}{4!} (U_1^2 + 2U_2) = \frac{\mathcal{H}^4}{4!} (1 + \alpha^2), \\ U_5 &= \frac{\mathcal{H}^2}{5!} (U_1^3 + 6U_1U_2 + 6U_3) = \frac{\mathcal{H}^5}{5!} (4 + \alpha^2)\alpha. \end{aligned}$$

All the remaining terms can be obtained from the formula (16) similarly. Let us compute the values of the derivative of $u(x)$ at zero. We have

$$\begin{aligned} u'(0) &= \frac{U_1}{\mathcal{H}} = \alpha, \\ u''(0) &= \frac{2!U_2}{\mathcal{H}^2} = 1, \\ u'''(0) &= \frac{3!U_3}{\mathcal{H}^3} = \alpha, \\ u^{(4)}(0) &= \frac{4!U_4}{\mathcal{H}^4} = (1 + \alpha^2), \\ u^{(5)}(0) &= \frac{5!U_5}{\mathcal{H}^5} = (4 + \alpha^2)\alpha. \end{aligned}$$

Hence, the series approximation of $u(x)$ is given by

$$u(x) = \alpha x + \frac{1}{2!}x^2 + \frac{\alpha}{3!}x^3 + \frac{(1 + \alpha^2)}{4!}x^4 + \frac{\alpha(4 + \alpha^2)}{5!}x^5 + \cdots,$$

which is identical to the series approximation obtained using the series approximation obtained in [16, Eq. 38].

7 Conclusion

We used transformations and recursive algebraic equations to obtain the series of solutions to the Bratu problem and Bratu-type equations. In all cases, the obtained series solutions agree with the series solutions given in [16]. Because there are well-developed computer routines for the computation of Bell polynomials, it can be compared with the alternative method of successive differentiation method. We use Bell polynomials to handle the nonlinear terms and, with readily available computer routines for computing these polynomials. Neither methods do not directly compute the value of $u'(0)$ using the boundary condition $u(1) = 0$, but instead uses it to solve for the unknown value, α . The methods also do not distinguish a critical value of λ for which the Bratu problem has no solution.

Acknowledgements

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

References

- [1] Aregbesola, Y., *Numerical solution of Bratu problem using the method of weighted residual*, Electron. J. South. Afr. Math. Sci. Assoc. 3(1) (2003), 1–7.
- [2] Ascher, U.M. and Russell, R.D., *Numerical solution of boundary value*

- problems for ordinary differential equations*, SIAM, Philadelphia, PA, 1995.
- [3] Bratu, G., *Sur les équations intégrale non linéaires*, Bull. Soc. Math. France 42 (1914), 113–142.
 - [4] Chang, S.-H. and Chang, I.-L., *A new algorithm for calculating one-dimensional differential transform of nonlinear functions*, Appl. Math. Comput. 195(2) (2008), 799–808.
 - [5] Chen, C.K. and Ho, S.H., *Applications of differential transformation to eigenvalue problem*, J. Appl. Math. Comput. 79 (1996), 173–188.
 - [6] Chen, C.K. and Ho, S.H., *Transverse vibration of a rotating twisted Timoshenko beam under axial loading using differential transform*, Int. J. Mech. Sci. 41 (1999), 1339–1356.
 - [7] Chou, W.-S., Hsu, L.C. and Shiue, P.J.-S., *Application of Faà di Bruno's formula in characterization of inverse relations*, J. Comput. Appl. Math. 190(1-2) (2006), 151–169.
 - [8] Gezer, N.A., *An application of recurrence relations to central force fields*, Turk. J. Astron. Astrophys. 5(2) (2024), 13–21.
 - [9] Gharechahi, R., Ameri, M.A. and Bisheh-Niasar, M., *High order compact finite difference schemes for solving Bratu-type equations*, J. Appl. Comput. Mech. 5(1) (2019), 91–102.
 - [10] Hassan, I.H.A.H. and Erturk, V.S., *Applying differential transformation method to the one-dimensional planar Bratu problem*, Int. J. Contemp. Math. Sci. 2 (2007), 1493–1504.
 - [11] Khuri, S.A., *A new approach to Bratu's problem*, Appl. Math. Comput. 147 (2004), 131–136.
 - [12] Mohsen, A., *A simple solution of the Bratu problem*, Comput. Math. Appl. 67 (2014), 26–33.
 - [13] Syam, M.I. and Hamdan, A., *An efficient method for solving Bratu equations*, Appl. Math. Comput. 176 (2006), 704–713.

- [14] Wazwaz, A.M., *The modified decomposition method applied to unsteady flow of gas through a porous medium*, Appl. Math. Comput. 118(2-3) (2001), 123–132.
- [15] Wazwaz, A.M., *Adomian decomposition method for a reliable treatment of the Bratu-type equations*, Appl. Math. Comput. 166 (2005), 652–663.
- [16] Wazwaz, A.M., *The successive differentiation method for solving Bratu-type equations*, Rom. J. Phys. 61(5–6) (2016), 774–783.
- [17] Yu, L.T. and Chen, C.K., *The solution of the Blasius equation by the differential transformation method*, Math. Comput. Modelling 28 (1998), 101–111.
- [18] Zhou, J.K., *Differential transformation and its applications for electric circuits*, Huazhong Univ. Press, Wuhan, China (in Chinese), 1986.



Comparative evaluation of large-scale many objective algorithms on complex optimization problems

R. Chaudhary and A. Prajapati*, 

Abstract

In the field of optimization, there has been an enormous surge in interest in addressing large-scale many-objective problems. Numerous academicians and practitioners have contributed to evolutionary computation by developing a variety of optimization algorithms tailored to tackle computationally challenging optimization problems. Recently, various large-scale many-objective optimization algorithms (LSMaOAs) have been proposed to address complex large-scale many-objective optimization problems (LSMaOPs). These LSMaOAs have shown remarkable performance in addressing a variety of LSMaOPs. However, there is a pressing need to further investigate their performance in comparison to each other on

*Corresponding author

Received 14 December 2024; revised 9 July 2025; accepted 26 August 2025

Ritika Chaudhary

Department of Computer Science Engineering and Information Technology, Jaypee Institute of Information Technology Noida, India. e-mail: ritzchaudhary19@gmail.com

Amarjeet Prajapati

Department of Computer Science Engineering and Information Technology, Jaypee Institute of Information Technology Noida, India. e-mail: amarjeetnitkkr@gmail.com

How to cite this article

Chaudhary, R. and Prajapati, A., Comparative evaluation of large-scale many objective algorithms on complex optimization problems. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1498-1537. <https://doi.org/10.22067/ijnao.2025.91210.1569>

different classes of LSMaOPs. In this study, we conduct a comparative investigation of three established LSMaOAs namely, LMEA, LMOCSO and S3CMAES over rigorous benchmarking on DTLZ, LSMOP, UF9-10, WFG test suites, encompassing problem sets with three to ten objectives and varying numbers of variables between 100 and 500. Additionally, we assess the algorithm's efficacy on a test suite specifically designed for large-scale multi/many-objective problems (100-1000 decision variables). In addition, we propose Hybrid-LMEA, a light hybrid that integrates decision-variable clustering with competitive learning to improve both convergence and diversity. The hybrid works especially well on high-dimensional large-scale many-objective optimization problems with better performance in 8 and 12 out of 27 test cases for IGD and GD, respectively. The outcomes of the experiments indicate the relative efficacy and effectiveness of the different algorithms in addressing large-scale many-objective problems. Researchers can leverage this comparative data to make informed decisions about which algorithms to employ for particular optimization problem domains.

AMS subject classifications (2020): Primary 68T05, 68W50; Secondary 90C59, 90C90, 65Y20.

Keywords: Optimization problems; Large-scale multi-objective optimization; Large-scale many-objective optimization

1 Introduction

A large-scale many-objective problem (LSMaOP) refers to an optimization problem involving a significant number of decision variables ($D > 100$) and a considerable number of objectives ($M > 3$) [47, 35]. Such optimization problems frequently arise in real-world scenarios such as software package redesign [39], software module clustering [20], hybrid car controller design [13], and pickup and delivery logistics [52]. While several multi/many-objective optimization algorithms (MOA/MaOA) [18] have proven effective for addressing multi/many-objective optimization problems (MOPs/MaOPs) [40] with two or three objectives, they often struggle to strike a balance between convergence and diversity when faced with the extensive numbers of decision variables and objectives in LSMaOPs. The decision space, along with the

objective search space, grows exponentially, resulting in the curse of dimensionality [9].

Numerous researchers and practitioners have proposed various approaches in the literature to overcome scalability challenges imposed by LSMaOPs. These approaches include grouping decision variables [27], reducing the decision space [16], and introducing novel search techniques [43], all specifically developed to address the complexities inherent in LSMaOPs, such as conflicting objectives and complex interactions between decision variables [26]. Due to the poor scalability of conventional MOAs/MaOAs [24] on LSMaOPs, the first approach involves splitting decision variables into groups, possibly at random or using heuristic approaches, followed by optimizing each group individually. For instance, Ma et al. [30] proposed MOA/DVA, which divides decision variables into distance, position, and mixed variables based on convergence and diversity control properties, while Zhang et al. [50] suggested a large-scale many-objective evolutionary algorithm (LMEA), which partitions decision variables into groups based on convergence and diversity.

The second approach aims to address the high-dimensional decision space of LSMaOPs by reducing its volume. This can be achieved through dimensionality reduction [5] and problem transformation [53] strategies, as demonstrated by Zille et al. [54] with the weighted optimization framework (WOF), which focuses on grouping decision variables into subgroups and assigning weights to each subgroup. The third approach involves developing novel search techniques tailored to LSMaOPs, creating offspring within the actual decision space without reducing its dimensionality. These techniques may include probability models [43] and reproduction operators [43] designed to efficiently handle LSMaOPs and produce optimal results. For example, Chen et al. [10] proposed S3CMAES, a scalable small subpopulations-based covariance matrix adaptation evolution strategy utilizing probability-based methods. In addition to these approaches, other strategies [43] have been adopted in the literature to solve LSMaOPs. For instance, Zhang, Shen, and Yen [48] introduced LSMaODE, a multi-population-based large-scale many-objective differential algorithm, while Ma et al. [29] presented LSMOEAD, incorporating an adaptive localized decision variable strategy within a decomposition framework [32]. Xu et al. [46] have proposed multi-population based

MOEA, that is, DVCOEA, which involves analyzing the decision variable contribution to objectives, here decision variables are categorized depending on their contribution objectives. Using these approaches, LSMaOPs may be solved by tweaking/using traditional evolutionary algorithms [4], reducing the difficulties of the high-dimensional search space [42].

Despite these advancements, there remains a gap in the literature concerning the performance of large-scale many-objective optimization algorithms (LSMaOAs) on LSMaOPs with a significant number of decision variables. Previous studies by Deb et al. [17] and Cheng et al. [12] have focused on LSMaOP benchmarks with a limited number of objectives and decision variables. Apart from these, various other LSMaOAs have been proposed to address complex LSMaOPs. These LSMaOAs have shown remarkable performance in addressing a variety of LSMaOPs. However, there is a pressing need to further research on the investigation of their performance in comparison to each other over different classes of LSMaOPs. Thus, our study aims to fill this gap by providing insights into the comparative performance of LSMaOAs on LSMaOPs with more than three objectives and varying decision variables between 100-2000.

In this paper, we conduct a detailed comparative analysis of three different algorithms based on strategies from current literature: LMEA [50], large-scale multi-objective optimization based on competitive swarm optimizer (LMOCSSO) [17], and S3CMAES [10]. We evaluate the efficacy of these algorithms using benchmark test suites (DLTZ) [41] and LSMaOP test suites [31], employing the inverted generational distance (IGD) as our performance metric [55, 44]. In addition to this comparative evaluation, we introduce a new variant referred to as LMEA-hybrid, blending the decision variable grouping of LMEA with the competitive learning strategy of LMOCSSO. The hybridization is designed to improve the balance of convergence and diversity in high-dimensional search spaces. The algorithm is empirically compared with LMEA, LMOCSSO, and S3CMAES on benchmark sets DTLZ, LSMOP, UF, and WFG. The outcomes show that Hybrid-LMEA improves performance in respect to IGD, generation distance (GD), and HV significantly in all but a few test instances, particularly in problems involving over 500 decision variables.

The motivation behind the selection of algorithms for comparative analysis is guided by their different strategies to cope with large-scale many-objective optimization scenarios. LMEA is a population decomposition-based algorithm that divides decision variables into groups with similar convergence and diversity characteristics, a feature that greatly improves its performance against the curse of dimensionality. Conversely, LMOCSO employs swarm intelligence and competitive learning mechanisms to maintain a middle ground between convergence and diversity in high-dimensional space. Finally, S3CMAES uses a covariance matrix adaptation strategy with small subpopulations that allows it to explore complex search spaces effectively while remaining computationally inexpensive.

The three chosen algorithms are representative of different methodologies—an approach based on decomposition, one based on swarm intelligence, and an evolutionary strategy; thus, a diverse comparative analysis will be possible. Additionally, their adequate performance during previous studies makes them well-suited for evaluating LSMaOAs. To facilitate a relative comparison of the performance of various MOEAs, and understand their relative strengths and weaknesses, this study is centered around a set of such benchmark problems, intending to provide insights into which algorithm might be selected for many-objective optimization in large-scale dimensions.

The paper is further segmented into four different sections. In Section 2, the basic concepts of large-scale multi/many objective optimization (LSMOO) and Large-scale many-objective optimization (LSMaO) are discussed. Section 3 presents our experimental design. Section 4 demonstrates the results of experiments. Lastly in Section 5, the findings and recommendations for future research directions are laid out.

2 Related work

To tackle LSMaOPs, several optimization approaches have been proposed by customizing existing multi-objective optimization techniques. In the following subsection, we outline the major approaches based on the strategies of large-scale multi-objective optimization and large-scale many-objective optimization.

2.1 Large-scale multi-objective optimization

Over the last 20 years, MOEAs have shown their efficacy in dealing with MOPs. They are a useful method for handling MOPs since they are good at producing a large number of solutions in just one run. For this reason, numerous MOEAs have been proposed, namely SPEA2 [56], MOEA/D [49] and NSGA-II [15]. The majority of these algorithms are based on Pareto-based techniques, which select solutions that have greater Pareto ranks by using the concept of Pareto dominance. Moreover, diversity-related criteria are utilized to encourage a broad range of solutions dispersed throughout the Pareto front (PF). While Pareto-based strategies including SPEA2 and NSGA-II have shown potential in addressing MOPs that have two or three objectives however are ineffective when confronted with MaOPs. The primary objective of evolutionary multi-objective algorithms is to strike an equilibrium between two opposing objectives: significant diversity and effective convergence. The substantial diversity involves dispersing solutions extensively along the PF whereas effective convergence refers to a narrowing of gaps between solutions and the PF.

Large-scale MOPs/MaOPs present a substantial problem in optimization due to the presence of over one hundred decision variables. Maintaining a careful balance between diversity and convergence within an evolutionary algorithm's search process gets increasingly difficult as the quantity of non-dominated solutions rises significantly as the search space expands. Over the past few years, several evolutionary algorithms focusing on decision variable analysis have been presented [21] to solve these challenges. MOEA/DVA, a unique approach focusing on analyzing decision variables, was developed by Ma et al. [30]. This is a revolutionary MOEA that was created explicitly to address the challenges given by large-scale MOPs. This method groups decision variables into three categories based on their dominance relationships such as diversity-related, convergence-related, and both convergence and diversity. Similarly, Antonio and Coello [2] have proposed a cooperative co-evolutionary framework where decision variables are partitioned into numerous co-evolved sub-components to address the complexities of large-

scale/High-Dimensional MOPs. This framework has proven to be successful in handling significant number of decision variables (up to 5000).

Zille et al. [54] introduced a WOF. This method employs decision variable grouping and weighing to enhance the effectiveness of population-based algorithms for LSMOPs. This framework has been evaluated on NSGA-II and SMPSO algorithms with benchmark problems (WFG) having two and three objectives and with 1000 decision variables. He et al. [22] presented another generalized framework for LSMOO, that is, LSMOF. The primary objective of framework was to increase the multi-objective algorithm's computational efficiency on LSMOPs. The introduced framework LSMOF employs a two-stage approach. The first stage of this approach involves problem reformulation which is responsible for generating an array of optimal solutions close to the PS and the second stage involves spreading a uniformly distributed approximate Pareto set, the LSMOF employed candidate solution spreading techniques and problem reformulation. Without employing any decision variable analysis method or grouping technique, LSMOF demonstrates effective performance and computational efficiency as compared to the current strategies in the literature. Deb et al. [17] put forwarded a competitive swarm optimizer (CSO) called LMOCSSO for efficient search in LSMOPs. It involves particle updating strategy and competitive mechanism to enhance search efficiency. Hong et al. [23] proposed a model based on probabilistic prediction called LMOPPM. This model uses sampling and trend prediction models for effectively solving LSMOPs. Liu et al. [28] suggested an evolutionary algorithm based on dimensionality reduction and clustering for LSMOPs (dimensions up to 5000). The interdependence analysis has been employed to partition convergence variables into subgroups.

2.2 Large-scale many-objective optimization

Zhang et al. [50] presented LMEA, a decision variable-based strategy for LSMaOPs. Using the convergence and diversity properties as a basis, this method groups decision variables. The statistical findings demonstrate how well LMEA handles large-scale MaOPs involving as many as 5000 decision

variables. Xu et al. [46]. presented a novel technique for LSMOO/LS-MaO termed DVCOEA. The primary goal of this is to increase convergence and diversity by classifying decision variables according to their objectives of contribution. The suggested methodology may find application in various machine learning techniques and evolutionary algorithms for estimating fitness.

Cao et al. [8] proposed distributed parallel PSO algorithms. This paper emphasizes that new PSO algorithms are required to handle LSMaOPs. This also highlights how crucial distributed parallel computing is to cutting down on operation time. Gu and Wang [19] introduced IFM-NSGA-III algorithms, which improve NSGAIII by using information feedback models for LSMOPs and LSMaOPs. The study highlights how important information feedback models are for MOEA optimization. Zhang et al. [51] proposed MOEA/D-IFM which includes information feedback models to enhance performance. The paper emphasizes using population information in optimizing algorithms. Wang et al. [45] proposed 1EA-IFM framework for LSMOPs. This framework balances diversity and convergence by retaining historical information and using fitness function values. Cao et al. [7] presented LS-MaOEA to overcome the drawback of RVEA [11] in handling problems which are large-scale evolutionary algorithm.

Table 1 showcases a range of many-objective and large-scale many-objective optimization approaches, highlighting how various methods have been developed by customizing existing optimization algorithms. Meanwhile, Table 2 provides brief descriptions of various comparative studies conducted by previous researchers.

3 Experiment design

We empirically assess the efficacy of three algorithms, that is, LMEA [50], LMOCSO [12] and S3CMAES [10] by conducting experiments on a well-established set of benchmark problems, namely, the DTLZ test suites [41] and LSMOP test suite [31]. All the simulations were conducted using MATLAB R2023a environment on a Macbook Air powered by macOS, equipped with

Table 1: Many-objective and large-scale many-objective Approaches

| Application | Scale | Solution Technique | Algorithm | Modified rithm | Algo- | Problems |
|--|---|--|-----------------------------|--|-------|-----------------------|
| Software module cluster- ing [1] | More than four ob- jective functions | Clustering-based ap- proach | ABC | MaABC | | MaSMCP |
| Software package restruc- turing Problem [36] | nine objective func- tions and more than 100 decision variables | Dominance-based ap- proach | PSO | Customized PSO | | LSMaOSPR |
| Software architecture re- covery [37] | Decision variables between 100-991 and objective func- tions range 3-7 | Search-based optimization approaches | PSO | LSM-PSO | | LSMaO-SAR problems |
| Software module cluster- ing [38] | More than six ob- jective functions | Grid-based approach | PSO | GLMPSO | | LMSMCPs |
| Optimization of Edge Servers (ESs) deployment [6] | 5 objective func- tions | Clustering algorithms and evolutionary algorithms | PSO | PCMaLIA | | MaODES |
| Food-energy-water nexus [33] | five objective func- tions, more than 100 decision vari- ables | Search-based approach, Dimensionality reduction | LCSA | Modified LCSA | | FEWN |
| Optimizing electric vehi- cle (EV) charging and dis- charging schedules [34] | Decision variables more than 480 and four objective functions | Preference-based ap- proach | Coevolutionary algorithm | Preference inspired coevolutionary algorithm | | PICEAg-EV |

Table 2: Summary of comparative results of various algorithms

| Algorithm | Test Suite | Decision Variables | Objectives | Performance Metric | Runs |
|---|----------------------|--------------------|------------|--------------------|------|
| LS-SMS-MOEA, LS-NSGAI, LS-MOEA/D, etc. [29] | LSMOP(1-9) | 1000 | 3 | IGD | 20 |
| LCSA, LMOC SO, S3CMAES, etc. [14, 25] | LSMOP(1-9), MaOPFEWN | 315 | 5 | GD, IGD, HV | 30 |
| MOEA/D, NSGA-III, KnEA, etc. [50] | DLTZ(1-9) | 100, 500, 1000 | 5, 10 | IGD | 20 |
| MOEA/D, NSGA-III, KnEA, etc. [50] | WFG 3 | 100, 500, 1000 | 5, 10 | IGD | 20 |
| MOEA/D, NSGA-III, KnEA, etc. [50] | UF(9,10) | 100, 500, 1000 | 3 | IGD | 20 |
| LMEA [50] | DLTZ(1-7) | 2000, 5000 | 5, 10 | HV | 20 |
| MOEA/D, NSGA-III, KnEA, etc. [50] | LSMOP(1-9) | 500 | 5 | IGD | 20 |
| DVCOEA/ DVCOEA(NO-CO) [46] | DLTZ(1-7) | 100, 500 | 5 | IGD | 20 |
| DVCOEA/DVCOEA(NO-CO) [46] | LSMOP(1-9) | 112 | 3 | IGD | 20 |
| DVCOEA, DVCOEA(NO-CO) [46] | UF(3-8) | 100, 500 | 2 | IGD | 20 |
| DVCOEA, DVCOEA(NO-CO) [46] | UF(9-10) | 100, 500 | 3 | IGD | 20 |
| DVCOEA, MOEA/DVA, LMEA, CPSO [46] | UF(1-7) | 100, 500 | 2 | IGD | 20 |
| DVCOEA, MOEA/DVA, LMEA, CPSO [46] | UF(8-10) | 100, 500 | 3 | IGD | 20 |
| DVCOEA, MOEA/DVA, LMEA, CPSO [46] | DLTZ(1-7) | 100, 500 | 5 | IGD | 20 |
| DVCOEA, MOEA/DVA, LMEA, CPSO [46] | LSMOP(1-9) | 514 | 5 | IGD | 20 |
| LSMaODE, MOEA/DVA, LMOC SO, S3CMAES [48] | LSMOP(1-9) | 300, 500, 1000 | 3 | IGD | 30 |

an M1 Apple chip, 8 GB memory, 7 core Graphics processing unit, 8 core Central processing unit and 256 GB Solid state drive.

3.1 Test problems

Previous researchers and practitioners have developed various test suites to assess the performance of optimization problems. These test suites encompass a range of complexities and difficulties. In this work, to compare the optimization approaches, we use the following two test suites: **1) DTLZ** -Riquelme, von Lücken, and Baran [41] introduced test suite (DTLZ). The problems of this suite are flexible to meet wide range of objectives. The PF is represented by the initial M-1 decision variables, whereas the remaining variables are related to the convergence characteristic. In particular, the DTLZ suite has several distinguishing features. DTLZ1 and DTLZ3 include many local PFs. In DTLZ4, Pareto optimum solutions are not evenly distributed, and in DTLZ5 and DTLZ6, PF curves are degenerated. In DTLZ7, PF is broken, whereas DTLZ8 and DTLZ9 consist of constrained test problems. The DTLZ test suite makes a significant contribution by proposing a general design paradigm for creating test problems that are extended to accommodate a broad range of objectives and decision variables. **2) LSMOP**- The test suite comprises nine problems tailored for LSMOO and LSMaO. Across these problems, the Pareto Front (PF) characteristics vary: LSMOP1-4 features linear PFs, LSMOP5-8 exhibits nonlinear PFs, and LSMOP9 presents a disconnected PF. **3) WFG** [12]- Huband et al. proposed the WFG test suite in 2006. This test suite contains several many-objective test cases, which are characterised by their specific features. The PF of WFG test suite test cases have a different type of shapes. Therefore, the WFG test suite would accurately reflect the performance of convergence and diversity algorithms. **4) UF** [12]- This test suite contains ten unconstrained test problems UF1-10. In our study, we have considered only two problems UF9 and UF10 for comparative analysis.

3.2 Algorithms in comparison

The selection of the LMEA, LMOCSO, and S3CMAES algorithms for conducting the comparative study was based on several factors: 1) relevance to the problem domain, 2) diversity in approaches, and 3) scalability and performance. By selecting LMEA, LMOCSO, and S3CMAES for our comparative study, we aim to provide valuable insights into their relative strengths and weaknesses in addressing the challenges posed by large-scale many-objective optimization problems.

- **LMEA**- A novel method that addresses the challenges of LSMaOPs is the clustering of a decision variable based evolutionary algorithm (LMEA). This method provides a decision variable clustering technique that classifies variables based on their properties of convergence and diversity. This method makes use of a fast nondominated sorting technique called T-ENS to enhance computation performance. According to empirical research, LMEA is quite useful for handling large-scale MaOPs, particularly those with up to 5000 decision variables.
- **LMOCSO**- An enhanced version of CSO, that is, large-scale Multi-Objective Competitive Swarm Optimizer (LMOCSO) is developed for LSMOPs. It incorporates a competitive mechanism and a unique particle updating approach to enhance search efficiency. This approach is adopted to overcome the limitations of traditional MOEAs when dealing with large-scale MOPs. Through experimental evaluations, this method has demonstrated its superiority over existing MOEAs. Its potential to handle the complexity of optimization problems would become more apparent with more improvements.
- **S3CMAES**-The S3CMAES is designed primarily to address MOPs having large-scale decision variables. Its primary purpose consists of approximate a collection of solutions that are Pareto-optimal, using a novel technique employing tiny subpopulations rather than a standard whole-population strategy. In this novel approach, each subpopulation is dedicated to exploring and improving a specific solution, leveraging a small population for this purpose. An important feature of S3CMAES

is the inclusion of a diversity enhancement approach. This approach is used to choose new solutions from those generated by convergent subpopulations.

- **LMEA-hybrid**- To tackle the scalability issues inherent in MaOPs, we introduce a hybrid Learning-based multi-objective evolutionary algorithm (LMEA-hybrid). The proposed approach combines the virtues of decision space decomposition and sophisticated selection mechanisms to improve both convergence and diversity when solving large-scale MaOPs. The algorithm starts with the initialization of a random population and partitioning the decision variables into the groups related to convergence and diversity. It is done through the LMEA's variable clustering strategy to maintain the problem structure and better direct the process of creating offspring. Once clustering is done, the population is assessed and the Tchebycheff-based environmental selection (T-ENS) is utilized to assign the fitness values. T-ENS balances convergence and diversity and acts as a sorting mechanism for the whole optimization process. In each iteration, two parents are chosen randomly, and a fitness-based tournament selection is employed to decide upon a winner and a loser. The offspring is produced by taking advantage of decision variable group knowledge: convergence-related variables are copied from the winner to preserve convergence pressure, and diversity-related variables are perturbed to search the space more widely. The offspring is assessed and re-placed back into the population via an environmental selection procedure that hybridizes T-ENS with angle-penalized distance (APD). This hybrid selection method guarantees that only individuals of good quality and well-distributed are maintained across generations. The algorithm proceeds until the number of function evaluations reaches the maximum. The last nondominated solutions in the population are retrieved to estimate the PF. In general, Hybrid-LMEA efficiently optimizes convergence and diversity by leveraging both structural properties in the decision space and advanced objective-space selection methods, making it applicable to large-scale many-objective cases. The Pseudocode of LMEA-hybrid is provided in Algorithm 1.

Algorithm 1 Hybrid-LMEA

```

1:  $P \leftarrow \text{InitializePopulation}(N)$ 
2:  $[C_{group}, D_{group}] \leftarrow \text{ClusterDecisionVariables}(P)$   $\triangleright$  LMEA clustering
3:  $\text{EvaluatePopulation}(P)$ 
4:  $Fitness \leftarrow \text{TENS}(P)$   $\triangleright$  T-ENS sorting
5:  $evalCount \leftarrow N$ 
6: while  $evalCount < MaxEval$  do
7:    $[parent_1, parent_2] \leftarrow \text{SelectRandomPair}(P)$ 
8:   if  $Fitness(parent_1) < Fitness(parent_2)$  then
9:      $Winner \leftarrow parent_1$ 
10:     $Loser \leftarrow parent_2$ 
11:   else
12:      $Winner \leftarrow parent_2$ 
13:     $Loser \leftarrow parent_1$ 
14:   end if
15:    $Offspring \leftarrow \text{GenerateOffspring}(Winner, Loser, C_{group}, D_{group})$ 
16:    $\text{Evaluate}(Offspring)$ 
17:    $P \leftarrow \text{EnvironmentalSelection}(P \cup \{Offspring\}, \text{TENS}, \text{APD})$ 
18:    $evalCount \leftarrow evalCount + 1$ 
19: end while
20: return  $PF \leftarrow \text{ExtractNonDominatedSolutions}(P)$ 

```

3.3 Parameter setting

To attain the best performance, the suggested parameter values have been used. The experimental parameter settings are described below.

- Size of population: To guarantee the experiment's fairness, some of the experiment's settings were set consistently. The population size of every algorithm is fixed at 100.
- Runs and stopping criteria: The stopping criteria for all algorithm involves reaching permitted no. of evaluations. The optimum no. of evaluations for every test instance having 100, 300, or 500 decision variables is fixed to 10,000. To acquire statistical findings, each algorithm was subjected to twenty independent runs.
- Mutation and crossover: For DTLZ test suite problems all three algorithms being compared utilizes simulated binary crossover (SBX) [14]

approach to produce offspring. In addition, in all algorithms, a polynomial mutation is employed to every test problem. The parameters for the both crossover (nc) and mutation (nm) distribution indices are equal to twenty. The crossover probability (pc) is equal to 1.0 and mutation probability (pm) is equal to $1/K$, where K represents the amount of decision variables. As recommended in [25], the control parameters F is equal to 0.5 and CR is equal to 1.0 in the differential evolution (DE) [3] technique.

- Other parameters: In the reference to LMEA, we establish the following parameters: In decision variable clustering, nSel (number of solutions) is equal to 2, nPer (no. of perturbations for solution) is equal to 4. Furthermore, nCor is set to six, which signifies the number of solutions used in the analysis of decision variable interaction. The penalty parameter (α) of APD in the case of LMOCSO is fixed to 2.
- Metric used for evaluating performance: IGD [41], GD, and HV analyzes the effectiveness of all three algorithms that are being compared. IGD considers convergence as well as diversity. A lower IGD value, Low GD value and higher HV value suggests that the solution set obtained is of higher quality.
- Significance Test: The Wilcoxon signed-rank test is used to determine the statistical significance of differences at a 5 % level. This test is used to compare the results of three competing algorithms. With a significance level of 0.05, it uses a two-tailed test to assess if S3CMAES performs better (“+”), similarly (“=”), or worse (“-”) than the comparison algorithms on each test problem.

4 Results and discussions

In this section, we present the performance results and their implications of LMEA, LMOCSO, and S3CMAES on benchmark MaOPs and LSMOPs.

Table 3: IGD values (Mean and Standard deviation) of three algorithms on DTLZ (1-7)

| Problem | M | D | LMEA | LMOCSS | S3CMAES |
|---------|---|-----|----------------------|----------------------|---------------------|
| DTLZ1 | 3 | 100 | 2.8056e+3 (1.07e+2)- | 4.6583e+2 (1.15e+2)+ | 2.4024e+3 (1.64e+2) |
| | | 300 | 8.9264e+3 (2.42e+2)+ | 1.4490e+3 (1.98e+2)+ | 1.1277e+4 (1.53e+3) |
| | | 500 | 1.5172e+4 (2.62e+2)+ | 2.3005e+3 (3.53e+2)+ | 2.0124e+4 (2.69e+3) |
| DTLZ2 | 3 | 100 | 6.5683e+0 (2.21e-1)+ | 6.3459e-1 (1.05e-1)+ | 7.9829e+0 (5.12e-1) |
| | | 300 | 2.1774e+1 (4.39e-1)+ | 1.8936e+0 (2.19e-1)+ | 2.5233e+1 (1.37e+0) |
| | | 500 | 3.7663e+1 (6.57e-1)+ | 3.5089e+0 (4.65e-1)+ | 4.1922e+1 (1.81e+0) |
| DTLZ3 | 3 | 100 | 8.7444e+3 (2.96e+2)- | 1.4619e+3 (2.02e+2)+ | 8.2134e+3 (3.18e+2) |
| | | 300 | 2.9070e+4 (6.50e+2)+ | 4.1622e+3 (6.87e+2)+ | 3.1935e+4 (1.34e+3) |
| | | 500 | 5.0050e+4 (4.85e+2)+ | 6.2391e+3 (1.34e+3)+ | 5.3996e+4 (1.63e+3) |
| DTLZ4 | 3 | 100 | 6.9133e+0 (2.33e-1)+ | 1.191e+0 (3.12e-1)+ | 8.4777e+0 (3.98e-1) |
| | | 300 | 2.2244e+1 (4.83e-1)+ | 5.1170e+0 (2.39e+0)+ | 2.5002e+1 (1.01e+0) |
| | | 500 | 3.7661e+1 (5.94e-1)+ | 1.0649e+1 (5.17e+0)+ | 4.3203e+1 (1.33e+0) |
| DTLZ5 | 3 | 100 | 6.4890e+0 (2.43e-1)+ | 5.1211e-1 (1.14e-1)+ | 8.1563e+0 (3.72e-1) |
| | | 300 | 2.1754e+1 (5.59e-1)+ | 2.0570e+0 (3.13e-1)+ | 2.5061e+1 (1.31e+0) |
| | | 500 | 3.7625e+1 (5.55e-1)+ | 3.5189e+0 (6.32e-1)+ | 4.1749e+1 (1.33e+0) |
| DTLZ6 | 3 | 100 | 8.7049e+1 (3.82e-1)- | 3.3804e+1 (5.72e+0)+ | 7.9943e+1 (1.30e+0) |
| | | 300 | 2.6715e+2 (4.57e-1)+ | 1.3167e+2 (5.41e+0)+ | 2.7156e+2 (1.50e+0) |
| | | 500 | 4.4843e+2 (8.45e-1)+ | 2.2919e+2 (8.35e+0)+ | 4.5261e+2 (1.97e+0) |
| DTLZ7 | 3 | 100 | 1.0803e+1 (4.73e-1)- | 6.9905e+0 (9.38e-1)+ | 9.9503e+0 (6.29e-1) |
| | | 300 | 1.1243e+1 (2.87e-1)+ | 1.0059e+1 (3.52e-1)+ | 1.3865e+1 (8.06e-1) |
| | | 500 | 1.1348e+1 (2.93e-1)+ | 1.0525e+1 (2.67e-1)+ | 1.3494e+1 (1.03e+0) |
| +/-/≈ | | | 17/4/0 | 21/0/0 | |

Table 4: IGD values (Mean and standard dDeviation) of three algorithms on DTLZ (1-7)

| Problem | M | D | LMEA | LMOCOS | S3CMAES |
|-----------------|---|-----|----------------------|-----------------------|---------------------|
| DTLZ1 | 5 | 100 | 2.3502e+3 (9.23e+1)- | 4.5784e+2 (1.00e+2)+ | 2.1072e+3 (1.10e+2) |
| | | 300 | 7.3092e+3 (1.88e+2)+ | 1.3851e+3 (1.85e+2) + | 1.1644e+4 (2.00e+3) |
| | | 500 | 1.2393e+4 (3.41e+2)+ | 2.2234e+3 (3.57e+2)+ | 1.8099e+4 (3.89e+3) |
| DTLZ2 | 5 | 100 | 6.6572e+0 (2.43e-1)+ | 9.4997e-1 (7.61e-2)+ | 7.9336e+0 (6.02e-1) |
| | | 300 | 2.1670e+1 (5.15e-1)+ | 2.0493e+0 (2.82e-1)+ | 2.4789e+1 (9.80e-1) |
| | | 500 | 3.7182e+1 (7.26e-1)+ | 3.3173e+0 (4.60e-1)+ | 4.2061e+1 (1.79e+0) |
| DTLZ3 | 5 | 100 | 8.6843e+3 (2.72e+2)- | 1.5013e+3 (2.23e+2)+ | 8.2648e+3 (3.59e+2) |
| | | 300 | 2.8728e+4 (3.18e+2)+ | 4.4336e+3 (6.63e+2)+ | 3.1929e+4 (1.31e+3) |
| | | 500 | 4.9434e+4 (5.22e+2)+ | 7.0644e+3 (8.91e+2)+ | 5.2833e+4 (1.46e+3) |
| DTLZ4 | 5 | 100 | 6.8204e+0 (2.08e-1)+ | 1.4514e+0 (2.47e-1)+ | 8.1818e+0 (3.87e-1) |
| | | 300 | 2.1744e+1 (5.98e-1)+ | 4.4682e+0 (8.52e-1)+ | 2.5080e+1 (1.38e+0) |
| | | 500 | 3.7147e+1 (6.80e-1)+ | 8.6185e+0 (1.78e+0)+ | 4.2167e+1 (9.33e-1) |
| DTLZ5 | 5 | 100 | 6.1582e+0 (2.99e-1)+ | 7.1284e-1 (2.31e-1)+ | 7.9740e+0 (5.01e-1) |
| | | 300 | 2.1501e+1 (5.56e-1)+ | 1.9727e+0 (3.52e-1)+ | 2.5092e+1 (1.55e+0) |
| | | 500 | 3.7145e+1 (5.99e-1)+ | 3.3942e+0 (7.90e-1)+ | 4.1757e+1 (1.76e+0) |
| DTLZ6 | 5 | 100 | 8.5176e+1 (2.93e-1)- | 3.7580e+1 (2.20e+0)+ | 7.8989e+1 (1.28e+0) |
| | | 300 | 2.6512e+2 (8.75e-1)+ | 1.3220e+2 (8.30e+0)+ | 2.6912e+2 (1.53e+0) |
| | | 500 | 4.4612e+2 (7.75e-1)+ | 2.2844e+2 (8.98e+0)+ | 4.5008e+2 (1.67e+0) |
| DTLZ7 | 5 | 100 | 1.7922e+1 (5.37e-1)- | 9.2129e+0 (4.86e+0)+ | 1.6081e+1 (9.61e-1) |
| | | 300 | 1.9050e+1 (5.67e-1)+ | 1.6877e+1 (2.12e+0)+ | 2.3008e+1 (9.58e-1) |
| | | 500 | 1.9066e+1 (4.42e-1)+ | 1.7968e+1 (1.22e+0)+ | 2.2818e+1 (1.27e+0) |
| +/- / \approx | | | 17/4/0 | 21/0/0 | |

Table 5: IGD values (Mean and Standard Deviation) of three algorithms on DTLZ (1-7)

| Problem | M | D | LMEA | LMOCSS | S3CMAES |
|---------|----|-----|----------------------|----------------------|---------------------|
| DTLZ1 | 10 | 100 | 2.0209e+3 (1.68e+2)- | 4.7954e+2 (2.64e+2)+ | 1.5317e+3 (9.58e+1) |
| | | 300 | 6.4660e+3 (3.57e+2)+ | 1.1081e+3 (4.27e+2)+ | 1.0343e+4 (1.71e+3) |
| | | 500 | 1.0131e+4 (2.44e+3)+ | 1.8623e+3 (6.58e+2)+ | 1.8370e+4 (3.20e+3) |
| DTLZ2 | 10 | 100 | 6.4863e+0 (2.72e-1)+ | 1.2657e+0 (1.88e-1)+ | 7.6738e+0 (3.62e-1) |
| | | 300 | 2.1846e+1 (6.00e-1)+ | 2.6807e+0 (1.18e+0)+ | 2.4682e+1 (1.43e+0) |
| | | 500 | 3.7295e+1 (5.68e-1)+ | 4.6228e+0 (2.05e+0)+ | 4.1881e+1 (1.78e+0) |
| DTLZ3 | 10 | 100 | 8.0562e+3 (3.08e+2)= | 1.4040e+3 (2.26e+2)+ | 7.9046e+3 (4.80e+2) |
| | | 300 | 2.8608e+4 (3.56e+2)+ | 4.3359e+3 (7.42e+2)+ | 3.1742e+4 (9.98e+2) |
| | | 500 | 4.9083e+4 (5.49e+2)+ | 6.9921e+3 (1.48e+3)+ | 5.3422e+4 (1.58e+3) |
| DTLZ4 | 10 | 100 | 6.7155e+0 (1.81e-1)+ | 1.7600e+0 (1.22e-1)+ | 7.2796e+0 (3.82e-1) |
| | | 300 | 2.1871e+1 (4.69e-1)+ | 4.2560e+0 (8.62e-1)+ | 2.4822e+1 (1.22e+0) |
| | | 500 | 3.7280e+1 (6.66e-1)+ | 6.9600e+0 (1.43e+0)+ | 4.1852e+1 (1.64e+0) |
| DTLZ5 | 10 | 100 | 5.9466e+0 (3.66e-1)+ | 7.1356e-1 (5.15e-2)+ | 7.6499e+0 (3.97e-1) |
| | | 300 | 2.1430e+1 (5.64e-1)+ | 1.8793e+0 (1.05e+0)+ | 2.4832e+1 (1.15e+0) |
| | | 500 | 3.6438e+1 (7.19e-1)+ | 3.8951e+0 (1.51e+0)+ | 4.1086e+1 (1.48e+0) |
| DTLZ6 | 10 | 100 | 8.0928e+1 (3.25e-1)- | 3.8122e+1 (1.96e+0)+ | 7.5570e+1 (1.23e+0) |
| | | 300 | 2.6064e+2 (8.53e-1)+ | 1.3104e+2 (4.47e+0)+ | 2.6520e+2 (1.48e+0) |
| | | 500 | 4.4133e+2 (7.52e-1)+ | 2.2564e+2 (6.79e+0)+ | 4.4615e+2 (1.78e+0) |
| DTLZ7 | 10 | 100 | 3.7228e+1 (1.72e+0)- | 6.7123e+0 (3.45e+0)+ | 3.1854e+1 (1.53e+0) |
| | | 300 | 4.0189e+1 (8.21e-1)+ | 2.6920e+1 (3.19e+0)+ | 4.6188e+1 (1.91e+0) |
| | | 500 | 3.9829e+1 (8.05e-1)+ | 3.0637e+1 (3.35e+0)+ | 4.6048e+1 (2.16e+0) |
| +/-/≈ | | | 17/3/1 | 21/0/0 | |

4.1 Performance comparison of LMEA, LMOCSO, and S3CMAES on benchmark MaOPs

Tables 3, 4, and 5 display the IGD values (mean and standard deviation) of the three compared algorithms on the seven DLTZ problems having decision variables (100,300,500), as well as number of objectives (3,5,10) acquired via 20 independent runs. The superior result on every test instance is highlighted in bold. The Wilcoxon rank sum test has been employed at a significance level of 0.05. The symbols “+”, “−”, and “ \approx ” signify that the result is statistically better, significantly worse and significantly similar to that obtained by S3CMAES, respectively. We can infer the following observations from the Tables 3–5. This can be observed that IGD values generated by LMOCSO, LMEA on every test problem are consistently improve as the amount of decision variables grows from 100 to 500, that indicates a potential scalability of LMOCSO and LMEA. LMOCSO surpasses the other two algorithms in solving DLTZ1-7 whereas LMEA coming second. LMOCSO is the best in all instances while S3CMAES have inferior performance. LMEA performs fairly better in DLTZ2, DLTZ4 and DLTZ5. Low IGD values indicate higher algorithm performance.

The performance of the three algorithms is visually compare with respect to diversity and convergence, the coordinates of output solution acquired by these three algorithms on ten objective DLTZ6 are presented in Figure 1. From Figure 1b, we can deduce that the output solution of the LMOCSO algorithm outperforms the other two compared algorithms with respect to diversity and convergence. In LMEA, solutions at objective ten fails to converge.

4.2 Performance comparison of LMEA, LMOCSO and S3CMAES on LSMOP test suite

To assess the effectiveness of LMEA, LMOCSO, and S3CMAES on difficult Large-scale MaOPs, experiments are conducted on the LSMOP suite. The IGD values of three algorithms are presented in Table 6 having six objectives

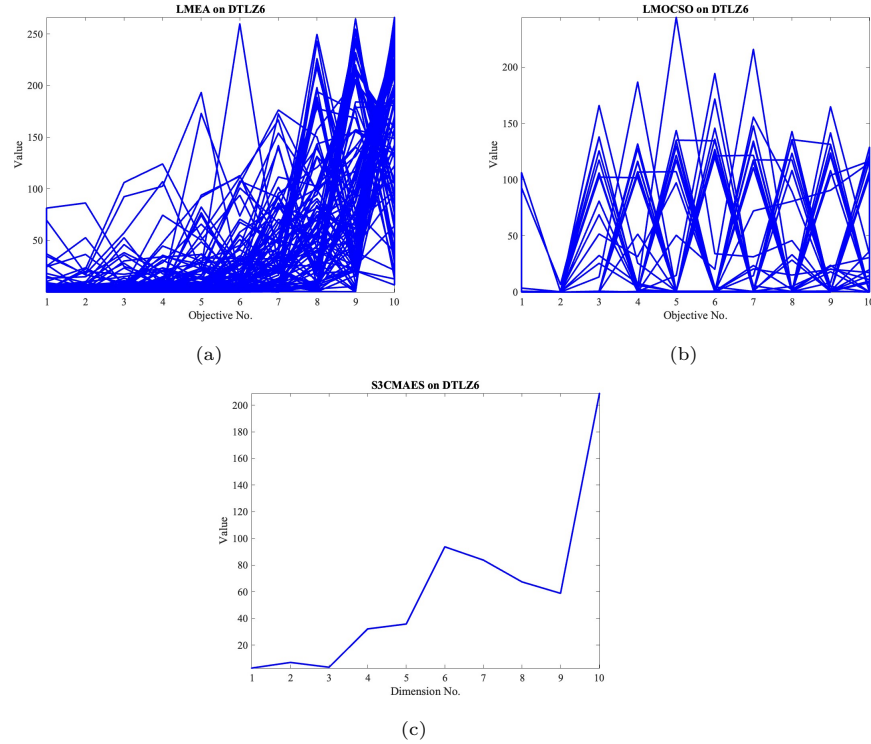


Figure 1: Solutions acquired by three compared algorithms on ten-objective DTLZ six problems with 300 decision variables (a) LMEA on DTLZ6. (b) LMOCSO on DTLZ6. (c) S3CMAES on DTLZ6.

and (100, 300, and 500) decision variables acquired via twenty independent runs. From Table 6, the subsequent observations can be drawn. LMEA having 500 decision variables is performing better than two algorithms on LSMOP2, LSMOP3, and LSMOP7-8. LMOCSO completely outperforms the other two algorithms on LSMOP1, LSMOP4, LSMOP5-6, and LSMOP9. In the case of LSMaOPs, LMOCSO outperforms the other two algorithms on 23 out of 27 test instances.

Table 7 presents the GD values of all three compared algorithms. LMEA performs better on 14 out of 27 test instances, while LMOCSO performs better on 9 out of 27 test instances. S3CMAES performs better on 4 out of 27 test instances. Therefore, LMEA performs superior to the other two algorithms in terms of GD values, while LMOCSO performs second best.

LMEA outperforms LSMOP1-2 with 500 decision variables and LSMOP4,7,8 with 300 and 500 decision variables. LMEA significantly performs better on LSMOP5,8. LMOCSO performs better than other algorithms on LSMOP1-2 with 300 decision variables. LMOCSO performs significantly better on LSMOP3,6 with 100, 300, and 500 decision variables and on LSMOP 9 with 100 decision variables.

Table 8 shows the HV values of three algorithms. Out of 27 LSMOP test instances, LMEA performs better in 19 instances. In comparison, LMOCSO performs better in 8 out of 27 instances. LMOCSO outperforms the other two algorithms on LSMOP1-2 having 100 and 300 decision variables, also LSMOP4 with 100, 300 and 500 decision variables and LSMOP5 with 100 decision variables. With regards to HV performance on the majority of LSMOP issues, it can be concluded that LMEA and LMOCSO are the best algorithms.

To confirm the computational efficiency of LMEA, LMOCSO and S3CMAES, Table 9 presents the average runtime(s) of the three compared algorithms on all runs for the test instances with 100, 300, and 500 decision variables, respectively. It can be noted that LMOCSO is computationally more efficient than the LMEA and S3CMAES on mostly all the test instances except LSMOP9 with 100 decision variables, where S3CMAES outperforms the other two algorithms.

The performance of the LMEA, LMOCSO, and S3CMAES is compared visually in terms of both convergence and diversity, and the output solution acquired by the three algorithms on 10-objective LSMOP6 is presented in Figure 2. From Figure 2a, we can deduce that the output solution set of the LMEA algorithm is much better than the other two compared algorithms with regard to both convergence and diversity. S3CMAES is performing worst in terms of convergence. The output solution of three algorithms on 10-objective LSMOP7 is presented in Figure 3. From Figure 3c, we can observe that S3CMAES is performing drastically worse than the other two algorithms with regard to convergence. From Figure 3a we can conclude that LMEA is performing more effectively.

The output solution of three algorithms on 10-objective LSMOP8 is presented in Figure 4. From Figure 4a, solutions of LMEA fail to converge at

objective 10. From Figure 4b, the solution set in the case of LMOCSO is not diverse.

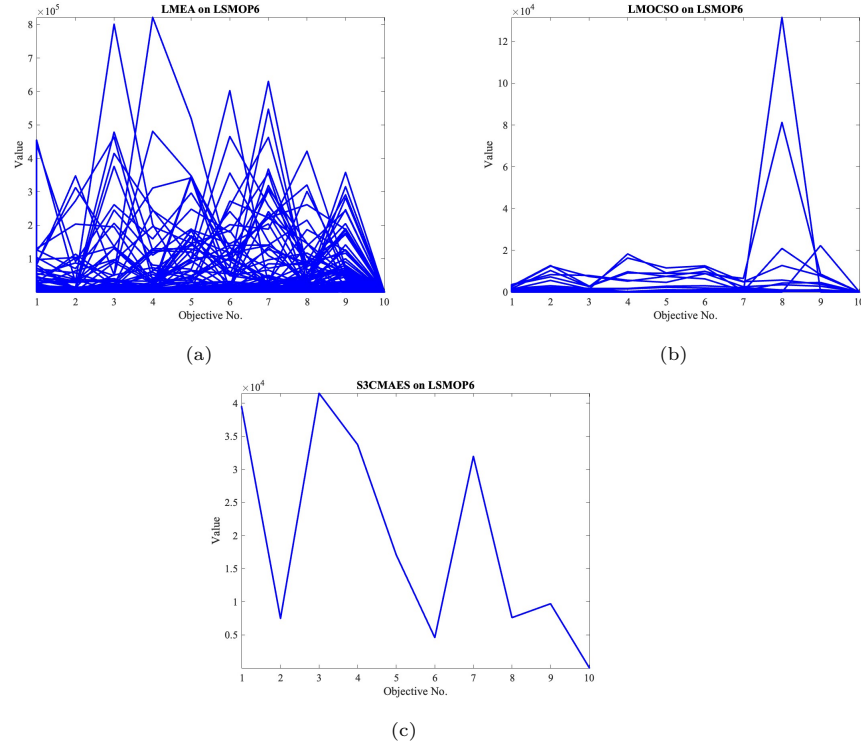


Figure 2: Solutions acquired by three compared algorithms on 10-objective LSMOP6 problems with 1000 decision variables (a) LMEA on LSMOP6. (b) LMOCSO on LSMOP6. (c) S3CMAES on LSMOP6.

4.3 Performance comparison of LMEA, LMOCSO and S3CMAES on UF and WFG test suites

In this section, we have compared the performance analysis of LMEA, LMOCSO and S3CMAES on UF and WFG test suites. The two problems UF9, UF10 and five problems WFG1-WFG5 have been considered from UF and WFG test suites respectively. Table 10 displays the statistical results of the IGD values of three algorithms with three objectives and 100, 300,

Table 6: IGD values (Mean and standard deviation) of three algorithms on LSMOP (1-9) using Wilcoxon signed-Rank test.

| Problem | M | D | LMEA | LMOCSSO | S3CMAES |
|---------|---|-------------------|---|--|--|
| LSMOP1 | 6 | 100 300 500 | 7.8505e+0(1.18e+0) ≈ 9.7993e+0(8.53e-1)+ 1.0371e+1(6.96e-1)+ | 1.1162e+0(2.20e-1)+ 1.9560e+0 (4.79e-1) + 1.0365e+1 (1.03e+0)+ | 9.3382e+0(6.24e+0) 3.4866e+1(2.53e+1) 5.7233e+1(3.93e+1) |
| LSMOP2 | 6 | 100 300 500 | 6.5253e-1(3.89e-2)- 3.8340e-1(2.62e-2)+ 3.2267e-1 (2.41e-2) | 4.9476e-1 (2.38e-2) + 2.8391e-1 (2.77e-2) + 3.4882e-1(1.60e-2)+ | 5.6671e-1(5.79e-2) 1.1558e+0(5.50e-1) 9.9936e-1(3.57e-1) |
| LSMOP3 | 6 | 100 300 500 | 5.1876e+3(2.21e+3)- 8.9438e+3(4.77e+3)+ 9.1783e+3 (4.15e+3) + | 1.5982e+1 (2.25e+1) + 5.5185e+1 (9.98e+1) + 1.2884e+4(7.44e+3)+ | 6.9334e+2(1.34e+3) 2.2229e+5(3.51e+5) 2.3369e+5(3.74e+5) |
| LSMOP4 | 6 | 100 300 500 | 7.8327e-1(8.90e-2)+ 4.9135e-1(3.24e-2)+ 4.1160e-1(2.53e-2)+ | 5.5492e-1 (4.99e-2) + 3.8595e-1 (4.89e-2) + 3.9979e-1 (7.76e-2) + | 9.6423e-1(1.82e-1) 1.6182e+0(8.10e-1) 1.0240e+0(4.33e-1) |
| LSMOP5 | 6 | 100 300 500 | 1.2874e+1(2.03e+0)+ 1.5523e+1(1.31e+0)+ 1.6260e+1(7.47e-1)+ | 2.1034e+0 (5.56e-1) + 4.4084e+0 (9.32e-1) + 1.1221e+1 (6.08e+0) + | 1.9173e+1(4.15e-1) 4.6207e+1(3.61e+1) 6.1219e+1(3.92e+1) |
| LSMOP6 | 6 | 100 300 500 | 4.3009e+1(5.91e+1)- 1.8804e+3(1.84e+3)+ 1.8879e+3(1.53e+3)+ | 1.6894e+0 (2.19e-1) + 1.6835e+0 (6.56e-2) + 7.3031e+1 (3.20e+2) + | 2.6136e+0(3.26e-1) 2.0585e+5(2.41e+5) 2.4187e+5(2.78e+5) |
| LSMOP7 | 6 | 100 300 500 | 1.6657e+4(4.20e+3) ≈ 2.5816e+4(4.81e+3)+ 2.8057e+4 (4.39e+3) + | 3.1630e+1 (2.78e+1) + 1.7644e+3 (1.15e+3) + 2.9089e+4(8.93e+3)+ | 1.9017e+4(8.13e+3) 3.1050e+5(3.35e+5) 2.8951e+5(2.82e+5) |
| LSMOP8 | 6 | 100 300 500 | 1.0310e+1(1.53e+0)+ 1.1069e+1(1.35e+0)+ 1.1879e+1 (7.97e-1) + | 2.3124e+0 (6.34e-1) + 3.7021e+0 (5.83e-1) + 1.2575e+1 (1.24e+0)+ | 1.6548e+1(5.45e+0) 3.0094e+1(1.67e+1) 3.2233e+1(1.91e+1) |
| LSMOP9 | 6 | 100 300 500 | 3.5670e+2(3.19e+1)+ 4.0461e+2(2.14e+1)+ 4.2396e+2(2.17e+1)+ | 4.7217e+1 (2.21e+1) + 1.1058e+2 (5.15e+1) + 1.5070e+2 (6.13e+1) + | 4.3889e+2(7.85e+1) 8.8449e+2(3.87e+2) 9.4298e+2(3.38e+2) |
| +/- / ≈ | | | 22 / 3 / 2 | 27 / 0 / 0 | |

Table 7: GD values (Mean and standard deviation) of three algorithms on LSMOP (1-9) using Wilcoxon signed- Rank test.

| Problem | M | D | LMEA | LMOCSO | S3CMAES |
|-----------------|---|-----|--------------------------------------|--------------------------------------|----------------------------|
| LSMOP1 | 6 | 100 | 4.9528e + 0(6.55e - 1) - | 3.0575e + 0(8.21e - 1) - | 1.7881e+0 (1.62e+0) |
| | | 300 | 5.1742e + 0(5.11e - 1) \approx | 4.5261e+0(1.54e+0) + | 8.5532e + 0(6.30e + 0) |
| | | 500 | 5.2389e+0 (3.86e-1) + | 7.7224e + 0(1.19e + 0) \approx | 1.4128e + 1(9.81e + 0) |
| LSMOP2 | 6 | 100 | 2.0471e - 1(1.08e - 2) - | 1.1158e - 1(8.90e - 3) - | 8.1107e-2 (4.86e-3) |
| | | 300 | 7.4545e - 2(3.94e - 3) \approx | 4.6465e-2 (7.63e-3) + | 1.3081e - 1(9.84e - 2) |
| | | 500 | 4.2939e-2 (2.93e-3) + | 6.7061e - 2(8.24e - 3) \approx | 7.8811e - 2(5.31e - 2) |
| LSMOP3 | 6 | 100 | 4.4162e + 4(8.99e + 3) - | 4.0139e+2 (8.10e+2) + | 7.5513e + 3(2.21e + 3) |
| | | 300 | 7.2507e + 4(8.24e + 3) - | 3.9390e+3 (6.40e+3) + | 5.5572e + 4(8.77e + 4) |
| | | 500 | 7.5216e + 4(7.48e + 3) - | 3.7786e+4 (5.28e+4) \approx | 5.8423e + 4(9.36e + 4) |
| LSMOP4 | 6 | 100 | 3.8524e - 1(2.57e - 2) - | 3.1893e - 1(5.87e - 2) - | 1.1344e-1 (8.29e-2) |
| | | 300 | 1.3473e-1 (1.00e-2) \approx | 1.3485e - 1(2.61e - 2) \approx | 2.4017e - 1(1.69e - 1) |
| | | 500 | 7.6567e-2 (6.30e-3) \approx | 1.3315e - 1(4.38e - 2) - | 1.0934e - 1(7.95e - 2) |
| LSMOP5 | 6 | 100 | 1.1538e+1 (1.85e+0) \approx | 1.6819e + 1(7.42e + 0) \approx | 1.3671e + 1(6.48e + 0) |
| | | 300 | 8.4363e+0(8.73e-1) \approx | 1.2601e + 1(4.34e + 0) \approx | 1.1438e + 1(9.03e + 0) |
| | | 500 | 8.0116e+0(6.60e-1) + | 1.2601e + 1(4.34e + 0) \approx | 1.5188e + 1(9.81e + 0) |
| LSMOP6 | 6 | 100 | 2.6669e + 4(5.36e + 3) + | 1.2090e+1(2.47e+1) + | 8.5007e + 4(5.97e + 4) |
| | | 300 | 5.0053e + 4(8.07e + 3) \approx | 3.1980e+4(4.01e+4) \approx | 5.1463e + 4(6.03e + 4) |
| | | 500 | 4.8144e + 4(8.84e + 3) \approx | 4.6665e+4(3.76e+4) \approx | 6.0468e + 4(6.96e + 4) |
| LSMOP7 | 6 | 100 | 3.6251e + 4(2.15e + 4) - | 1.1068e + 4(2.00e + 4) \approx | 9.1504e + 3(7.38e + 3) |
| | | 300 | 5.1071e+4(2.03e+4) \approx | 5.3012e + 4(5.64e + 4) \approx | 7.7625e + 4(8.38e + 4) |
| | | 500 | 4.2243e+4(1.25e+4) \approx | 5.6367e + 4(2.23e + 4) \approx | 7.2377e + 4(7.04e + 4) |
| LSMOP8 | 6 | 100 | 7.0013e+0 (1.38e+0) \approx | 8.9649e + 0(3.51e + 0) \approx | 7.1583e + 0(3.95e + 0) |
| | | 300 | 4.4857e+0(3.33e-1) + | 9.0603e + 0(2.34e + 0) - | 7.4020e + 0(4.18e + 0) |
| | | 500 | 4.1446e+0 (4.83e-1) + | 7.6076e + 0(6.95e - 1) \approx | 7.9355e + 0(4.78e + 0) |
| LSMOP9 | 6 | 100 | 1.2912e + 2(1.19e + 1) - | 4.8371e + 1(1.78e + 1) + | 6.5185e + 1(1.01e + 1) |
| | | 300 | 1.1175e+2 (7.51e+0) + | 1.5240e + 2(6.36e + 1) + | 2.2024e + 2(9.66e + 1) |
| | | 500 | 1.0975e+2 (6.53e+0) + | 1.9926e + 2(7.90e + 1) \approx | 2.3486e + 2(8.45e + 1) |
| +/- / \approx | | | 8/8/11 | 7/5/15 | |

Table 8: HV values (Mean and standard deviation) of three algorithms on LSMOP (1-9) using Wilcoxon signed-Rank test.

| Problem | M | D | LMSEA | LMOCSSO | S3CMAES |
|-------------------|-----|-----|------------------------------|------------------------------|----------------------|
| LSMOP1 | 100 | 100 | $0.0000e+0(0.00e+0) \approx$ | $2.9069e-2(4.14e-2) +$ | $0.0000e+0(0.00e+0)$ |
| | 300 | 300 | $0.0000e+0(0.00e+0) \approx$ | $4.1879e-4(1.49e-3) +$ | $0.0000e+0(0.00e+0)$ |
| | 500 | 500 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| LSMOP2 | 100 | 100 | $2.6831e-1(3.84e-2)-$ | $5.1213e-1(3.87e-2) +$ | $4.6599e-1(8.46e-2)$ |
| | 300 | 300 | $6.5017e-1(2.59e-2) +$ | $8.3408e-1(3.92e-2) +$ | $7.5139e-2(8.32e-2)$ |
| | 500 | 500 | $7.5768e-1(2.21e-2) +$ | $6.7722e-1(2.89e-2) +$ | $9.1660e-2(9.43e-2)$ |
| LSMOP3 | 100 | 100 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 300 | 300 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 500 | 500 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| LSMOP4 | 100 | 100 | $1.5055e-1(5.86e-2) +$ | $5.4001e-1(9.23e-2) +$ | $7.6731e-2(1.20e-1)$ |
| | 300 | 300 | $5.1515e-1(2.95e-2) +$ | $7.4661e-1(8.03e-2) +$ | $4.3841e-2(6.35e-2)$ |
| | 500 | 500 | $6.2718e-1(3.16e-2) +$ | $6.4191e-1(1.25e-1) +$ | $1.0102e-1(7.89e-2)$ |
| LSMOP5 | 100 | 100 | $0.0000e+0(0.00e+0) \approx$ | $1.7955e-4(8.03e-4) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 300 | 300 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 500 | 500 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| LSMOP6 | 100 | 100 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 300 | 300 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 500 | 500 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| LSMOP7 | 100 | 100 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 300 | 300 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 500 | 500 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| LSMOP8 | 100 | 100 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 300 | 300 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 500 | 500 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| LSMOP9 | 100 | 100 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 300 | 300 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| | 500 | 500 | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0) \approx$ | $0.0000e+0(0.00e+0)$ |
| + / - / \approx | | | 5/1/21 | 8/0/19 | |

Table 9: Run Time values (Mean and Standard Deviation) of three algorithms on LSMOP (1-9) using Wilcoxon signed-Rank test.

| Problem | M | D | LMEA | LMOCSS | S3CMAES |
|-------------------|---|-----|-----------------------------------|-----------------------|----------------------------|
| LSMOP1 | 6 | 100 | 3.2656e + 1 (1.13e + 2) - | 4.7273e-1 (7.90e-2) + | 2.9315e + 0 (2.24e - 1) |
| | | 300 | 8.5448e + 1 (4.33e + 1) - | 8.6804e-1 (1.34e-1) + | 1.7116e + 1 (1.51e - 1) |
| | | 500 | 8.2375e + 2 (2.33e + 3) - | 5.5474e-2 (4.94e-2) + | 4.7838e + 1 (1.57e + 0) |
| LSMOP2 | 6 | 100 | 8.0957e + 0 (3.61e + 0) - | 6.3782e-1 (5.55e-2) + | 2.8009e + 0 (5.68e - 2) |
| | | 300 | 6.3424e + 1 (4.14e + 1) - | 7.6440e-1 (4.81e-2) + | 1.9163e + 1 (1.42e - 1) |
| | | 500 | 4.5228e + 2 (1.26e + 3) - | 8.5914e-2 (1.02e-1) + | 5.3438e + 1 (3.15e + 0) |
| LSMOP3 | 6 | 100 | 8.5731e + 0 (2.99e + 0) - | 1.7761e+0 (8.12e-1) + | 2.7852e + 0 (6.46e - 2) |
| | | 300 | 9.5423e + 1 (4.09e + 1) - | 1.3840e+0 (4.57e-1) + | 1.8732e + 1 (1.21e - 1) |
| | | 500 | 3.0195e + 2 (2.47e + 2) - | 6.7954e-2 (4.85e-2) + | 5.2643e + 1 (1.92e + 0) |
| LSMOP4 | 6 | 100 | 8.7249e + 0 (4.40e + 0) - | 6.4083e-1 (2.03e-2) + | 2.8146e + 0 (1.17e - 1) |
| | | 300 | 9.9880e + 1 (4.28e + 1) - | 8.1007e-1 (5.90e-2) + | 1.8770e + 1 (2.12e - 1) |
| | | 500 | 5.6169e + 2 (9.16e + 2) - | 1.4838e-1 (1.45e-1) + | 5.2188e + 1 (4.37e - 1) |
| LSMOP5 | 6 | 100 | 7.6427e + 0 (6.46e + 0) - | 7.0459e-1 (9.46e-2) + | 2.6329e + 0 (1.15e - 1) |
| | | 300 | 7.9805e + 1 (2.86e + 1) - | 7.0803e-1 (4.88e-2) + | 1.8190e + 1 (2.53e - 1) |
| | | 500 | 1.3344e + 2 (1.18e + 2) - | 1.2870e-1 (9.68e-2) + | 5.0431e + 1 (7.99e - 1) |
| LSMOP6 | 6 | 100 | 4.5128e + 0 (2.48e + 0) \approx | 9.6853e-1 (2.03e-1) + | 3.0962e + 0 (1.10e - 1) |
| | | 300 | 4.6114e + 1 (2.46e + 1) - | 8.5650e-1 (1.47e-1) + | 2.0664e + 1 (2.09e + 1) |
| | | 500 | 6.6066e + 2 (2.42e + 3) \approx | 2.2104e-1 (1.40e-1) + | 5.7946e + 1 (1.05e + 0) |
| LSMOP7 | 6 | 100 | 8.5955e + 0 (2.63e + 0) - | 1.9460e+0 (1.00e+0) + | 2.9440e + 0 (9.10e - 2) |
| | | 300 | 7.2473e + 1 (4.23e + 1) - | 1.4504e+0 (4.83e-1) + | 2.0520e + 1 (2.96e - 1) |
| | | 500 | 2.0263e + 2 (1.20e + 2) - | 5.0781e-2 (1.73e-2) + | 5.6966e + 1 (4.94e - 1) |
| LSMOP8 | 6 | 100 | 8.8211e + 0 (2.42e + 0) - | 6.9920e-1 (1.27e-1) + | 2.8547e + 0 (7.95e - 2) |
| | | 300 | 7.8453e + 1 (3.62e + 1) - | 8.3320e-1 (9.83e-2) + | 2.1785e + 1 (4.12e - 1) |
| | | 500 | 5.7518e + 2 (1.11e + 3) - | 7.6261e-2 (3.79e-2) + | 5.6005e + 1 (3.54e - 1) |
| LSMOP9 | 6 | 100 | 6.8140e + 0 (4.24e + 0) - | 3.6036e+0 (4.28e-1) - | 2.6631e+0 (7.04e-2) |
| | | 300 | 8.1672e + 1 (5.39e + 1) - | 4.2458e+0 (3.68e-1) + | 1.8807e + 1 (3.62e - 1) |
| | | 500 | 2.6924e + 2 (1.41e + 2) - | 4.3651e+0 (4.67e-1) + | 5.2610e + 1 (6.81e - 1) |
| + / - / \approx | | | 0/25/2 | 26/1/0 | |

Table 10: IGD values (Mean and standard ddeviation) of three algorithms on UF9,10 and WFG (1-9) using Wilcoxon signed-Rank test.

| Problem | M | D | LMEA | LMOCSSO | S3CMAES |
|-------------------|---|-----|--------------------------|-----------------------|------------------------|
| UF9 | 3 | 100 | 4.0297e + 0(1.87e - 1)- | 8.5927e-1 (9.65e-2) + | 2.0081e + 0(4.32e - 1) |
| | | 300 | 4.3633e + 0(1.15e - 1)+ | 1.7067e+0 (3.76e-1) + | 7.1931e + 0(2.40e + 0) |
| | | 500 | 4.4402e + 0(1.04e - 1)+ | 1.8838e+0 (3.87e-1) + | 7.3970e + 0(2.55e + 0) |
| UF10 | 3 | 100 | 1.8980e + 1(9.27e - 1)- | 4.7046e+0 (9.49e-1) + | 9.7564e + 0(1.68e + 0) |
| | | 300 | 2.0183e + 1(6.79e - 1)+ | 8.4764e+0 (1.79e+0) + | 2.9497e + 1(8.98e + 0) |
| | | 500 | 2.0747e + 1(4.20e - 1)+ | 1.0587e+1 (1.91e+0) + | 2.7901e + 1(6.56e + 0) |
| WFG1 | 3 | 100 | 2.4084e + 0(6.41e - 2)- | 1.8117e+0 (1.56e-1) + | 1.9070e + 0(3.46e - 2) |
| | | 300 | 2.4084e + 0(6.20e - 2)+ | 1.9678e+0 (1.86e-1) + | 2.7204e + 0(5.07e - 2) |
| | | 500 | 2.3966e + 0(5.69e - 2)+ | 1.8972e+0 (2.12e-1) + | 2.7337e + 0(4.60e - 2) |
| WFG2 | 3 | 100 | 1.1216e + 0(1.38e - 1)+ | 4.6531e-1 (3.14e-2) + | 1.1275e + 0(1.03e + 0) |
| | | 300 | 1.0423e + 0(1.22e - 1)+ | 5.0352e-1 (3.72e-2) + | 4.1776e + 0(9.06e - 1) |
| | | 500 | 1.0731e + 0(8.60e - 2)+ | 5.1609e-1 (4.68e-2) + | 4.0737e + 0(1.08e + 0) |
| WFG3 | 3 | 100 | 8.5789e - 1(1.04e - 2) = | 5.3772e-1 (2.89e-2) + | 8.4965e - 1(3.47e - 2) |
| | | 300 | 8.8114e - 1(8.36e - 3)+ | 5.8650e-1 (2.28e-2) + | 2.3970e + 0(4.62e - 1) |
| | | 500 | 8.8828e - 1(6.01e - 3)+ | 5.9368e-1 (1.25e-2) + | 2.4104e + 0(4.38e - 1) |
| WFG4 | 3 | 100 | 8.8413e - 1(7.30e - 2)- | 3.4440e-1 (1.16e-2) + | 6.5867e - 1(2.39e - 2) |
| | | 300 | 8.6043e - 1(8.15e - 2)+ | 3.5481e-1 (1.06e-2) + | 3.4322e + 0(4.16e - 1) |
| | | 500 | 8.7826e - 1(1.07e - 1)+ | 3.5734e-1 (1.19e-2) + | 3.0926e + 0(5.54e - 1) |
| WFG5 | 3 | 100 | 8.8359e - 1(3.15e - 2)- | 2.7060e-1 (1.16e-2) + | 6.5978e - 1(1.99e - 2) |
| | | 300 | 8.8290e - 1(2.24e - 2)+ | 2.7397e-1 (1.05e-2) + | 3.1492e + 0(3.59e - 1) |
| | | 500 | 8.8048e - 1(2.87e - 2)+ | 2.7839e-1 (8.80e-3) + | 3.4122e + 0(2.24e - 1) |
| + / - / \approx | | | 15 / 5 / 1 | 21 / 0 / 0 | |

Table 11: IGD values (Mean and standard deviation) of three algorithms on LSMOP (1-9) using Wilcoxon signed-Rank test.

| Problem | M | D | LMEA | LMOCSSO | S3CMAES |
|-------------------|---|------|-----------------------|-----------------------|---------------------|
| LSMOP1 | 6 | 1000 | 1.0244e+1 (7.71e-1) + | 2.1477e+0 (2.88e-1) + | 3.7324e+1 (4.88e+1) |
| LSMOP2 | 6 | 1000 | 3.0604e-1 (2.39e-2) + | 3.0383e-1 (3.27e-2) + | 7.2293e-1 (1.93e-1) |
| LSMOP3 | 6 | 1000 | 8.4019e+3 (6.98e+3) + | 7.8985e+3 (4.22e+3) + | 3.4912e+5 (2.40e+5) |
| LSMOP4 | 6 | 1000 | 3.8051e-1 (3.25e-2) + | 3.3057e-1 (3.09e-2) + | 8.8056e-1 (2.12e-1) |
| LSMOP5 | 6 | 1000 | 1.6723e+1 (1.10e+0) + | 1.6423e+1 (7.28e-1) + | 5.7533e+1 (4.40e+1) |
| LSMOP6 | 6 | 1000 | 2.6925e+3 (2.18e+3) + | 1.3219e+3 (1.75e+3) + | 1.4742e+5 (2.75e+5) |
| LSMOP7 | 6 | 1000 | 2.7767e+4 (5.22e+3) + | 2.6676e+4 (3.01e+3) + | 1.3955e+5 (1.13e+5) |
| LSMOP8 | 6 | 1000 | 1.1996e+1 (1.19e+0) + | 1.157e+1 (3.88e-1) + | 2.9253e+1 (1.48e+1) |
| LSMOP9 | 6 | 1000 | 4.5064e+2 (1.27e+1) + | 4.3014e+2 (1.59e+1) + | 8.3041e+2 (2.62e+2) |
| + / - / \approx | | | 9/0/0 | 9/0/0 | |

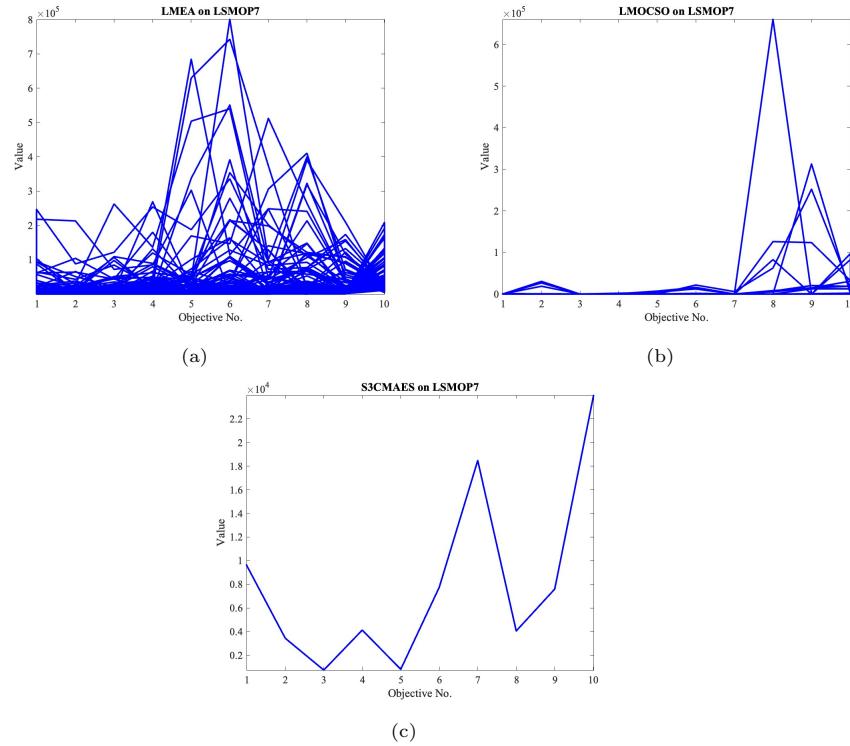


Figure 3: Solutions acquired by three compared algorithms on 10-objective LSMOP7 problems with 1000 decision variables (a) LMEA on LSMOP7. (b) LMOCSO on LSMOP7. (c) S3CMAES on LSMOP7.

and 500 decision variables obtained during 20 separate runs. From Table 10, it can be observed that LMOCSO performs significantly better than the other two algorithms. Whereas LMEA performs significantly better than S3CMAES on 15 test instances.

4.4 Performance comparison of LMEA, LMOCSO and S3CMAES with decision variables 1000

In this section, the performance of LMEA, LMOCSO, S3CMAES has been compared by challenging them with 1000 decision variables. Table 11 presents IGD values obtained by the three algorithms on LSMOP1-LSMOP9 with

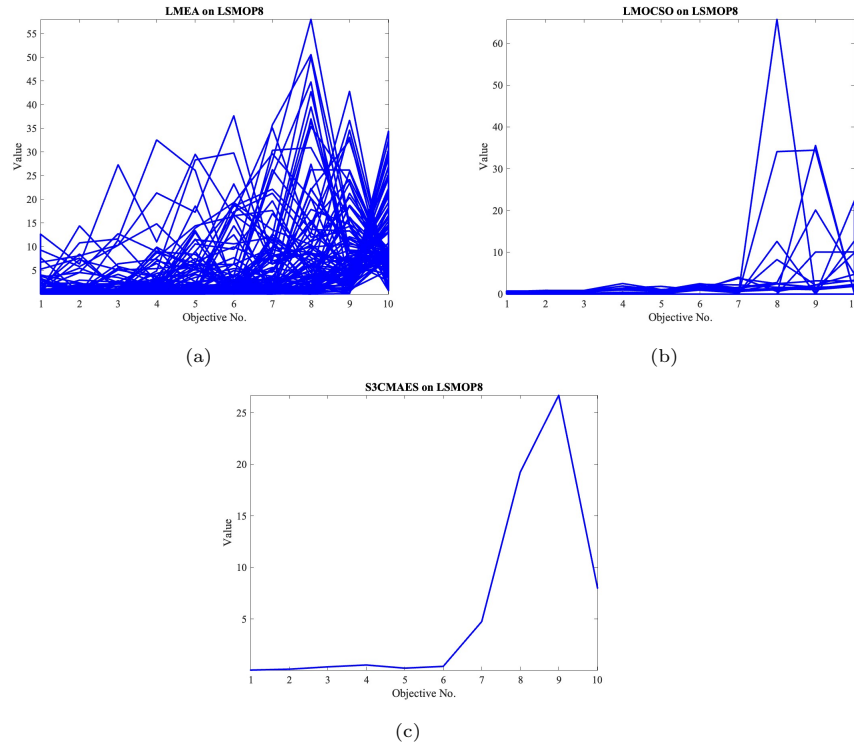


Figure 4: Solutions acquired by three compared algorithms on 10-objective LSMOP8 problems with 1000 decision variables (a) LMEA on LSMOP8. (b) LMOCSO on LSMOP8. (c) S3CMAES on LSMOP8.

six objectives and 1000 decision variables, via 20 runs. It can be seen that LMOCSO still performs the best on LSMOPs, which confirms its effectiveness in solving LSMaOPs.

4.5 Performance comparison of LMEA, LMOCSO, S3CMAES and LMEA-hybrid with decision variables varying between 100-500

In this section, we have compared the performance of LMEA, LMOCSO, S3CMAES with the proposed algorithm LMEA-hybrid. Table 12 shows IGD values obtained by four algorithm on LSMOP1-9 with six objectives. It

can be observed that LMOCSO performs best on 100 and 300 decision variables (LSMOP1,3,4,5,6,7,9), while LMEA-hybrid performs better in case of 500 decision variables (LSMOP1,2,3,4,6,7,8). Table 13 shows GD values of four algorithms on LSMOP1-9. It can be observed that LMOCSO performs better on LSMOP(3,6), whereas LMEA-hybrid is performing best on 12 test instances out of 27.

5 Conclusion

With an increase in the number of objectives, Pareto dominance becomes less efficient, such that most solutions end up being nondominated. This calls for algorithms that can efficiently solve LSMaOPs with a large number of decision variables and objectives.

To overcome this difficulty, we performed a thorough comparative study of three cutting-edge LSMaOEAs—LMEA, LMOCSO, and S3CMAES—over decision variable ranges of 100 to 1000, with uniform objective counts. Their performance was measured by four important metrics: IGD, GD, HV, and runtime. The results showed that LMOCSO performs consistently well for the majority of test cases, showing scalability and robustness on benchmark suites like DTLZ, LSMOP, WFG, and UF.

As part of this study, we also made a new variant—Hybrid-LMEA—that combines decision variable clustering with competitive learning dynamics. The hybrid model maintains a balance of the convergence-diversity tradeoff by retaining structure during exploitation and encouraging exploration by diversity-aware selection. It clearly demonstrated performance benefits on LSMOPs with more than 500 decision variables, where it improves both IGD and GD metrics consistently.

In future research, we intend to compare the performance of Hybrid-LMEA on actual optimization problems, for example, software module clustering, in order to further investigate its potential utility.

Table 12: IGD values (Mean and standard deviation) of four algorithms on LSMOP (1-9) using Wilcoxon signed-rank test.

| Problem | M | D | LMEA | LMOCSSO | S3CMAES | LMEA_hybrid |
|---------|---|-----|----------------------------------|----------------------------------|----------------------------------|------------------------|
| LSMOP1 | 6 | 100 | 7.8505e + 0(1.18e + 0) \approx | 1.1162e+0 (2.20e-1) + | 9.3382e + 0(6.24e + 0) - | 7.1491e + 0(1.51e + 0) |
| | | 300 | 9.7993e + 0(8.53e - 1) - | 1.9560e+0 (4.79e-1) + | 3.4866e + 1(2.53e + 1) - | 7.4620e + 0(7.95e - 1) |
| | | 500 | 1.0371e + 1(6.96e - 1) - | 1.0365e + 1(1.03e + 0) - | 5.7233e + 1(3.93e + 1) - | 7.3197e+0 (6.28e-1) |
| LSMOP2 | 6 | 100 | 6.5253e - 1(3.89e - 2) - | 4.9476e-1 (2.38e-2) + | 5.6671e - 1(5.79e - 2) \approx | 5.8248e - 1(3.36e - 2) |
| | | 300 | 3.8340e - 1(2.62e - 2) - | 2.8391e - 1(2.77e - 2) \approx | 1.1558e + 0(5.50e - 1) - | 2.7352e-1 (7.03e-3) |
| | | 500 | 3.2267e - 1(2.41e - 2) - | 3.4882e - 1(1.60e - 2) - | 9.9936e - 1(3.57e - 1) - | 2.3563e-1 (7.04e-3) |
| LSMOP3 | 6 | 100 | 5.1876e + 3(2.21e + 3) - | 1.5982e+1 (2.25e+1) + | 6.9334e + 2(1.34e + 3) + | 3.0315e + 3(1.68e + 3) |
| | | 300 | 8.9438e + 3(4.77e + 3) \approx | 5.5185e+1 (9.98e+1) + | 2.2229e + 5(3.51e + 5) - | 6.2121e + 3(2.65e + 3) |
| | | 500 | 9.1783e + 3(4.15e + 3) \approx | 1.2884e + 4(7.44e + 3) - | 2.3369e + 5(3.74e + 5) - | 6.7433e+3 (2.93e+3) |
| LSMOP4 | 6 | 100 | 7.8327e - 1(8.90e - 2) + | 5.5492e-1 (4.99e-2) + | 9.6423e - 1(1.82e - 1) \approx | 9.2827e - 1(1.43e - 1) |
| | | 300 | 4.9135e - 1(3.24e - 2) - | 3.8595e-1 (4.89e-2) + | 1.6182e + 0(8.10e - 1) - | 4.6084e - 1(3.48e - 2) |
| | | 500 | 4.1160e - 1(2.53e - 2) - | 3.9979e - 1(7.76e - 2) \approx | 1.0240e + 0(4.33e - 1) - | 3.6192e-1 (2.41e-2) |
| LSMOP5 | 6 | 100 | 1.2874e + 1(2.03e + 0) - | 2.1034e+0 (5.56e-1) \approx | 1.9173e + 1(4.15e - 1) - | 3.3727e + 0(2.09e + 0) |
| | | 300 | 1.5523e + 1(1.31e + 0) - | 4.4084e+0 (9.32e-1) + | 4.6207e + 1(3.61e + 1) - | 1.1137e + 1(2.57e + 0) |
| | | 500 | 1.6260e + 1(7.47e - 1) - | 1.1221e+1 (6.08e+0) \approx | 6.1219e + 1(3.92e + 1) - | 1.1688e + 1(1.55e + 0) |
| LSMOP6 | 6 | 100 | 4.3009e + 1(5.91e + 1) - | 1.6894e+0 (2.19e-1) + | 2.6136e + 0(3.26e - 1) \approx | 2.4560e + 0(2.33e - 1) |
| | | 300 | 1.8804e + 3(1.84e + 3) - | 1.6835e+0 (6.56e-2) \approx | 2.0585e + 5(2.41e + 5) - | 1.6849e + 0(7.88e - 2) |
| | | 500 | 1.8879e + 3(1.53e + 3) - | 7.3031e + 1(3.20e + 2) - | 2.4187e + 5(2.78e + 5) - | 1.5111e+0 (3.01e-2) |
| LSMOP7 | 6 | 100 | 1.6657e + 4(4.20e + 3) - | 3.1630e+1 (2.78e+1) + | 1.9017e + 4(8.13e + 3) - | 1.6939e + 2(1.61e + 2) |
| | | 300 | 2.5816e + 4(4.81e + 3) - | 1.7644e+3 (1.15e+3) \approx | 3.1050e + 5(3.35e + 5) - | 1.9055e + 3(1.25e + 3) |
| | | 500 | 2.8057e + 4(4.39e + 3) - | 2.9089e + 4(8.93e + 3) - | 2.8951e + 5(2.82e + 5) - | 8.1598e+3 (4.93e+3) |
| LSMOP8 | 6 | 100 | 1.0310e + 1(1.53e + 0) - | 2.3124e + 0(6.34e - 1) \approx | 1.6548e + 1(5.45e + 0) - | 2.1625e+0 (1.12e+0) |
| | | 300 | 1.1069e + 1(1.35e + 0) - | 3.7021e+0 (5.83e-1) + | 3.0094e + 1(1.67e + 1) - | 7.0942e + 0(2.38e + 0) |
| | | 500 | 1.1879e + 1(7.97e - 1) - | 1.2575e + 1(1.24e + 0) - | 3.2233e + 1(1.91e + 1) - | 8.1381e+0 (1.14e+0) |
| LSMOP9 | 6 | 100 | 3.5670e + 2(3.19e + 1) - | 4.7217e+1 (2.21e+1) + | 4.3889e + 2(7.85e + 1) - | 1.0407e + 2(1.31e + 1) |
| | | 300 | 4.0461e + 2(2.14e + 1) - | 1.1058e+2 (5.15e+1) + | 8.8449e + 2(3.87e + 2) - | 7.0942e + 0(2.38e + 0) |
| | | 500 | 4.2396e + 2(2.17e + 1) - | 1.5070e+2 (6.13e+1) + | 9.4298e + 2(3.38e + 2) - | 1.6877e + 2(2.14e + 1) |
| | | | 1 / 23 / 3 | 14 / 6 / 7 | 1 / 23 / 3 | |

Table 13: GD values (Mean and standard dDeviation) of three algorithms on LSMOP (1-9) using Wilcoxon signed- Rank test.

| Problem | M | D | LMEA | LMOCOS | S3CMAES | LMEA_hybrid |
|-----------|-----|-----|-----------------------------|-----------------------------|-----------------------------|----------------------------|
| LSMOP1 | 100 | 100 | 4.9528e+0(6.55e-1)- | 3.0575e+0(8.21e-1)+ | 1.7881e+0 (1.62e+0)+ | 3.6892e+0 (3.65e-1) |
| | 300 | 300 | 5.1742e+0(5.11e-1)- | 4.5261e+0(1.54e+0)≈ | 8.5532e+0(6.30e+0)- | 3.7694e+0 (2.73e-1) |
| | 500 | 500 | 5.2389e+0 (3.86e-1) - | 7.7224e+0(1.19e+0)- | 1.4128e+1(9.81e+0)- | 3.7437e+0 (2.13e-1) |
| LSMOP2 | 100 | 100 | 2.0471e-1(1.08e-2)- | 1.1158e-1(8.90e-3)+ | 8.1107e-2 (4.86e-3)+ | 1.5240e-1(8.77e-3) |
| | 300 | 300 | 7.4545e-2(3.94e-3)- | 4.6465e-2(7.63e-3)≈ | 1.3081e-1(9.84e-2)- | 4.3978e-2 (1.26e-3) |
| | 500 | 500 | 4.2939e-2 (2.93e-3)- | 6.7061e-2(8.24e-3)- | 7.8811e-2(5.31e-2)- | 2.5300e-2 (8.42e-4) |
| LSMOP3 | 100 | 100 | 4.4162e+4(8.99e+3)+ | 4.0139e+2 (8.10e+2)+ | 7.5513e+3(2.21e+3)+ | 5.3278e+4(9.20e+3) |
| | 300 | 300 | 7.2507e+4(8.24e+3)- | 3.9390e+3 (6.40e+3)+ | 5.5572e+4(8.77e+4)+ | 6.2057e+4(7.10e+3) |
| | 500 | 500 | 7.5216e+4(7.48e+3)- | 3.7786e+4(5.28e+4)+ | 5.8423e+4(9.36e+4)+ | 6.2963e+4(6.23e+3) |
| LSMOP4 | 100 | 100 | 3.8524e-1(2.57e-2)+ | 3.1893e-1(5.87e-2)+ | 1.1344e-1 (8.29e-2)+ | 5.2223e-1(3.04e-2) |
| | 300 | 300 | 1.3473e-1 (1.00e-2)- | 1.3485e-1(2.61e-2)≈ | 2.4017e-1(1.69e-1)≈ | 1.2202e-1 (7.76e-3) |
| | 500 | 500 | 7.6567e-2(6.30e-3)- | 1.3315e-1(4.38e-2)- | 1.0934e-1(7.95e-2)≈ | 6.7597e-2 (5.18e-3) |
| LSMOP5 | 100 | 100 | 1.1538e+1(1.85e+0)≈ | 1.6819e+1(7.42e+0)- | 1.3671e+1(6.48e+0)≈ | 9.5541e+0 (8.54e+0) |
| | 300 | 300 | 8.4363e+0(8.73e-1)+ | 1.4408e+1(3.76e+0)- | 1.1438e+1(9.03e+0)≈ | 1.2049e+1(3.26e+0) |
| | 500 | 500 | 8.0116e+0(6.60e-1)+ | 1.2601e+1(4.34e+0)≈ | 1.5188e+1(9.81e+0)≈ | 1.2628e+1(1.59e+0) |
| LSMOP6 | 100 | 100 | 2.6669e+4(5.36e+3)+ | 1.2090e+1(2.47e+1)+ | 8.5007e+4(5.97e+4)+ | 1.1011e+5(2.07e+4) |
| | 300 | 300 | 5.0053e+4(8.07e+3)+ | 3.1980e+4(4.01e+4)+ | 5.1463e+4(6.03e+4)+ | 8.7146e+4(1.80e+4) |
| | 500 | 500 | 4.8144e+4(8.84e+3)+ | 4.6665e+4 (3.76e+4)+ | 6.0468e+4(6.96e+4)+ | 9.3339e+4(1.57e+4) |
| LSMOP7 | 100 | 100 | 3.6251e+4(2.15e+4)- | 1.1068e+4(2.00e+4)- | 9.1504e+3(7.38e+3)- | 2.5470e+2 (2.28e+2) |
| | 300 | 300 | 5.1071e+4(2.03e+4)- | 5.3012e+4(5.64e+4)- | 7.7625e+4(8.38e+4)- | 2.9197e+3 (5.92e+3) |
| | 500 | 500 | 4.2243e+4(1.25e+4)- | 5.6367e+4(2.23e+4)- | 7.2377e+4(7.04e+4)- | 1.2982e+4 (2.45e+4) |
| LSMOP8 | 100 | 100 | 7.0013e+0 (1.38e+0)- | 8.9649e+0(3.51e+0)- | 7.1583e+0(3.95e+0)- | 3.9635e+0 (3.19e+0) |
| | 300 | 300 | 4.4857e+0(3.33e-1)+ | 9.0603e+0(2.34e+0)- | 7.4020e+0(4.18e+0)≈ | 5.8976e+0(2.22e+0) |
| | 500 | 500 | 4.1446e+0 (4.83e-1)+ | 7.6076e+0(6.95e-1)- | 7.9355e+0(4.78e+0)≈ | 6.2532e+0(9.69e-1) |
| LSMOP9 | 100 | 100 | 1.2912e+2(1.19e+1)+ | 4.8371e+1 (1.78e+1)+ | 6.5185e+1(1.01e+1)+ | 1.4273e+2(8.48e+0) |
| | 300 | 300 | 1.1175e+2(7.51e+0)- | 1.5240e+2(6.36e+1)- | 2.2024e+2(9.66e+1)- | 9.9064e+1(5.96e+0) |
| | 500 | 500 | 1.0975e+2 (6.53e+0) - | 1.9926e+2(7.90e+1)- | 2.3486e+2(8.45e+1)- | 9.6269e+1 (5.07e+0) |
| + / - / ≈ | | | 10 / 16 / 1 | 10 / 13 / 4 | 10 / 10 / 7 | |

Declarations

- **Funding** The authors have not disclosed any funding
- **Conflict of interest** All the authors declare that he/she has no conflict of interest.
- **Ethics approval and consent to participate** This article does not contain any studies with human participants or animals performed by any of the authors
- **Data availability** Enquiries about data availability should be directed to the authors.

References

- [1] Amarjeet and Chhabra, J.K. *Many-objective artificial bee colony algorithm for large-scale software module clustering problem*, Soft Comput., 22(19) (2018), 6341–6361.
- [2] Antonio, L.M. and Coello, C.A.C. *Use of cooperative coevolution for solving large scale multiobjective optimization problems*, Proc. IEEE Congr. Evol. Comput., (2013), 2758–2765.
- [3] Babu, B. and Jehan, M.M.L. *Differential evolution for multi-objective optimization*, Proc. Congr. Evol. Comput. 2003. CEC'03., vol. 4, pp. 2696–2703. IEEE, 2003.
- [4] Bechikh, S., Elarbi, M. and Ben Said, L. *Many-objective optimization using evolutionary algorithms: A survey*, Recent Adv. Evol. Multi-Obj. Optim., (2017), 105–137.
- [5] Brockhoff, D. and Zitzler, E. *Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization*, Proc. Int. Conf. Parallel Prob. Solving Nature, (2006), 533–542.

- [6] Cao, B., Fan, S., Zhao, J., Tian, S., Zheng, Z., Yan, Y. and Yang, P. *Large-scale many-objective deployment optimization of edge servers*, IEEE Trans. Intell. Transp. Syst., 22(6) (2021), 3841–3849.
- [7] Cao, B., Zhang, Y., Zhao, J., Liu, X., Skonieczny, L. and Lv, Z. *Recommendation based on large-scale many-objective optimization for the intelligent internet of things system*, IEEE Internet Things J., 9(16) (2021), 15030–15038.
- [8] Cao, B., Zhao, J., Lv, Z., Liu, X., Yang, S., Kang, X. and Kang, K. *Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization*, IEEE Access, 5 (2017), 8214–8221.
- [9] Chand, S. and Wagner, M. *Evolutionary many-objective optimization: A quick-start guide*, Surv. Oper. Res. Manage. Sci., 20(2) (2015), 35–42.
- [10] Chen, H., Cheng, R., Wen, J., Li, H. and Weng, J. *Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations*, Inf. Sci., 509 (2020), 457–469.
- [11] Cheng, R., Jin, Y., Olhofer, M. and Sendhoff, B. *A reference vector guided evolutionary algorithm for many-objective optimization*, IEEE Trans. Evol. Comput., 20(5) (2016), 773–791.
- [12] Cheng, R., Jin, Y., Olhofer, M. and Sendhoff, B. *Test problems for large-scale multiobjective and many-objective optimization*, IEEE Trans. Cybern., 47(12) (2016), 4108–4121.
- [13] Cheng, R., Rodemann, T., Fischer, M., Olhofer, M. and Jin, Y. *Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation*, IEEE Trans. Emerg. Topics Comput. Intell., 1(2) (2017), 97–111.
- [14] Deb, K. and Agrawal, R.B. *Simulated binary crossover for continuous search space*, Complex Syst., 9(2) (1995), 115–148.

- [15] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Trans. Evol. Comput., 6(2) (2002), 182–197.
- [16] Deb, K., Sindhya, K. and Hakanen, J. *Multi-objective optimization*, in Decision Sciences, CRC Press (2016), 161–200.
- [17] Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. *Scalable test problems for evolutionary multiobjective optimization*, in Evolutionary Multi-objective Optimization: Theoretical Advances and Applications, Springer (2005), 105–145.
- [18] Fleming, P.J., Purshouse, R.C. and Lygoe, R.J. *Many-objective optimization: An engineering design perspective*, Proc. Int. Conf. Evol. Multi-Criterion Optim., (2005), 14–32.
- [19] Gu, Z.M. and Wang, G.G. *Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization*, Future Gener. Comput. Syst., 107 (2020), 49–69.
- [20] Harman, M. and Yao, X. *Software module clustering as a multi-objective search problem*, IEEE Trans. Softw. Eng., 37(2) (2010), 264–282.
- [21] He, C., Cheng, R., Li, L., Tan, K.C. and Jin, Y. *Large-scale multiobjective optimization via reformulated decision variable analysis*, IEEE Trans. Evol. Comput., 28(1) (2022), 47–61.
- [22] He, C., Li, L., Tian, Y., Zhang, X., Cheng, R., Jin, Y. and Yao, X. *Accelerating large-scale multiobjective optimisation via problem reformulation*, IEEE Trans. Evol. Comput., 23(6) (2019), 949–961.
- [23] Hong, H., Ye, K., Jiang, M., Cao, D. and Tan, K.C. *Solving large-scale multiobjective optimization via the probabilistic prediction model*, Memetic Comput., 14(2) (2022), 165–177.
- [24] Li, B., Li, J., Tang, K. and Yao, X. *Many-objective evolutionary algorithms: A survey*, ACM Comput. Surv., 48(1) (2015), 1–35.

- [25] Li, H. and Zhang, Q. *Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II*, IEEE Trans. Evol. Comput., 13(2) (2008), 284–302.
- [26] Li, K., Wang, R., Zhang, T. and Ishibuchi, H. *Evolutionary many-objective optimization: A comparative study of the state-of-the-art*, IEEE Access, 6 (2018), 26194–26214.
- [27] Liu, Q., Zou, J., Yang, S. and Zheng, J. *A multiobjective evolutionary algorithm based on decision variable classification for many-objective optimization*, Swarm Evol. Comput., 73 (2022), 101108.
- [28] Liu, R., Ren, R., Liu, J. and Liu, J. *A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems*, Appl. Soft Comput., 89 (2020), 106120.
- [29] Ma, L., Huang, M., Yang, S., Wang, R. and Wang, X. *An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization*, IEEE Trans. Cybern., 52(7) (2021), 6684–6696.
- [30] Ma, X., Liu, F., Qi, Y., Wang, X., Li, L., Jiao, L., Yin, M. and Gong, M. *A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables*, IEEE Trans. Evol. Comput., 20(2) (2015), 275–298.
- [31] Maltese, J., Ombuki-Berman, B.M. and Engelbrecht, A.P. *A scalability study of many-objective optimization algorithms*, IEEE Trans. Evol. Comput., 22(1) (2016), 79–96.
- [32] Miguel Antonio, L. and Coello Coello, C.A. *Decomposition-based approach for solving large scale multi-objective problems*, Proc. Parallel Prob. Solving Nature (PPSN XIV), (2016), 525–534. Springer.
- [33] Okola, I., Omulo, E.O., Ochieng, D.O. and Ouma, G. *A comparison of evolutionary algorithms on a large scale many-objective problem in food–energy–water nexus*, Results Control Optim., 10 (2023), 100195.

- [34] Pan, X., Wang, L., Qiu, Q., Qiu, F. and Zhang, G. *Many-objective optimization for large-scale EVs charging and discharging schedules considering travel convenience*, Appl. Intell., 52(3) (2022), 2599–2620.
- [35] Prajapati, A. *A comparative study of many-objective optimizers on large-scale many-objective software clustering problems*, Complex Intell. Syst., 7(2) (2021), 1061–1077.
- [36] Prajapati, A. *A customized PSO model for large-scale many-objective software package restructuring problem*, Arab. J. Sci. Eng., 47(8) (2022), 10147–10162.
- [37] Prajapati, A. *A particle swarm optimization approach for large-scale many-objective software architecture recovery*, J. King Saud Univ. Comput. Inf. Sci., 34(10) (2022), 8501–8513.
- [38] Prajapati, A. *Software module clustering using grid-based large-scale many-objective particle swarm optimization*, Soft Comput., 26(17) (2022), 8709–8730.
- [39] Prajapati, A. and Chhabra, J.K. *Madhs: Many-objective discrete harmony search to improve existing package design*, Comput. Intell., 35(1) (2019), 98–123.
- [40] Purshouse, R.C. and Fleming, P.J. *Evolutionary many-objective optimisation: An exploratory analysis*, Proc. Congr. Evol. Comput., 3 (2003), 2066–2073.
- [41] Riquelme, N., von Lüken, C. and Baran, B. *Performance metrics in multi-objective optimization*, Proc. Latin Amer. Comput. Conf. (CLEI), (2015), 1–11.
- [42] Saxena, D.K. and Deb, K. *Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective*, Proc. IEEE Congr. Evol. Comput., (2008), 3204–3211.
- [43] Tian, Y., Si, L., Zhang, X., Cheng, R., He, C., Tan, K.C. and Jin, Y. *Evolutionary large-scale multi-objective optimization: A survey*, ACM Comput. Surv., 54(8) (2021), 1–34.

- [44] Tian, Y., Zheng, X., Zhang, X. and Jin, Y. *Efficient large-scale multiobjective optimization based on a competitive swarm optimizer*, IEEE Trans. Cybern., 50(8) (2019), 3696–3708.
- [45] Wang, Y., Zhang, Q. and Wang, G.G. *Improving evolutionary algorithms with information feedback model for large-scale many-objective optimization*, Appl. Intell., 53(10) (2023), 11439–11473.
- [46] Xu, Y., Xu, C., Zhang, H., Huang, L., Liu, Y., Nojima, Y. and Zeng, X. *A multi-population multi-objective evolutionary algorithm based on the contribution of decision variables to objectives for large-scale multi/many-objective optimization*, IEEE Trans. Cybern., 53(11) (2022), 6998–7007.
- [47] Zhang, J., Wei, L., Fan, R., Sun, H. and Hu, Z. *Solve large-scale many-objective optimization problems based on dual analysis of objective space and decision space*, Swarm Evol. Comput., 70 (2022), 101045.
- [48] Zhang, K., Shen, C. and Yen, G.G. *Multipopulation-based differential evolution for large-scale many-objective optimization*, IEEE Trans. Cybern., 53(12) (2022), 7596–7608.
- [49] Zhang, Q. and Li, H. *MOEA/D: A multiobjective evolutionary algorithm based on decomposition*, IEEE Trans. Evol. Comput., 11(6) (2007), 712–731.
- [50] Zhang, X., Tian, Y., Cheng, R. and Jin, Y. *A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization*, IEEE Trans. Evol. Comput., 22(1) (2016), 97–112.
- [51] Zhang, Y., Wang, G.G., Li, K., Yeh, W.C., Jian, M. and Dong, J. *Enhancing MOEA/D with information feedback models for large-scale many-objective optimization*, Inf. Sci., 522 (2020), 1–16.
- [52] Zhou, Y., Kong, L., Cai, Y., Wu, Z., Liu, S., Hong, J. and Wu, K. *A decomposition-based local search for large-scale many-objective vehicle routing problems with simultaneous delivery and pickup and time windows*, IEEE Syst. J., 14(4) (2020), 5253–5264.

- [53] Zille, H. *Large-scale multi-objective optimisation: New approaches and a classification of the state-of-the-art*, PhD thesis, Otto von Guericke Univ. Magdeburg, 2019.
- [54] Zille, H., Ishibuchi, H., Mostaghim, S. and Nojima, Y. *A framework for large-scale multiobjective optimization based on problem transformation*, IEEE Trans. Evol. Comput., 22(2) (2017), 260–275.
- [55] Zille, H. and Mostaghim, S. *Comparison study of large-scale optimisation techniques on the LSMOP benchmark functions*, Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), (2017), 1–8.
- [56] Zitzler, E. *SPEA2: Improving the performance of the strength Pareto evolutionary algorithm*, Tech. Rep., Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology (ETH Zurich) 2001.



Combining an interval approach with a heuristic to solve constrained and engineering design problems

D. Sharma*,  and S.D. Jabeen

Abstract

Solving intricate constrained optimization problems with nonlinear constraints is usually difficult. To optimize the constraint and structure engineering design challenges, this work presents a novel hybrid method called SDDS-SABC, which is based on the split-detect-discard-shrink technique and the Sophisticated ABC algorithm inspired by the integration of branch-and-bound-like concepts of interval analysis with heuristics, and it differs from other methods in the literature. The advantage of the SDDS process is that it shrinks the entire search region through recursive breakdown and improves computational effort to focus on subregions covering potential solutions for further decomposition. In order to identify the most promising

*Corresponding author

Received 6 May 2025; revised 8 August 2025; accepted 4 September 2025

Dhirendra Sharma

Department of Mathematics, Birla Institute of Technology Mesra, Ranchi 835215, India,
e-mail: dhirendrasharma428@gmail.com

Syeda Darakhshan Jabeen

Department of Mathematics, Birla Institute of Technology Mesra, Ranchi 835215, India,
e-mail: syed_sdj@yahoo.co.in

How to cite this article

Sharma, D. and Jabeen, S.D., Combining an interval approach with a heuristic to solve constrained and engineering design problems. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1538-1588. <https://doi.org/10.22067/ijnao.2025.93363.1640>

subregion, SABC's values are crucial in assisting in the extraction of the best solutions from the subregions. Until the region shrinks to a nominal width that represents the global or nearly global solution(s) to the optimization problem, both SDDS and SABC are successively repeated. The selection and rating criteria are used to support positive decision-making, with the mindset of removing the subregion containing the unpromising solution(s). Simultaneously, the subregion exhibiting a viable solution is acknowledged as the present shrink region in anticipation of a subsequent split. We present a new initialization technique for food sources in the SABC algorithm, called the quasi-random sequence-based Halton set, which outperforms the current initialization procedure. Create a composite strategy that uses the employed bee phase to investigate their neighborhood while preserving their cooperative nature. In order to increase the optimization efficiency, we also present a new dynamic penalty approach that does not rely on any additional characteristics or factors like the majority of existing penalty methods. We test the statistical validity of SDDS-SABC by applying it to engineering design problems and benchmark functions (CEC 2006). The results demonstrate that SDDS-SABC performs better than its most studied competitors and proves its viability in resolving difficult real-life problems. Additionally, the SDDS-SABC approach is appropriate and numerically stable for the optimization problems. The main innovation of the approach being described is its capacity to perform a static and better optimal solution in the majority of runs, even when the problem is excessively complex.

AMS subject classifications (2020): 65K10, 90C26, 90C31, 90C59

Keywords: Constrained optimization, ABC algorithm, Dynamic penalty function, Engineering design problem

1 Introduction

Optimization is a numerical approach to solving practical problems in a variety of domains and is a difficult decision-making process that keeps getting difficult. The goal of decision-making is to, given the situation, determine the best set of variables to combine in order to maximize or decrease the objective function within the specified bounds. Sustainable restrictions mean that

it is usually more difficult to solve scientific and engineering problems when the search space structure is limited to regions that are both feasible and nonfeasible. This makes it challenging to extract the best feasible solution from a subset of the feasible space. Constrained optimization problems are what these problems are known as (COPs). In general, typical mathematical programming techniques found in literature cannot address nonconvex or discrete problems since they require gradient information in order to find optimal solutions. They are also vulnerable to starting points. When the optimization problems feature several or impulsive peaks, selecting the initial points incorrectly makes the search for the global optimum difficult and unstable. Furthermore, as the dimension of the choice variable increases and optimization problems become extremely nonlinear, conventional methods break down if the problem's complexity rises any further. Using sophisticated, effective algorithms that possess derivation-free formulations, simplicity, and flexibility is the best option. These algorithms can offer excellent results in various real-world optimization scenarios. Numerous nature-inspired intelligent optimization strategies and constraint-handling methodologies have been developed to address these kinds of problems [2]. Particle swarm optimization (PSO) [27], ant colony optimization (ACO) [13], artificial bee colony (ABC) [26], grey wolf optimizer (GWO) [41], and other swarm-based algorithms are popular. The Darwinian evolution theory underpins evolutionary algorithms such as genetic algorithm (GA) [42], differential evolution (DE) [48], and Memetic algorithm [15], among others. The simulated annealing (SA) [4], gravitational search algorithm (GSA) [50], big-bang big-crunch algorithm (BBBC) [52], and other chemistry- or physics-based methods are inspired by chemical or physical phenomena. The last type of algorithms is social or human-based ones, such as the arithmetic optimization algorithm (AOA) [1], teaching learning-based optimization (TLBO) [54], and brain storm optimization (BSO) [57]. These algorithms are inspired by human or social behaviors. These algorithms' drawbacks include their propensity for local convergence, need for parameter tweaking, and so on. They perform so poorly in the majority of optimization problems. Furthermore, the majority of these techniques is designed to address unconstrained optimization problems. Several techniques for handling constraints are integrated into these algorithms to

handle restricted optimization problems. During the algorithm's iterative process, these techniques direct the population of solutions towards the more feasible region. Furthermore, the aforementioned significant shortcomings of the heuristic algorithms prompted the necessity of creating a more sophisticated version of the algorithms in terms of computing efficiency and solution quality. Therefore, obtaining more robust algorithms, fusing local search methods combined with additional heuristic approaches, or combining multiple heuristic methods captured the researcher's interest to form hybrid algorithms (see [14, 2, 58, 20, 21]). Hybridization of PSO with GWO [55], GA with GSA [16], Cuckoo Search (CS) with DE [67], hybrid firefly algorithm with grouping attraction (HFA-GA) [8], an improved firefly algorithm (UFA) [6], an enhanced leadership-based GWO (GLF-GWO) [19], GGA with the gradient-descent (GD) [10], PSO with DE [32], and many more have been recently proposed hybrid algorithms. Additionally, other approaches to managing constraints have been proposed, such as (1) penalty function methods [64, 60] (2) handling the goal function and constraints independently [38, 45, 12, 53, 59] (3) Hybrid approaches [45, 38] and (4) multiobjective-optimization methods [62, 23]. Using penalty functions, such as the death penalty, static penalties, dynamic penalties, annealing penalties, adaptive penalties, and co-evolutionary penalties, is a common and simple method of handling constraints. These functions convert restricted problems into unconstrained ones. The majority of these constraint handling strategies have significant disadvantages. Certain methods may yield an unfeasible solution or necessitate numerous more factors with uncertain values; still others are situation-specific, meaning that a special approach must be developed for a given problem. By lowering their fitness values in proportion to the degrees of constraint violation, these techniques penalize impractical solutions. Furthermore, the majority of these penalty systems have parameters that need to be carefully experimented with in order to determine the proper values in order to produce workable solutions. Certain problems may respond well to the specified parameter values, while other problems may not respond well to them. In order to address this reliance of algorithm performance on penalty parameters, scholars have developed advanced penalty function methodologies.

It is worth questioning what the better sampling methods are that could be employed for generating initial solutions in evolutionary algorithms and what the appropriate penalty function value is to improve the advancement of solutions towards feasibility. It is usually more difficult to solve scientific and engineering problems when the search space structure is limited to regions that are both feasible and nonfeasible. This makes it challenging to extract the best feasible solution from a subset of the feasible space. The objective of this study is to present a hybrid optimization approach combining an interval approach with the ABC algorithm to solve the benchmark constrained optimization and engineering design problems. The aims of this work are to implement a new initialization method in the ABC algorithm and formulate a new dynamic penalty function formula to handle the constraints of the benchmark optimization problems.

The No Free Lunch idea is supported by the fact that, despite the fact that numerous heuristic algorithms have been created; they are all plagued by the inability to effectively address every optimization problem that is presented to us. Furthermore, research demonstrated that certain algorithms yield more optimal outcomes than others. Thus, creating an enhanced heuristic method for various optimization issues remains an unresolved matter and is greatly appreciated by scholars, provided that they provide a noteworthy contribution to the domain. This encourages us to present a novel, efficient heuristic algorithm for solving COPs that differs from the one seen in the literature.

We present a novel hybrid optimization technique in this work, termed “(SDDS-SABC)”, that combines two phases: Phase 1 involves the split-detect-discard-shrink (SDDS) strategy, and Phase 2 involves the sophisticated artificial bee colony (SABC) algorithm being executed. The fundamental principle of the SDDS is to divide the whole of n -dimensional Euclidean space into two subregions of a specific form by first splitting near the first variable’s midpoint on its axis. Every subregion has undergone the SABC phase in order to identify and eliminate any subregions that cover unpromising solutions. By analyzing the two solutions that SABC created in each subregion and selecting the subregion with the most promising solution, the interval arithmetic rule has been utilized to identify the subregions that

can be rejected. The hybrid algorithm's first cycle is now finished. Through repetitive switching of the SDDS and SABC phases, the approach explores the search for the region of promise. The i th variable's axis is split about the chosen subregion's midpoint in n -dimensional Euclidean space during the i th cycle SDDS phase. During each splitting, the selected subregion is split into two subregions, and ABC is implemented for every subregion. The procedure for splitting, detecting, discarding, and shrinking the chosen subregion is repeated several times using SDDS and SABC, concentrating computing effort on the promising subregion each time. When the reduced zone approaches negligible width, the cycle count comes to an end. This condition means that the global optimum, or something close to it, has been identified. This improves the search space's exploitation potential while enhancing the ability to explore high-quality solutions and combine algorithmic strength. A variety of numerical test issues and engineering design difficulties have been resolved in order to assess the effectiveness of the suggested methodology. The Friedman and Wilcoxon test was used to compare the suggested SDDS-SABC algorithm to other cutting-edge algorithms.

Our results demonstrate that for most benchmark functions in the domain of wide and restricted search spaces. We find that our hybrid methodology outperforms the majority of recently developed techniques.

2 Novelty and contributions of our proposed method

We propose a novel hybrid algorithm called SDDS-SABC to address the limitations of the existing heuristic algorithms and the penalty function. In the following paragraphs, we highlight our paper's novelty and contributions.

- a. In this paper, we provide a novel hybrid algorithm that combines the ideas of interval analysis and heuristics to solve intricate restricted optimization problems. The goal of this integration is to increase algorithms' robustness, convergence rate, and solution quality while guiding them toward an efficient, effective, and robust search.
- b. We present a quasi-random formulation for initialization after recognizing the drawbacks of random, chaotic, or logistic initialization of the food source. In this case, we create an initial population that is more uniformly dispersed

over the search area, which may lead to more reliable results.

c. To increase the algorithm's adaptability, the composite strategy is employed the bee stage to introduce a two-way search space exploration.

d. We present a new dynamic penalty approach that is straightforward in form and does not require any additional parameters or penalty factors in order to address the drawbacks of the current penalty factors. In order to increase the optimization efficiency, the burden of fine-tuning the penalty factors and parameters has been eliminated. It has not yet been documented in the literature that these three key features were used in the heuristic algorithm's creation.

e. Analyze our proposed hybrid algorithm's effectiveness through comprehensive numerical testing on benchmark CEC 2006 and some engineering design problems in MATLAB.

f. Compare our SDDS-SABC hybrid algorithm with other state-of-the-art algorithms using the Friedman and Wilcoxon test and demonstrate its statistical performance.

g. Our results demonstrate that our method works well for the majority of benchmark optimization problems in the domain of wide and narrow search spaces. The proposed approach performs more accurately than the most recent methods.

3 An overview of the hybrid ABC algorithm

Karaboga introduced the ABC algorithm, a nature-inspired stochastic optimization technique based on swarm intelligence, in 2005 (see[24]). The clever way that honey bees look for food and communicate that knowledge to other bees in their hive has served as inspiration for the algorithm. The technique was originally designed to solve unconstrained optimization problems. On the one hand, its capacity to resolve a wide range of multidimensional and multimodal real-world optimization problems has drawn a lot of interest since its inception. However, it also had some significant drawbacks, including a slow rate of convergence, inept exploration, limited exploitation, and a propensity to become trapped in local optima. The algorithm has been found to be better than other algorithms despite its drawbacks due of its adaptabil-

ity, simplicity, resilience, and requirement for a smaller number of training parameters. Therefore, it is easier to combine it with multiple algorithms. Considering its benefits and shortcomings, academics have been inspired to expand, alter, or combine ABC with different population-based algorithms or traditional techniques in order to improve its efficiency. The study expanded the scope of hybrid ABC algorithm development beyond numerical COPs to include a broad spectrum of application-based problem optimization.

For example, in order to deal with restrictions, Karaboga and Akay [25] devised an updated ABC to solve COPs, applying Deb's feasibility-based tournament selection operator criteria [25]. Changes have been made to ABC's scout bee operator and selection mechanism by Mezura-Montes and Cetina-Domínguez [40]. They dealt with using the search area confined by the equality and inequality criteria by using the dynamic tolerance property and tournament selection. In order to improve exploitation, Brajevic [5] suggested updating the ABC algorithm and changing the phases of the employed and scout bees. Deb's feasibility-based principles helped them keep the limitations under control. Once more, Li and Yin [28] have presented a self-adaptive constrained ABC algorithm (SACABC) based on the feasible rule approach and multiobjective optimization technique. The employed bees produced better results in their method by using the new search scheme that adheres to the feasible rule. To further investigate the new search area, the observer bees employed an improved search approach based on the multiobjective optimization problem technique. Brajevic and Tuba [7] proposed an upgraded ABC algorithm and modified employed and scout bee's phases for better exploitation. They used Deb's feasibility-based rules to manage the constraints. Furthermore, Brajevic [5] has presented a new version of the crossover-based ABC method, called CBABC, to solve constrained optimization issues.

Two distinct formulas for inequality and equality constraints were established in order to address border restrictions and dynamic tolerance. Conversely, Deb's feasibility-based rules have been loosened in the improved ABC (IABC) algorithm suggested [30] by approximating feasible solutions to a better objective function value with a slight violation. Inspired by the gbest-guided ABC (GABC) method, they have also developed a new search

technique to improve exploitation, utilizing the most optimal solution's data. It has been pointed out by Liu et al. [33] that Deb's feasibility-based rules may result in premature convergence, especially for the problems with an equality constraint. In order to address limitations, Long et al. [37] have developed a unique constrained optimization technique called IABC-MAL. This method combines the advantages of the modified augmented Lagrangian (MAL) method with IABC algorithm capability for achieving the global optimum. The first attempt to combine the augmented Lagrangian approach and the ABC algorithm is presented in this publication. For restricted optimization problems, Bansal, Joshi, and Sharma [3] suggested modifying GABC (MGABC). In their work, GABC [68] is adjusted by introducing the idea of fitness probability-based individual mobility in both the employed and onlooker bee phases. This inspired them to suggest a brand-new dynamic penalty function and an ABC-based Levy flight algorithm (DPLABC) for resolving the COPs.

To expedite the local search, they have used a dynamic logistic map in conjunction with the Levy flying technique with the used bee phase. Wang and Kong [63] have discussed the enhanced artificial bee colony (EABC) algorithm and its application in solving optimization problems. The algorithm is compared to other variants of the ABC algorithm on various test functions and engineering optimization problems. Phoemphon [46] has introduced grouping and reflection of the artificial bee colony, a distinctive adaptation of the traditional ABC algorithm meticulously tailored to meet the specific demands of high-dimensional numerical optimization problems by balancing exploration and exploitation processes. Patra et al. [44] have presented an efficient multi-objective optimization approach utilizing the ABC algorithm for minimizing generation fuel cost and transmission loss through the optimal placement and sizing of flexible AC transmission system (FACTS) controllers. Liu et al. [36] have developed a learning-based ABC algorithm by integrating deep reinforcement learning for operation optimization in gas pipelines. In addition to the aforementioned algorithms, some hybrid ABC algorithms were specially created by researchers to address real-world application problems in the fields of economics (ABC with CMA-ES [66]), industrial engineering, electrical engineering [39], and mechanical engineering (ABC with LS-SVM

[18]); inventory model (ABC with GA [47]; DE with ABC, [9]; scheduling (HABC [17]); cluster analysis (PSO-ABC [49]); routing problem [51, 22]; and wireless network [61, 65, 43].

4 Sophisticated artificial bee colony (SABC) algorithm

Based on the fundamentals of Karaboga's ABC algorithm, we provide a SABC algorithm that includes changes to the initialization procedure, used, and scout bees search approach, all of which we will cover in the upcoming subsections. According to the idea, one potential solution to the optimization problem is to locate the food sources. The associated solution's fitness and the objective function are represented by the amount of nectar present in the food supply. Finding the food source with the most nectar is the goal (optimal solution). The employed bee, spectator bee, and scout bee are the three groups into which the SABC algorithm divides the bees in order to do this. There are an equal number of food sources, working bees, and bystander bees. Each bee group's participation is crucial for producing higher-quality honey. In order to reach the optimum, the mathematical formula used by the bees in a new food location update must be sufficiently competitive. In order to draw in other interested bees, employed bees search for food sources and disseminate information about them. Assuming a probability related to the quality of the food sources, observer bees follow and utilize the food sources found by all working bees. The hired bees, known as scouts, abandon a food source and look for other sources if a solution matching that food supply is not improved by a particular number of limits. Each step of the SABC process is explained in depth in the algorithm below:

Algorithm-1:

begin

Define it, MaxIt and MNC as current iteration numbers, maximum iteration number and maximum number of cycle, respectively.

Compute initial population of probable solutions $x_{i,j}$ of popsize SN ($i = 1, 2, \dots, SN, j = 1, 2, \dots, n$) using our proposed quasi-random method (see Sec 4.2) and calculate their fitness value.

repeat

begin Employed bee's stage:

The Employed bee's, produce new solutions $v_{i,j}$ using our proposed strategy search techniques. (see Sec. 4.3).

Apply greedy selection

Memorise the best solution achieved so far

end

begin Onlooker bee's stage:

Compute the probability value for the new found i th solution using the formula:

$$p_i = \frac{fit_i(x_i)}{\sum_{i=1}^{SN} fit_i(x_i)},$$

where

$$fit_i(x_i) = \begin{cases} 1 + |f(x_i)| & \text{if } f(x_i) \leq 0, \\ \frac{1}{1+f(x_i)} & \text{if } f(x_i) \geq 0. \end{cases}$$

is the fitness value of the i th solution and $f(x_i)$ is the objective function value of the solution x_i .

Using Roulette wheel selection to produce a new solution:

$$v_{i,j} = x_{i,j} + \sin\left(i - \frac{it}{MaxIt}\right) * (x_{i,j} - x_{k,j}) \quad (i \neq j)$$

Compute the fitness value and apply greedy selection

end

begin Scout bee's stage:

if $x_{i,j}$ remains same till max limit is reached abandon $x_{i,j}$ using our proposed scout bees formula (see sec. 4.4)

end

Compute the fitness of new found solutions

Memorize the best found solution achieved so far

end

Update the best found solution.

Until

predefined MNC is reached.

4.1 ABC parameters and the effect of the algorithm's initial solutions

We are well aware that ABC conducts excellent exploration but subpar exploitation. Furthermore, for certain complicated functions, it can become caught in local optima and has a rapid convergence rate. The ABC Algorithm's convergence depends heavily on the parameters needed to run it; hence, they must be carefully chosen. These include the population size or popsize (SN), the maximum number of cycles (MNC), the limit value (L) for giving up the food source, and the first potential solution (s) or location of food sources (n). Selecting these parameters incorrectly can lead to pre-convergence or the convergence to an optimal solution at a higher computational cost. ABC is also a black-box optimizer. As a result, when optimizing complex functions, it is impossible to pinpoint the ideal solution's location within the problem's search space. Consequently, these solutions will be improved iteratively by the ABC optimization process's steps until a stopping condition is satisfied, regardless of how well the initial population guess turned out. Generally speaking, accurate first guesses can facilitate the algorithm's search for the optima. Conversely, if poor predictions are made at the beginning, then the algorithm might not be able to discover the global optima. In these circumstances, scientists can decide to employ a sophisticated initialization procedure to produce a diversified initial population that spreads widely and covers interesting areas of the search space that may include good local optima or potential global optima. Furthermore, by improving the methods used by employed bees, observer bees, and scout bees to produce new food sources, one can overcome the negative effects of parameters and initial solutions. In each iteration, the structured approach

needs to provide sufficient force to move those suboptimal solutions towards the optimal region.

As a result, the ABC study has made avoiding local optima and quickening the rate of convergence attractive objectives. We change the search equation of the fundamental ABC algorithm and provide a new initialization technique to address these. This improved ABC algorithm is known as the SABC algorithm.

4.2 Quasi-random sequence based food sources initialization

To enhance the quality of the current initialization procedure, we are driven to implement a new initialization method in the ABC algorithm. One of the common techniques employed by researchers is random initialization, in which food sources and/or beginning solutions are randomly chosen from a uniform distribution between the lower and upper bounds of the decision variables. On the other hand, there is little likelihood that a randomly generated population will encompass interesting areas of the search space for tiny populations. Consequently, it reduces the likelihood of discovering global optima. Conversely, if the population size is extended to encompass the whole search space region, the computing cost goes up. Another possibility is that the population becomes concentrated in a certain location as a result of repeated generation and overlap of identical solutions. In order to obtain a decent distribution of initial solutions regardless of population size, some writers focused on substituting the chaotic initialization approach based on a chaotic or logistic map for the random initialization. With less calculation time, this mapping strategy explores a superior solution and performs considerably better than the random one. However, their disadvantage is that the mapping that is employed is dependent on a chaotic/bifurcation parameter, the values of which must be carefully chosen by the user and vary depending on the situation.

Using a stratified-random approach called quasi-random sequence, we create a parameter-free and more equally distributed beginning population in

our algorithms, which could yield more accurate results than the methods mentioned previously. To determine if using quasi-random sequences with the Halton set in the beginning population would result in a better value for the objective function at the end is our goal. A GA's starting population is typically referred to as random. However, it is a well-known fact that algorithms cannot produce random numbers. Commonly used algorithmically produced numbers simply attempt to mimic random numbers. More precisely, they are known as pseudorandom numbers. We refer to numbers that are genuinely independent as "genuine random numbers" in order to distinguish them from pseudorandom numbers. Quasi-random sequences are another type. A quasi-random sequence's points are arranged to keep as far away from one another as possible. Stated differently, the points produced by quasi-random sequences attempt to mimic points with a "perfect uniform distribution," whereas the points produced by pseudorandom numbers attempt to mimic actual random points. The former is unachievable, whereas the latter is highly challenging if not impossible. Large partitions that are not needed, however, will raise the cost of computing.

The Halton set initialization method based on quasi-random sequences is computed and discussed in *Algorithm-2* below.

Algorithm-2:

begin

*define the n -dimensional decision variable $x_i \in [x_{\min}, x_{\max}]$ ($i = 1, 2, \dots, n$)
& $x_{\min} < x_{\max}$, x_{\min} and x_{\max} may or may not be the same for each x_i*

Set $i = 1$

Set $SN =$ The number of food sources that will be accessible to bees

repeat

(i) define halton set object Q that contains n -dimensional decision variable x_i points

(ii) each $P(i, :)$ is a point in a Halton sequence, the j th coordinate of the point,

$P(i, j)$ is equal to $\sum_{k=1}^{\infty} a_{ij}(k)b_j^{-k-1}$, where b_j is the j th prime number

(iii) $a_{ij}(k)$ coefficients are nonnegative integers less than b_j such that

$$i - 1 = \sum_{k=0}^{\infty} a_{ij}(k) b_j^k$$

i.e. the $a_{ij}(k)$ values are the base b_j digits of the integer $i - 1$.

(iv) generate random variable Hal_i^r $r = 1, 2, \dots, SN$ using the formula

$$Hal_i^r = x_{\min} + Q \cdot (x_{\max} - x_{\min}), \text{ where } Q \text{ is Halton point set in } (0, 1)$$

(v) set $i = i + 1$

until ($i = n$)

represent $Hal_i^1, Hal_i^2, Hal_i^3, \dots, Hal_i^{SN}$ as the first, second, third, ..., SN th randomly selected food source ($i = 1, 2, \dots, n$)

end

Using quasi-random initialization, we display the widely dispersed locations of sources of food in two dimensions in Figure 1 and the deviation from the uniform random distribution in Figure 2.

From Figures 1 and 2, we can see that the quasi-random method covers the space better than all the other methods. Additionally, the quasi-random set initialization method increases the distance between the generated points, and this is also a good indicator for covering a large area in the space. On the other hand, with the uniform random method, which is the most commonly used sampling method, the generated points do not cover the whole space, and there are many gaps.

4.3 Composite-strategy for employed bees stage

Upon initializing the food sources or solutions $x_{i,j}$

($i = 1, \dots, SN, j = 1, 2, \dots, n$) of size popsize, the ABC algorithm directs the employed bees to the search zone that corresponds to the food source's position. Every bee travels to a single food source $x_{i,j}$ and learns where it is by heart. Then, in search of new food sources, hired bees start to search the area around the food sources they have committed to memory. Of course, not every bee behaves the same way when it comes to foraging, and

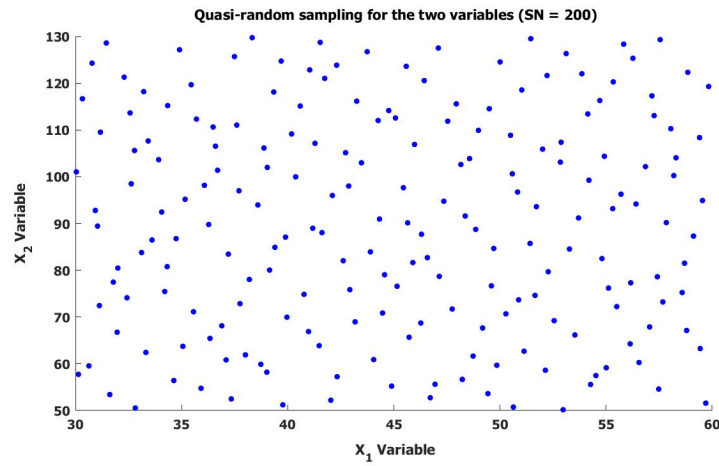


Figure 1: Quasi random sampling

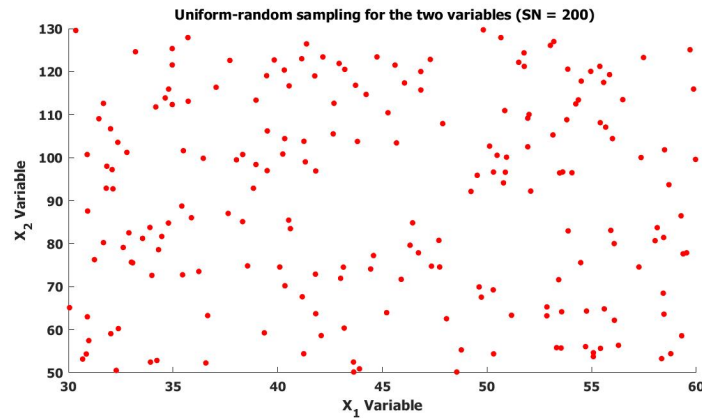


Figure 2: Uniform random sampling

individual bee behavior may vary throughout. Occasionally, their distinctly distinct behaviors serve as the foundation for developing new strategy equations for the neighborhood search process in (1). This allows them to explore their neighborhood and generate a new solution $v_{i,j}$ while

preserving their cooperative contribution.

$$v_{i,j} = \frac{x_{i,j} + x_{i,k}}{2} + \sin\left(\exp\left(i - \frac{it}{MaxIt}\right)\right) * (x_{i,j} - x_{i,k}), \quad \text{for } 1 \leq i \leq SN. \quad (1)$$

where $j, k \in \{1, 2, \dots, n\}$ are selected at random, and k and j are distinct from one another.

The bees in (1) swap out their previously learned food places, x_{ij} , for a randomly selected food position, $x_{i,k} (\neq x_{i,j})$. They then explore locally, rotating 360 degrees to create a new position, $v_{i,j}$, by moving left to right or above to downward. When deciding whether to keep the old location $x_{i,j}$ or consider the new one $v_{i,j}$, a greedy selection method is used to assess the fitness value at each step.

4.4 Scout bee's phase

We have put out a new formula for scout bees that will enhance the performance of the SABC algorithm. Here, using our suggested formula, the bees haphazardly create a new food source or solution to replace the abandoned one.

$$x_{i,j} = x_j^{\min} + P.(x_j^{\max} - x_j^{\min}), \quad (2)$$

where $P = \phi_{i,j} * \left(\frac{1}{1 + e^{\frac{j}{maxIt}}}\right)$, $\phi_{i,j} \in rand(0, 1)$, $i = 1, 2, \dots, SN$, $j = 1, 2, \dots, n$ and $MaxIt$ is the maximum number of iteration.

It is important to note that this formula aids in exploring the whole solution space, gradually shifting to the upper bound as the number of iterations increases from the lower bound.

5 Problem definition

We take a look at the COP, which is defined as follows:

$$\text{Optimize } f(x) = f(x_1, x_2, \dots, x_n),$$

$$\text{subject to } S = \{G_l(x) \leq 0, H_k(x) = 0; l = 1, 2, \dots, p; k = 1, 2, \dots, q\}, \quad (3)$$

$$\text{for all } x \in R^n.$$

The feasible region specified by a set of $p+q$ constraints is denoted by $S \subseteq D$, and the objective function $f(x)$ is defined on the search space $D \subseteq R^n$. The domain of the decision variables of the problem are defined by the n -dimensional interval vector in Euclidean space $R^n = \underline{x}_j \leq x_j \leq \overline{x}_j; j = 1, 2, \dots, n$, where x_j is the j th variable whose upper and lower bounds are \underline{x}_j and \overline{x}_j , respectively. The function $f(x)$ is not necessarily differentiable, but it might be linear, nonlinear, convex, nonconvex, and differentiable. The p equality and q inequality constraints are $G_l(x)$, $H_k(x)$, and they might be linear, nonlinear, convex, or nonconvex. In practice, inequality constraints $G_k(x) = |H_k(x)| - \varepsilon \leq 0, (k = 1, 2, \dots, q)$ are used in place of equality constraints $H_k(x) = 0, (k = 1, 2, \dots, q)$. Here, ε is a very tiny positive value. As a result, $(m+p)$ inequality restrictions replace all of the previously mentioned constraints.

In order to solve the inequality COPs using our proposed SDDS-SABC approach, we now employ the penalty function method, which is a widely used constraint handling technique appropriate for population-based optimization. The process of converting into an unconstrained one from a constrained optimization problem is the main characteristic of this approach.

When their related solutions defy the constraints, penalize the objective function by a certain amount. This allows for the preservation of workable solutions while rejecting unworkable ones. However, as neither over-nor under-penalization is desirable, determining a suitable penalty amount is a matter of interest. Many penalty methods have been proposed in the literature; each has its advantages and disadvantages. The functions for the death penalty, static penalty, and dynamic penalty approach (Joines and Houck 1994a), adaptive penalty (Yen 2009), exact penalty (Yu et al. 2010), and so on are a few examples of these techniques. The most sophisticated and widely used approach is the technique of the dynamic penalty function, which we will talk about in Section 5.1. This section introduces many variants of dynamic penalty function techniques that Liu et al. have recently developed

in the following years. In Section 5.2, we also present a novel approach using the dynamic penalty function.

5.1 Existing methods for dynamic penalty function

While there are other approaches to penalize the nonfeasible function, such as dynamic, adaptive, static, and so on, a dynamic penalty works better. Here, nonstationary values are applied at various iterations as a penalty to the unfeasible individuals. As near the feasible area inside the search space, the penalty parameter progressively increases with the number of iterations. They frequently rely on other, difficult-to-determine characteristics. We will now talk about the recently suggested dynamic penalty systems, pointing out their shortcomings and suggesting a new one to get around them.

(a) The dynamic penalty function was developed by Liu et al. [35] in 2015. It involves changing the values of the penalty parameters based on the generation number (gen). According to definitions, the penalized function is

$$\begin{aligned} F(x) &= f(x) + H(\beta, x), \\ &= f(x) + \sum_{j=1}^q p_j P_j^\beta(x) + \sum_{j=q+1}^m p_j P_j(x), \end{aligned} \quad (4)$$

where

$$P_j(x) = \begin{cases} 0 & \text{if } g_j(x) \leq 0, \\ |g_j(x)| & \text{otherwise.} \end{cases}$$

and

$$P_j(x) = \begin{cases} 0 & \text{if } -\epsilon \leq g_j(x) \leq \epsilon, \\ |g_j(x)| & \text{else.} \end{cases}$$

Here, p_j changes with generation number in the following way, and β is a constant that is selected as either $\beta = 1$ or $\beta = 2$:

$$p_j(gen) = \begin{cases} 10^{\theta_1} \cdot (1 + e^{\frac{\theta_2(\frac{G_{\max}}{2} - gen)}{G_{\max}}}), & \text{if } v_j > \epsilon, \\ 0, & \text{otherwise,} \end{cases}$$

where the tolerance for the constraint violation is ϵ , the maximum iteration number is G_{\max} , and the j th constraint violation is v_j . They made the assumptions that $gen = 500$, $\theta_1 = 3$, and $\theta_2 = 2, 4$, and 6 , respectively, in their study. They demonstrated that the value of p_j rose exponentially with generation and that it could be used for optimization purposes, both for exploration and exploitation.

(b) Subsequently, in 2016, Liu et al. [34] changed their suggested dynamic penalty function (a) and reformulated the penalized function as follows:

$$F(x) = f(x) + H(\beta, x), \quad (5)$$

where

$$p(gen) = 10^{1+\epsilon \frac{\frac{\theta_2 - \theta_1}{20(-gen + \frac{G}{4})} + \theta_1}{G}}.$$

Depending on the generation number, the dynamic penalty factor (S-type function) changes. The restricted range for this component is $[10^{\theta_1}, 10^{\theta_2}]$, where the penalty parameter's scope is indirectly defined by θ_1 and θ_2 . A smaller p will diverge in the search space early in the algorithm generation, increasing the variety of the population. With rise in generation, a more substantial p will boost the algorithm's convergence to the global optimum. They made the assumptions $\theta_1 = 2$ and $\theta_2 = 6$ in their work.

(c) Liu et al. [31] have further revised the penalty approach specified in (b). The penalized functions were expressed as follows:

$$F(x) = f(x) + P(x), \quad (6)$$

where

$$P(x) = \sum_{j=1}^q \mu_j H_j(g_j(x)) \cdot g_j(x) + \sum_{j=q+1}^m \gamma_j H_j(h_j(x)) |h_j(x)|$$

$H_j(g_j(x))$ and $H_j(h_j(x))$ are denoted as

$$H_j(g_j(x)) = \begin{cases} 1, & \text{if } g_j(x) > 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$H_j(h_j(x)) = \begin{cases} 1, & \text{if } |h_j(x)| > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Liu et al. also modified the dynamic penalty factor (S-type function) as

$$\mu(g) = 10^{1+\epsilon \frac{\frac{\theta_2 - \theta_1}{20(-g + \frac{MCN}{4})} + \theta_1}{MCN}},$$

where g is the iteration number and μ is constrained to lie in $[10^{\theta_1}, 10^{\theta_2}]$ by penalty parameters θ_1 and θ_2 . Similar to the penalty technique mentioned above, the search space will diverge at the beginning of algorithm development for a smaller μ , increasing population diversity. A larger μ will improve the algorithm's convergence to the global optimum as generation increases. They have assumed $\theta_1 = 4$ and $\theta_2 = 6$ in this function.

5.2 Proposed dynamic penalty method

(d) It is generally known that the control parameters β , θ_1 , and θ_2 have been the primary basis for the construction of all the aforementioned penalty techniques. The penalized objective function is greatly impacted by the values of these parameters, which must be adjusted suitably based on the algorithm's initial testing. It shows trouble biasing the search towards the viable region if these parameter values are not adequate. Unlike the previously stated penalty function approach, we present a novel dynamic penalty method in this work that has a straightforward form and does not require any additional parameters or punishment factors. Therefore, in order to increase the optimization efficiency, the burden of fine-tuning the penalty factors/parameters has been avoided here. In order to penalize the infeasible solutions and favor feasible solutions, we include in our penalty formula the maximum iteration ($MaxIt$), the current generation number (it), and the number of constrained violations ($nconv(it, i)$). This allows us to quickly and easily guide the population to the feasible region. The information of the objective function and restrictions violation that is clubbed by the penalized function has been specified as follows:

$$F(x) = f(x) + \hat{P} \sum_{l=1}^{p+q} G_l(x), \quad (7)$$

where

$$\hat{P} = \begin{cases} 0, & \text{if } G_l(x) \leq 0, \\ 10^{\{e^{2 * (ncov(it,i) * \frac{(MaxIt - it)}{4 * (MaxIt)^4})}\}}, & \text{if } G_l(x) > 0. \end{cases}$$

Our suggested dynamic penalty differs significantly in another way: It gradually reduces as solutions go towards the feasible solution area, and it increases for nonfeasible solutions the further they are from the viable zone. At each algorithm iteration, we illustrate in Figure 3 the fluctuation in the arbitrary nonfeasible solution's penalty coefficient (\hat{P}) when the population approaches the feasible solution space.

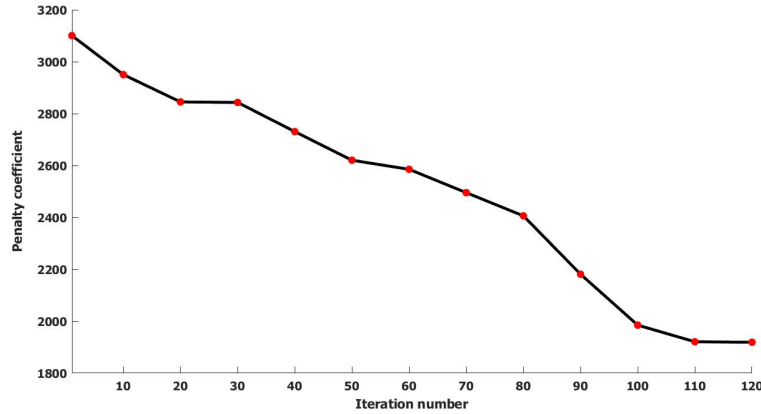


Figure 3: Variation in \hat{P} of arbitrary nonfeasible solution in terms of iteration number

6 Proposed hybrid SDDS-SABC algorithm

In order to create an improved algorithm that could effectively solve the COP with higher reliability, two techniques SDDS and SABC have been combined to create a novel hybrid optimization algorithm (SDDS-SABC). Through recursive decomposition, the SDDS approach reduces the size of the entire search region and concentrates computing effort on the subregion that has viable answers for additional decomposition. Comparatively, SABC is essential in identifying the most promising subregion by removing the best

solution to date from the subregion it is being implemented over. These procedures are carried out one after the other until the region shrinks to a nominal width. Below is a detailed discussion of these approaches' most notable feature.

6.1 Method of split-detect-discard-shrink (SDDS)

The basic motivation behind the SDDS is to initially split at the midpoint of the axis of the first variable of n -dimensional Euclidean space to partition the entire Euclidean space into two subregions of a particular shape. In the proposed SDDS approach, a series of stages involving SDDS is systematically undertaken to sequentially partition the Euclidean space where COP is defined into smaller and smaller subregions then solved recursively until no further division is conceivable. The following is how these steps are handled: (1) Using recursive splitting, we divided the search space D into two discrete subspaces, D_1 and D_2 , along the x_j , ($j = 1, 2, \dots, n$) axis, one variable at a time. (2) Determine which subspaces correspond to a better solution that represents a promising area in the search space by evaluating the function value at every feasible point in the two subspaces, D_1 and D_2 . (3) Discard any subspace without a promising solution, based on the matching solutions of D_1 and D_2 . (4) Reduce the initial search space D to the appropriate subspace of D_1 or D_2 , depending on which one contains the most promising solutions. Now either D_1 or D_2 becomes D . Until the search space D is limited to an area of nominal width containing the global optimal solution, all of these steps are repeatedly performed.

We perform the recursive splitting of D into D_1 and D_2 in the following manner. The first variable's range, $x_1 \in [\underline{x}_1, \bar{x}_1]$, should first be divided into two equal and disjoint sub-intervals: $[\underline{x}_1, m_1]$ and $[m_1, \bar{x}_1]$. Selected along the x_1 -axis, the point $m_1 = \frac{(\underline{x}_1 + \bar{x}_1)}{2}$ represents the first variable of n -dimensional Euclidean space. The promising region among D_1 and D_2 is replaced by D . Then, separating D into D_1 and D_2 is done by dividing the range of the second variable $x_2 \in [\underline{x}_2, \bar{x}_2]$ into two equal and disjoint subintervals $[\underline{x}_2, m_2]$ and $[m_2, \bar{x}_2]$ with a point $m_2 = (\underline{x}_2 + \bar{x}_2)/2$. This point is located along the

x_2 -axis at the center of the n -dimensional Euclidean space's second variable. Thus, dividing the whole Euclidean space into the two subregions, D_1 and D_2 , continues in this manner until the n th variable is reached. Actually, to continue the splitting process, each axis of variable x_j is taken in turn, starting with $j = 1$, going up to $j = n$. The expression for splitting D into two subregions of a certain form is as follows:

$$D_1 = \{x \in R^n : \underline{x}_i \leq x_i \leq m_i = (\frac{\underline{x}_i + \bar{x}_i}{2}), \underline{x}_j \leq x_j \leq \bar{x}_j, j = 1, 2, \dots, i-1, i+1, \dots, n\},$$

$$D_2 = \{x \in R^n : m_i = (\frac{\underline{x}_i + \bar{x}_i}{2}) \leq x_i \leq \bar{x}_i, \underline{x}_j \leq x_j \leq \bar{x}_j, j = 1, 2, \dots, i-1, i+1, \dots, n\}.$$

If D is not reduced to a region of nominal width after all the n -axis have been progressively separated, then we repeat the full sequential partition process.

The stages involved in SDDS have been demonstrated in Figure 4 below [56].

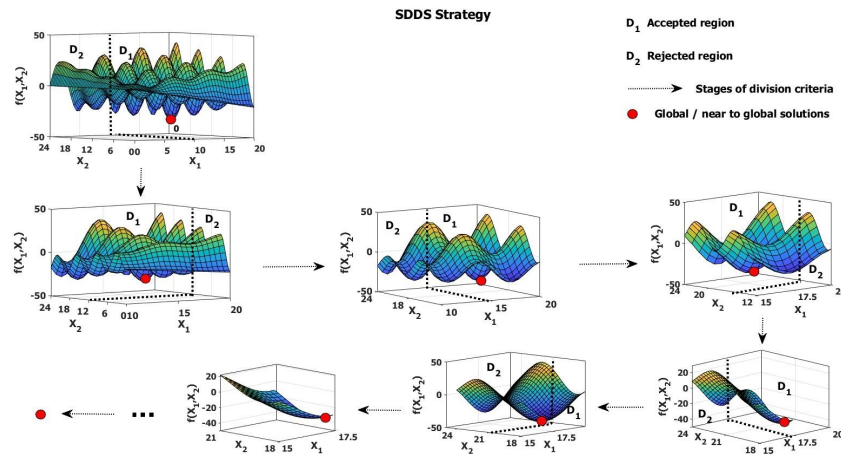


Figure 4: Displaying the SDDS strategy's steps

6.2 Using SABC to identify the promising subregion

View of both drawbacks and merits, we have been motivated to extend, modify, or hybridize ABC with variants of population-based algorithms or classical methods to boost its performance. The research did not just restrict the development of hybrid ABC algorithms to numerical COPs but also optimized a wide range of application-based problems. SABC phase has been applied to each subregion to detect and discard the subregion covering non-promising solution. The interval arithmetic rule has been used to indicate the subregions which can be discarded by comparing the two solutions, SABC produced in the subregions, and choosing the subregion holding a promising solution. This completes the first cycle of the hybrid algorithm. The method proceeds to explore the search for the promising region by repeatedly alternating the SDDS and SABC phases. Understanding if one subregion, represented by D_1 , covers a more promising solution(s) than the other subregion, D_2 , and vice versa, and then choosing the most promising subregion based on that understanding is a critical challenge in this black box optimization process. In order to accomplish this, we have included the SABC algorithm into the SDDS technique, which may be applied to optimization issues that are specified in both D_1 and D_2 . The subregion covering the nonpromising solution(s) was then discarded using a ranking and selection rule, and the subregion with the promising solution could be named the current shrink region D for further splitting. This process was done from the perspective of optimistic decision makers, who compared the best solutions obtained from both regions. The ranking and selection rule utilized for the minimization problem has been detailed below:

Let $F_1, Con_1 \in D_1$ and $F_2, Con_2 \in D_2$ be such that

$F_1 = f(x_1^*)$, $Con_1 = \sum_1^{p+q} G_l(x_1^*)$; x_1^* =best solution obtained by SABC in D_1 and

$F_2 = f(x_2^*)$, $Con_2 = \sum_1^{p+q} G_l(x_2^*)$; x_2^* =best solution obtained by SABC in D_2 .

1. If $Con_1 = Con_2 = 0$ & $F_1 < F_2$, then choose D_1 and discard D_2

2. If $Con_1 = Con_2 = 0$ & $F_2 < F_1$, then choose D_2 and discard D_1
3. If $Con_1 < Con_2$ & whether $F_1 < F_2$ or $F_1 > F_2$, then choose D_1 and discard D_2
4. If $Con_2 < Con_1$ & whether $F_1 < F_2$ or $F_1 > F_2$, then choose D_2 and discard D_1
5. If $Con_1 = 0$, $Con_2 = \gamma (\neq 0)$ & whether $F_1 < F_2$ or $F_1 > F_2$, then choose D_1 and discard D_2
6. If $Con_1 = \gamma (\neq 0)$, $Con_2 = 0$, & whether $F_1 < F_2$ or $F_1 > F_2$, then choose D_2 and discard D_1

6.3 Computational Complexity

Using the fundamental ABC method, we can evaluate the computational complexity of our proposed method. It should be noted that as it varies depending on the problem, we disregard the time required to compute the objective function here. We assume that T_{MaxIt} is the maximum number of iterations. For ABC, its computational complexity is $O(T_{MaxIt} \times SN)$. As for the proposed SDDS-SABC, the solutions need to be sorted at each iteration, so the computational complexity of SDDS-SABC is $O(T_{MaxIt} \times SN \times \log(SN))$. Although, the computational complexity of SDDS-SABC is higher than the basic ABC at the same maximum iteration, SDDS-SABC can achieve much better results than ABC. In addition, the time complexity of SDDS-SABC is similar to that of other ABC variants based on elite populations, but SDDS-SABC performs better than them.

7 Numerical results and discussion

Here, we demonstrate the validity of our proposed SDDS-SABC method through tests on well-known typical benchmark functions CEC 2006 [29] and some engineering design problems (EDPs) (see [19]). These test functions include diverse features like linear/nonlinear, low dimension/high di-

mension, continuous/discrete, separable/nonseparable, convex/nonconvex, unimodal/multi-modal varying feasible region (see Table 1). In this table,

Table 1: Test functions

| Benchmark functions | | | | | | | | |
|---------------------|----|------------------|-----------|----|----|----|----|----|
| Problem | n | Type of function | ρ | LI | NI | LE | NE | a |
| g01 | 13 | quadratic | 0.0111% | 9 | 0 | 0 | 0 | 6 |
| g02 | 20 | nonlinear | 99.99971% | 0 | 2 | 0 | 0 | 1 |
| g03 | 10 | polynomial | 0.0000% | 0 | 0 | 0 | 1 | 1 |
| g04 | 5 | quadratic | 52.1230% | 0 | 6 | 0 | 0 | 2 |
| g05 | 4 | cubic | 0.0000 % | 2 | 0 | 0 | 3 | 3 |
| g06 | 2 | cubic | 0.0066% | 0 | 2 | 0 | 0 | 2 |
| g07 | 10 | quadratic | 0.0003% | 3 | 5 | 0 | 0 | 6 |
| g08 | 2 | nonlinear | 0.8560 % | 0 | 2 | 0 | 0 | 0 |
| g09 | 7 | polynomial | 0.5121% | 0 | 4 | 0 | 0 | 2 |
| g10 | 8 | linear | 0.0010% | 3 | 3 | 0 | 0 | 6 |
| g11 | 2 | quadratic | 0.0000% | 0 | 0 | 0 | 1 | 1 |
| g12 | 3 | quadratic | 4.7713% | 0 | 1 | 0 | 0 | 0 |
| g13 | 5 | nonlinear | 0.0000% | 0 | 0 | 0 | 3 | 3 |
| g14 | 10 | nonlinear | 0.0000% | 0 | 0 | 3 | 0 | 3 |
| g15 | 3 | quadratic | 0.0000% | 0 | 0 | 1 | 1 | 2 |
| g16 | 5 | nonlinear | 0.0204% | 4 | 34 | 0 | 0 | 4 |
| g17 | 6 | nonlinear | 0.0000% | 0 | 0 | 0 | 4 | 4 |
| g18 | 9 | quadratic | 0.0000% | 0 | 13 | 0 | 0 | 6 |
| g19 | 15 | nonlinear | 33.4761% | 0 | 5 | 0 | 0 | 0 |
| g20 | 24 | linear | 0.0000% | 0 | 6 | 2 | 12 | 16 |
| g21 | 7 | linear | 0.0000% | 0 | 1 | 0 | 5 | 6 |
| g22 | 22 | linear | 0.0000% | 0 | 1 | 8 | 11 | 19 |
| g23 | 9 | linear | 0.0000% | 0 | 2 | 3 | 1 | 6 |
| g24 | 2 | linear | 79.6556 % | 0 | 2 | 0 | 0 | 2 |

“ n ” is the dimension of the problem, $\rho = |F|/|S|$ represents the proportion between the feasible region and the search space. Also, LI, NI, LE, NE and “ a ” represent the numbers of linear inequality constraints, nonlinear inequality constraints, linear equality constraints, nonlinear equality constraints, and active constraints, respectively. We study the robustness of our proposed dynamic penalty-based constraint handling techniques integrated into the SDDS-SABC method on selected problems of CEC 2006 (see Table 2). Furthermore, we compare these results with the present dynamic penalty methods. The overall best-found results have been displayed in Table 2.

Table 2: Comparative analysis between the proposed dynamic penalty and existing penalty methods

| Problems | (a) | (b) | (c) | (d) |
|----------|--------------|--------------|--------------|---------------|
| g01 | -15.9472628 | -15.8610643 | -15.6103179 | -15.4618784 |
| g05 | 5126.8837241 | 5125.046215 | 5124.1968423 | 5124.0049727 |
| g15 | 952.8593772 | 951.9805434 | 951.6104782 | 951.5300084 |
| g24 | -6.239074165 | -6.403721373 | -6.53271104 | -6.632598318 |
| | (a) [35]; | (b): [34]; | (c): [33] ; | (d): Proposed |

The results show that our proposed penalty method explores feasible solution space efficiently to provide promising results. We have coded the said optimization method in MATLAB and executed it in an HP Pavilion Laptop with Intel (R) 11th Gen Core i5-512GB SSD @ 2.40 GHz. The basic parameters used in the SDDS-SABC method include; colony size $SN = 100$ (equal to the number of employed and onlooker bees), the Scout bee's food source abandonment parameter $= \text{round}(SN \times n \times p_{val})$, where p_{val} is a small probability value in the range (0.05 - 0.08). The maximum cycle number (MCN) is 50, which serves as the termination criterion. The values of the control parameters of the SABC algorithm used in our simulation studies and the values assumed by the authors in their respective state-of-the-art algorithms, which we have used for comparison purposes, have been displayed in Table 3. Firstly, we study the robustness of SDDS-SABC method implemented over our proposed dynamic penalty-based constraint handling techniques through 24 benchmark functions, and secondly, on the engineering design problems comparing results obtained using different penalty methods. Using the following indices—exactness, consistency, efficacy, and statistical analysis—we assess resilience in several ways. The following definitions apply to these: a) Accuracy: the degree of the best-found solution's quality and its separation from the global solution. In a similar vein, the degree to which the worst-found solution deviates from the global answer and its quality, tested on 25 independent runs of the best and worst identified solutions.

(b) Consistency: comprehend the resilience and stability of the optimization technique on the problem that leads to the best possible solution. Test the following: the average and standard deviation of the solutions from the 25

Table 3: **Control parameter values of different algorithms**

| (a) Based on ABC | | | |
|---------------------|--------------|-------|--|
| Algorithms | Popsize | MaxIt | Limit |
| I-ABC | 20 | 6000 | $SN \times n$ |
| CB-ABC | 90 | 500 | $SN \times n$ |
| IABC-MAL | 30 | 500 | $0.5 \times SN \times n$ |
| MG-ABC | 50 | 1000 | $SN \times n$ |
| SDDS-SABC | 100 | 50 | $\text{round}(p_{val} \times n \times SN)$ |
| (b) Based on nonABC | | | |
| Algorithms | Popsize | MaxIt | Limit |
| UFA | 50 | 40 | 30 |
| GLF-GWO | $3 \times n$ | 3000 | $10^4 \times n$ |
| GGA | 50 | 50 | 25 |
| JaQA | 50 | 50 | 25 |
| SDDS-SABC | 100 | 50 | $\text{round}(p_{val} \times n \times SN)$ |

runs.

c) Statistical analysis: compare the significant difference in the performance of our proposed SDDS-SABC method with other existing algorithms. Test on algorithms through the Friedman test and Wilcoxon signed ranks test, which are standard nonparametric statistical tests.

7.1 Sensitivity analysis of some key parameters in SABC algorithm

In addition to improving algorithmic efficiency, appropriate input values for the algorithm's parameters are crucial for its robustness, stability, and best-found objective value. Sensitivity analysis has therefore been performed to examine the impact of the input values of the important SABC parameters, such as MaxIt and SN, on the performance of our algorithm SDDS-SABC towards achieving the optimal solution and its stability in each algorithm run. We have set MaxIt to 50 and the algorithm's iteration numbers range from 1 to MaxIt. Also, we have taken the population size SN is 100. We display this study graphically on a specific benchmark test function, g07. The evolution

of the best-found solutions at various iterations when $\text{MaxIt}=50$ is shown in Figure 5(i). Additionally, it demonstrates that when $\text{SN}=100$, the algorithm converges to the best-found value of 24.34849. The algorithm converges to the best-found value of 24.34849 at $\text{MaxIt}=50$ for $\text{SN}=100$, as seen in Figure 5(ii). Once more, the stable solution 24.34849 is reached with values higher than 50. For other test functions, we see a comparable effect. We cannot obtain the best or nearly best solution to the problems if we set the value of MaxIt and SN to be less than 50 and 100, respectively.

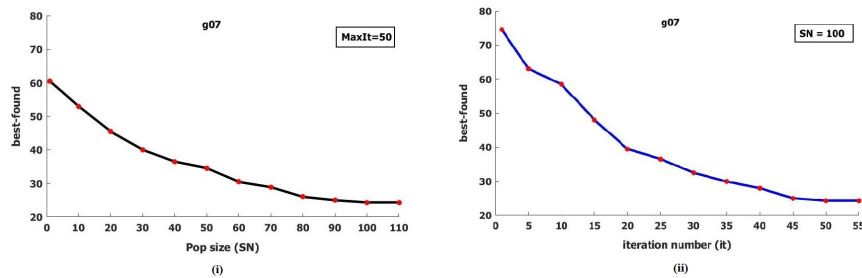


Figure 5: Evolution of best-found value with respect to iteration number and colony size

7.2 Study on algorithm performance based on exactness, consistency and effectiveness

We present in Table 4, the best-found, worst-found, mean and standard deviation (std) of the result obtained from each benchmark function in 25 independent runs of the SABC-SDDS algorithm. The results found are encouraging.

We also compare our results with other state-of-the-art hybrid algorithms viz., I-ABC, CB-ABC, IABC-MAL and MG-ABC (see Table 5) in terms of best-found, worst-found and mean values. These said hybrid algorithms had been developed by combining other heuristics or traditional methods with ABC. Table 5 shows that our SDDS-SABC algorithm outperforms other algorithms in some problems and works equally well for other problems. However,

Table 4: Statistical results of SDDS-SABC on benchmark functions, averaged over 25 independent runs

| Functions | optimal value | best-found | worst-found | mean | std |
|-----------|-------------------|--------------------|-------------------|-------------------|------------------|
| g01 | -15.0000000000 | -15.4618784341837 | -14.7361050072335 | -15.071895226323 | 5.99789658392 |
| g02 | -0.8036191042 | -0.875454224809 | -0.77323254632 | -0.741139419121 | 0.31517588432024 |
| g03 | -1.0005001000 | -1.098598548944 | -0.964899232223 | -1.01285024552 | 0.041820449166 |
| g04 | -30665.5386717834 | -30665.8337336955 | -30698.7685959581 | -30653.100331767 | 0.2687651065913 |
| g05 | 5126.4967140071 | 5124.0049727086 | 5176.27733899147 | 5134.7200444372 | 0.4966102760171 |
| g06 | -6961.8138755802 | -6983.37054536156 | -6698.63125675940 | -6946.0112672204 | 0.59174789701531 |
| g07 | 24.302090681 | 24.348490589297 | 27.8909837113848 | 25.70325545062 | 1.051962320689 |
| g08 | -0.0958250415 | -0.095722476579 | -0.0813461879519 | -0.0907741721133 | 0.0038036427493 |
| g09 | 680.6300573745 | 684.560836511063 | 695.96431708531 | 689.024312257864 | 2.0631346433681 |
| g10 | 7049.2480205286 | 7047.3431791133 | 7124.63089605634 | 7066.0444725057 | 19.190512293263 |
| g11 | 0.749900000 | 0.7505189211405 | 0.76423525277597 | 0.7562800118598 | 0.00344183933935 |
| g12 | -1.000000000 | -0.999954717347 | -0.9398378640313 | -0.990291881794 | 0.01317550762524 |
| g13 | 0.0539415140 | 0.0505131549885 | 0.0560012746738 | 0.0549429543562 | 0.00109084362246 |
| g14 | -47.7648884595 | - | - | - | - |
| g15 | 961.7150222899 | 951.530008446544 | 960.601193735686 | 955.3401698407 | 2.3820716382008 |
| g16 | -1.9051552586 | - | - | - | - |
| g17 | 8853.5396748064 | - | - | - | - |
| g18 | -0.8660254038 | -0.942906584997 | -0.87048513538499 | -0.90112279347847 | 0.0179637098296 |
| g19 | 32.6555929502 | -22410.82795313840 | 6013.94217031911 | -14859.743320706 | 4321.0628338304 |
| g20 | 0.2049794002 | - | - | - | - |
| g21 | 193.7245100700 | 193.508264477410 | 194.783450798573 | 193.62084757633 | 0.0148378378312 |
| g22 | 236.4309755040 | - | - | - | - |
| g23 | -400.0551000000 | -4156.949391713 | -4087.08875546532 | -4109.4664146789 | 11.035324578671 |
| g24 | -5.5080132716 | -6.6325983187262 | -5.34276622737366 | -5.921377563442 | 0.38492496179489 |

one difficulty noted in our algorithm is that it could not reach a reasonably good solution for problems g14, g16, g17, g20 and g22. Likewise, I-ABC, CB-ABC, IABC-MAL too were unable to locate the optimal solution for the problems g20 and g22. Also, MG-ABC could not solve problems g21, g22, g23 and g24, where our SDDS-SABC algorithm worked well except for g22. For g21, SDDS-SABC produced a better best-found value than the I-ABC and CB-ABC and equally well for IABC-MAL. In terms of mean, SDDS-SABC provides a much better result than the I-ABC. We also noted that no hybrid algorithms with ABC could find the reasonably good optimal/near to optimal solution for the problems g14, g17, g20, and g22 because the ratio between their feasible region and search space are minimal ($\rho=0.0000\%$),. A similar situation arises for problem g16 occupying $\rho=0.0204\%$, which is very small in percentage value (see Table 1). So, finding the global or near-to-global solution for those problems is challenging. In addition, problems g14, g16, and g17 have highly nonlinear type of functions. Although g20 and g22 are linear, due to their high dimension, that is, $n = 24$ and $n = 22$, respectively, their solutions could not be traced due to their small feasible space. Our SDDS-SABC algorithm could find better results for the nonlinear type of function (g02, g08 and g19) when the proportion between the feasible region

and search space is either large, small or midway (99.99971%, 0.8560% and 33.4761%) even for high dimensions problems ($n=20$, $n=2$ and $n=15$). An exciting observation noted through problem g13 is that for the low dimension problem ($n = 5$), even though it is highly nonlinear, we could find a better result from a tiny space of 0.0000% originating between the feasible region and search space. So, performance of SDDS-SABC depends on the nature of the function, the problem's dimension, and the percentage between the feasible region and search space.

We have also compared our SDDS-SABC results with the other popular non-ABC based hybrid algorithms like; UFA, GLF-GWO, GGA, and JaQa [11]. We can see from Table 6 that our proposed SDDS-SABC performs better than UFA, GLF-GWO, GGA, and JaQa algorithms in terms of best-found value for problem g03. SDDS-SABC performs better than GGA in terms of best-found value for problem g04. However, GLF-GWO could not find any feasible solution for g05, whereas our SDDS-SABC could provide a better result. For the problem g06, our proposed SDDS-SABC algorithm gives better solution than UFA, GLF-GWO, GGA, and JaQa algorithms in terms of best-found value. Similarly, for problem g15, SDDS-SABC works better than UFA, GLF-GWO, GGA, and JaQa in terms of best-found, worst-found, and mean values. In g21, SDDS-SABC outperforms UFA for best-found value and gives better value than GGA and JaQa in term of mean value. Also, GLF-GWO could not provide any feasible solution for the problem g21. In g24, we can see that our SDDS-SABC and GGA give approximately equal best-found values. Moreover, UFA and GLF-GWO could not find optimal solutions to problems g20 and g22. Compared with the rest of the algorithms, only GGA could solve problem g20 and find a better result. On the other hand, GGA could not work much well for problems g14 and g22. Finally, except for test problems g14, g16, g17, g20, and g22, our SDDS-SABC algorithm provides significantly better results than other algorithms.

Table 5: The comparison of best-found, worst-found and mean results with ABC based algorithms

| Functions | | I-ABC (2015) | CB-ABC (2015) | IABC-MAL (2017) | MG-ABC (2018) | SDDS-SABC |
|-----------|-------------|-----------------|------------------|--------------------|------------------|-------------------|
| g01 | best-found | -15.000 | -15.000 | -15.000000 | -15.000000 | -15.4618784341837 |
| | worst-found | -15.000 | -15.000 | -15.000000 | -9.000000 | -14.7361050072335 |
| | mean | -15.000 | -15.000 | -15.000000 | -13.553540 | -15.071895226323 |
| g02 | best-found | -0.803619 | -0.803619 | -0.803619 | -0.8036108 | -0.875454224809 |
| | worst-found | -0.778278 | -0.777844 | -0.785568 | -0.7604863 | -0.77323254632 |
| | mean | -0.800094 | -0.794522 | -0.799460 | -0.7890629 | -0.741139419121 |
| g03 | best-found | -1.000 | -1.0005 | -1.000500 | -1.000400 | -1.098598548944 |
| | worst-found | -0.999 | -1.0005 | -1.000500 | -1.000258 | -0.964899232223 |
| | mean | -1.0004 | -1.0005 | -1.000500 | -1.000383 | -1.01285024552 |
| g04 | best-found | -30665.539 | -30665.539 | -30665.539 | -30665.540 | -30665.8337336955 |
| | worst-found | -30665.539 | -30665.539 | -30665.539 | -30665.540 | -30698.7685959581 |
| | mean | -30665.539 | -30665.539 | -30665.539 | -30665.540 | -30653.100331767 |
| g05 | best-found | 5126.498 | 5126.197 | 5126.498 | 5126.4970 | 5124.0049727086 |
| | worst-found | 5126.944 | 5126.497 | 5126.498 | 6112.169 | 5176.27733899147 |
| | mean | 5131.861 | 5126.497 | 5126.498 | 5467.7560 | 5134.7200444372 |
| g06 | best-found | -6961.814 | -6961.814 | -6961.814 | -6961.8030 | -6983.37054536156 |
| | worst-found | -6961.814 | -6961.814 | -6961.814 | -6957.1230 | -6698.63125675940 |
| | mean | -6961.814 | -6961.814 | -6961.814 | -6959.4890 | -6946.0112672204 |
| g07 | best-found | 24.311 | 24.3062 | 24.3064 | 24.326530 | 24.348490589297 |
| | worst-found | 24.677 | 24.3062 | 24.3062 | 25.099270 | 27.8909837113848 |
| | mean | 24.366 | 24.3062 | 24.3062 | 24.780640 | 25.70325545062 |
| g08 | best-found | -0.095825 | -0.095825 | -0.095825 | -0.095825 | -0.095722476579 |
| | worst-found | -0.095825 | -0.095825 | -0.095825 | -0.095825 | -0.0813461879519 |
| | mean | -0.095825 | -0.095825 | -0.095825 | -0.095825 | -0.0907741721133 |
| g09 | best-found | 680.631 | 680.630 | 680.630 | 680.6302 | 684.560836511063 |
| | worst-found | 680.637 | 680.630 | 680.630 | 680.6322 | 695.96431708531 |
| | mean | 680.633 | 680.630 | 680.630 | 680.6309 | 689.024312257864 |
| g10 | best-found | 7049.321 | 7049.248 | 7049.248 | 7104.006 | 7047.3431791133 |
| | worst-found | 7049.343 | 7049.248 | 7049.248 | 7504.944 | 7124.63089605634 |
| | mean | 7124.042 | 7049.248 | 7049.248 | 7357.461 | 7066.0444725057 |
| g11 | best-found | 0.7499 | 0.7499 | 0.749900 | 0.749995 | 0.7505189211405 |
| | worst-found | 0.7499 | 0.7499 | 0.749900 | 0.750127 | 0.76423525277597 |
| | mean | 0.7499 | 0.7499 | 0.749900 | 0.750025 | 0.7562800118598 |
| g12 | best-found | -1.000 | -1.000 | -1.000000 | -1.000000 | -0.999954717347 |
| | worst-found | -1.000 | -1.000 | -1.000000 | -1.000000 | -0.9398378640313 |
| | mean | -1.000 | -1.000 | -1.000000 | -1.000000 | -0.990291881794 |
| g13 | best-found | 0.053958 | 0.053942 | 0.0539498 | 0.05394861 | 0.0505131549885 |
| | worst-found | 0.055130 | 0.43880 | 0.0539498 | 0.4377867 | 0.0560012746738 |
| | mean | 0.054144 | 0.066770 | 0.0539498 | 0.171074 | 0.0549429543562 |

Table 5: (continued)

| | | | | | | |
|-----|-------------|-----------|-------------|-------------|------------|--------------------|
| g14 | best-found | -47.665 | -47.765 | -47.765 | -47.675860 | - |
| | worst-found | -47.830 | -47.765 | -47.765 | -46.465260 | - |
| | mean | -47.201 | -47.765 | -47.765 | -47.246220 | - |
| g15 | best-found | 961.715 | 961.715 | 961.715 | 961.715100 | 951.530008446544 |
| | worst-found | 961.720 | 961.715 | 961.715 | 965.208600 | 960.601193735686 |
| | mean | 961.716 | 961.715 | 961.715 | 962.173700 | 955.34016984070 |
| g16 | best-found | -1.905 | -1.905 | -1.905 | -1.905155 | - |
| | worst-found | -1.905 | -1.905 | -1.905 | -1.905155 | - |
| | mean | -1.905 | -1.905 | -1.905 | -1.905155 | - |
| g17 | best-found | 8860.864 | 8853.533875 | 8853.533875 | 8853.53 | - |
| | worst-found | 8983.359 | 8941.940741 | 8927.597785 | 9241.820 | - |
| | mean | 8909.994 | 8902.869928 | 8883.163028 | 8915.998 | - |
| g18 | best-found | -0.866025 | -0.672216 | -0.866025 | -0.8660253 | -0.94290658499 |
| | worst-found | -0.856622 | -0.866025 | -0.866025 | -0.8648695 | -0.87048513538499 |
| | mean | -0.865310 | -0.866025 | -0.866025 | -0.8657735 | -0.90112279347847 |
| g19 | best-found | 32.784 | 35.746 | 32.6556 | -5.508013 | -22410.82795313840 |
| | worst-found | 34.856 | 32.6557 | 32.6556 | -5.508013 | 6013.94217031911 |
| | mean | 33.344 | 32.6556 | 32.6556 | -5.508013 | -14859.743320706 |
| g20 | best-found | - | - | - | 1.393571 | - |
| | worst-found | - | - | - | 1.399163 | - |
| | mean | - | - | - | 1.394359 | - |
| g21 | best-found | 193.725 | 257.156 | 193.725 | - | 193.725 |
| | worst-found | 964.030 | 193.725 | 193.725 | - | 194.783450798573 |
| | mean | 622.678 | 193.725 | 193.725 | - | 193.62084757633 |
| g22 | best-found | - | - | - | - | - |
| | worst-found | - | - | - | - | - |
| | mean | - | - | - | - | - |
| g23 | best-found | -358.183 | -400.055 | -400.055 | - | -4156.94939171 |
| | worst-found | 899.881 | -400.055 | -400.055 | - | -4087.08875546532 |
| | mean | 169.021 | -400.055 | -400.055 | - | -4109.4664146789 |
| g24 | best-found | -5.508 | -5.508 | -5.508 | - | -6.6325983187262 |
| | worst-found | -5.508 | -5.508 | -5.508 | - | -5.34276622737366 |
| | mean | -5.508 | -5.508 | -5.508 | - | -5.921377563442 |

Table 6: The comparison of best-found, worst-found and mean results with different nonABC algorithms

| Functions | | UFA (2019) | GLF-GWO (2020) | GGA (2021) | JaQA (2022) | SDDS-SABC |
|-----------|-------------|---------------|-------------------|-------------------|----------------|-------------------|
| g01 | best-found | -15.000000 | 15.00000 | -15.0000000000 | -15.00000 | -15.4618784341837 |
| | worst-found | -15.000000 | -14.9999 | -15.0000000000 | -15.00000 | -14.7361050072335 |
| | mean | -15.000000 | -15.000000 | -15.0000000000 | -15.00000 | -15.071895226323 |
| g02 | best-found | -0.8033 | -0.803619 | -0.8030191042 | -0.803605 | -0.875454224809 |
| | worst-found | -0.5205742 | -0.6275 | -0.8010191042 | -0.800272 | -0.77323254632 |
| | mean | -0.7458475 | -0.7249 | -0.8020191042 | -0.80111 | -0.741139419121 |
| g03 | best-found | -1.0005 | -1.0005 | -1.0004181146 | -1.0005 | -1.098598548944 |
| | worst-found | -1.0005 | -0.0006 | -0.9993067321 | -0.9987 | -0.964899232223 |
| | mean | -1.0005 | -0.7386 | -1.0000114315 | -1.00031 | -1.01285024552 |
| g04 | best-found | -30665.5203 | -30665.539 | -30678.4386717834 | -30665.5387 | -30665.8337336955 |
| | worst-found | -30665.539 | -30665.0825 | -30667.0386717834 | -30665.5387 | -30698.7685959581 |
| | mean | -30665.539 | -30665.3389 | -30667.6386717834 | -30665.5387 | -30653.100331767 |
| g05 | best-found | 5126.49671 | - | 5126.4967135571 | 5126.484 | 5124.0049727086 |
| | worst-found | 5126.49671 | - | 5126.4967135601 | 5126.611 | 5176.27733899147 |
| | mean | 5126.49671 | - | 5126.4967135581 | 5126.504 | 5134.7200444372 |
| g06 | best-found | -6961.83884 | -6961.4784 | -6961.8130705802 | -6961.814 | -6983.37054536156 |
| | worst-found | -6961.81388 | -6961.4886 | -6961.8130665802 | -6961.814 | -6698.63125675940 |
| | mean | -6961.81388 | -6961.7341 | -6961.8130685802 | -6961.814 | -6946.0112672204 |
| g07 | best-found | 24.306209 | 24.3851 | 24.3934806327 | 24.0012 | 24.348490589297 |
| | worst-found | 24.306209 | 25.6385 | 27.7034023081 | 24.6781 | 27.8909837113848 |
| | mean | 24.306209 | 24.7221 | 26.5452127381 | 24.2121 | 25.70325545062 |
| g08 | best-found | -0.09582504 | -0.0958 | -0.0958233590 | -0.095825 | -0.095722476579 |
| | worst-found | -0.09582504 | -0.0958 | -0.0951989658 | -0.095825 | -0.0813461879519 |
| | mean | -0.09582504 | -0.0958 | -0.0955852752 | -0.095825 | -0.0907741721133 |
| g09 | best-found | 680.630057 | 680.6538 | 680.6301199745 | 680.631 | 684.560836511063 |
| | worst-found | 680.630057 | 680.3862 | 680.6313973745 | 680.631 | 695.96431708531 |
| | mean | 680.630057 | 681.0990 | 680.6306403745 | 680.631 | 689.024312257864 |
| g10 | best-found | 7049.24802 | 7729.9603 | 7049.2479999286 | 7006.52 | 7047.3431791133 |
| | worst-found | 7049.24802 | 8554.3989 | 7049.2480002286 | 7121.83 | 7124.63089605634 |
| | mean | 7049.24802 | 8276.2365 | 7049.2480000286 | 7086.136 | 7066.0444725057 |
| g11 | best-found | 0.7499 | 0.7499 | 0.7493788256 | 0.7499 | 0.7505189211405 |
| | worst-found | 0.7499 | 0.9998 | 0.7498827597 | 0.7499 | 0.76423525277597 |
| | mean | 0.7499 | 0.7669 | 0.7496174447 | 0.7499 | 0.7562800118598 |
| g12 | best-found | -1.000000 | -1.000 | -1.0000000000 | -1.00 | -0.999954717347 |
| | worst-found | -1.000000 | -1.000 | -1.0000000000 | -1.00 | -0.9398378640313 |
| | mean | -1.000000 | -1.000 | -1.0000000000 | -1.00 | -0.990291881794 |
| g13 | best-found | 0.0539415 | 0.9527 | 0.0539181140 | 0.00174 | 0.0505131549885 |
| | worst-found | 0.0539415 | 2.3466 | 0.0539417680 | 0.00174 | 0.0560012746738 |
| | mean | 0.0539415 | 1.1903 | 0.0539299140 | 0.00174 | 0.0549429543562 |

Table 6: (continued)

| | | | | | | |
|-----|-------------|-------------|----------|-------------------|-----------|--------------------|
| g14 | best-found | -47.764879 | -46.7926 | 1172.2351115405 | -48.0111 | - |
| | worst-found | -47.764879 | -38.1941 | 1312.2351115405 | -46.1022 | - |
| | mean | -47.764879 | -41.8264 | 1252.2351115405 | -46.8843 | - |
| g15 | best-found | 961.7150223 | 961.7157 | 952.4957146499 | 961.6758 | 951.530008446544 |
| | worst-found | 961.7150223 | 971.8903 | 960.1071304699 | 961.6758 | 960.601193735686 |
| | mean | 961.7150223 | 965.7727 | 955.0510816799 | 961.6758 | 955.34016984070 |
| g16 | best-found | -1.90515526 | -1.9045 | -1.9051549586 | -1.9052 | - |
| | worst-found | -1.90515526 | -1.6776 | -1.9051543336 | -1.9052 | - |
| | mean | -1.90515526 | -1.8346 | -1.9051546316 | -1.9052 | - |
| g17 | best-found | 8853.533875 | - | 8892.5396953064 | 8853.5396 | - |
| | worst-found | 8853.533875 | - | 8962.5396748064 | 8902.223 | - |
| | mean | 8853.533875 | - | 8918.3396748064 | 8872.5142 | - |
| g18 | best-found | -0.8660254 | -0.8660 | -0.8619563027 | -0.86603 | -0.94290658499 |
| | worst-found | -0.8660254 | -0.6569 | -0.8461369131 | -0.86601 | -0.87048513538499 |
| | mean | -0.8660254 | -0.8233 | -0.8541324790 | -0.86602 | -0.90112279347847 |
| g19 | best-found | 32.655593 | 32.2874 | -66409.2048070498 | -32.6699 | -22410.82795313840 |
| | worst-found | 32.655593 | 82.7696 | -364.8315650498 | -32.7872 | 6013.94217031911 |
| | mean | 32.655593 | 43.0767 | 30.3617957802 | -32.6551 | -14859.743320706 |
| g20 | best-found | - | - | 0.3929794002 | 0.24072 | - |
| | worst-found | - | - | 0.5209794002 | 0.24794 | - |
| | mean | - | - | 0.4389794002 | 0.24381 | - |
| g21 | best-found | 193.724520 | - | 193.7245100700 | 193.4011 | 193.7245100700 |
| | worst-found | 520.165650 | - | 193.7245100700 | 203.9120 | 194.783450798573 |
| | mean | 255.559033 | - | 193.7245100700 | 193.7302 | 193.62084757633 |
| g22 | best-found | - | - | - | 5.08E+02 | - |
| | worst-found | - | - | - | 3.03E+07 | - |
| | mean | - | - | - | 2.14E+03 | - |
| g23 | best-found | -400.0551 | -0.0651 | -397.7451000000 | -412.520 | -4156.94939171 |
| | worst-found | -400.0551 | 809.3461 | -392.4451000000 | -388.2426 | -4087.08875546532 |
| | mean | -400.0551 | 269.7458 | -395.8651000000 | -399.3486 | -4109.4664146789 |
| g24 | best-found | -5.50801327 | -5.5080 | -6.7054079016 | -5.5094 | -6.6325983187262 |
| | worst-found | -5.50801327 | -3.0000 | -6.0978333186 | -5.5094 | -5.34276622737366 |
| | mean | -5.50801327 | -5.2834 | -6.3582253626 | -5.5094 | -5.921377563442 |

7.3 Study on algorithm performance using statistical analysis

On benchmark functions, nonparametric statistical tests on the best-found values have been carried out in order to rank the performance of the suggested and current algorithms. To determine whether there is a difference between the estimated outcomes produced by different algorithms, the nonparametric Friedmans test is employed. Furthermore, at a significance level of 5%, Wilcoxon's signed rank test has been applied independently to two groups of algorithms as a nonparametric test. It is expected that all algorithms function similarly under the null hypothesis. Reject the null hypothesis if the provided p-value is less than 0.05. The null hypothesis is rejected, indicating that all of the algorithms under investigation perform significantly

differently. From Table 7 of the Friedmans mean rank test, we see that the mean rank of SDDS-SABC attains the lowest value 2.26 and hence ranked 1. It means that our SDDS-SABC is better than the other hybrid ABC-based algorithms CBABC with rank 2, IABC-MAL with rank 3, SACABC with rank 4 and the last rank is 5 of IABC.

In Wilcoxon signed rank test results (see Table 8) comparing pairwise SDDS-SABC with IABC, CB-ABC, IABC-MAL and MG-ABC respectively, the results show that in every pairwise comparison, the sum of SDDS-SABC's positive ranks is significantly greater than the sum of its negative ranks. Furthermore, their p-value is less than 0.05, as can be shown. This suggests that SDDS-SABC performs better than other available methods.

Table 7: On ABC based hybrid algorithms, the mean ranking attained by Friedman's mean rank test at a significance level of 5%.

| Algorithms | Mean rank | Rank |
|------------------------------|-------------|----------|
| IABC(Liang et al., 2015) | 3.66 | 5 |
| CBABC (Brajevic, 2015) | 2.75 | 2 |
| IABC-MAL (Long et al., 2017) | 3.00 | 3 |
| MGABC (Bansal, 2018) | 3.34 | 4 |
| SDDS-SABC | 2.25 | 1 |

Table 8: Results of the Wilcoxon signed rank test on hybrid algorithms that are ABC based, with a significance threshold of 5%.

| Comparison | Observations | No. of test functions | Sum of positive rank | Sum of negative rank | p-value |
|-------------------------|--------------------|-----------------------|----------------------|----------------------|---------|
| SDDS-SABC with I-ABC | SDDS-SABC<I-ABC | 14 | 165.00 | 25.00 | 0.005 |
| | SDDS-SABC>I-ABC | 5 | | | |
| SDDS-SABC with CB-ABC | SDDS-SABC<CB-ABC | 14 | 164.00 | 26.00 | 0.005 |
| | SDDS-SABC>CB-ABC | 5 | | | |
| SDDS-SABC with IABC-MAL | SDDS-SABC<IAB-CMAL | 14 | 164.00 | 26.00 | 0.005 |
| | SDDS-SABC>IABC-MAL | 5 | | | |
| SDDS-SABC with MG-ABC | SDDS-SABC<MG-ABC | 11 | 113.00 | 23.00 | 0.020 |
| | SDDS-SABC>MG-ABC | 5 | | | |

Again, comparing the result of SDDS-SABC with the different types of nonABC hybrid algorithms (see Table 9) using Friedman's mean rank test, we can see that the mean rank of SDDS-SABC is lowest with a value of 1.88 and therefore ranked as 1. The other lowest values are 2.37, 2.41, 2.53, and 3.18 for JaQA, GGA, UFA, and GLF-GFO, respectively, giving ranks 2, 3, 4, and 5 per their performance. The first rank of SDDS-SABC indicates that this works better than the other algorithms. For the same hybrid algo-

rithms, the Wilcoxon signed rank test results have been displayed in Table 10. The pairwise comparison of SDDS-SABC with UFA, GLF-GWO, GGA, and JaQa, respectively, demonstrates that for every pairwise comparison, the total of the positive rank of SDDS-SABC is significantly larger than the negative rank. Furthermore, the p-values of SDDS-SABC are less than 0.05, indicating that it performs better than the other algorithms in the comparison. IBM SPSS statistics has been used to conduct these Friedman and Wilcoxon tests.

Table 9: On nonABC based hybrid algorithms, the mean ranking attained by Friedman's mean rank test at a significance level of 5%.

| Algorithms | Mean rank | Rank |
|------------------------------|-------------|----------|
| UFA (Brajevic et al., 2019) | 2.53 | 4 |
| GLF-GFO (Gupta et al., 2020) | 3.18 | 5 |
| GGA (D'Angelo et al., 2021) | 2.41 | 3 |
| JaQA (Das et al., 2022) | 2.37 | 2 |
| SDDS-SABC | 1.88 | 1 |

Table 10: Results of the Wilcoxon signed rank test on hybrid algorithms that are non-ABC based, with a significance threshold of 5%.

| Comparison | Observations | No. of test functions | Sum of positive rank | Sum of negative rank | p-value |
|------------------------|-------------------|-----------------------|----------------------|----------------------|---------|
| SDDS-SABC with UFA | SDDS-SABC<UFA | 14 | 165.00 | 25.00 | 0.005 |
| | SDDS-SABC>UFA | 5 | | | |
| SDDS-SABC with GLF-GWO | SDDS-SABC<GLF-GWO | 13 | 136.00 | 17.00 | 0.005 |
| | SDDS-SABC>GLF-GWO | 4 | | | |
| SDDS-SABC with GGA | SDDS-SABC<GGA | 13 | 144.00 | 46.00 | 0.049 |
| | SDDSSABC>GGA | 6 | | | |
| SDDS-SABC with JaQA | SDDS-SABC<JaQA | 13 | 146.00 | 47.00 | 0.051 |
| | SDDSSABC>JaQA | 6 | | | |

8 Applications of algorithm on real-life engineering design problems:

In this section, we present the challenging five real-life engineering design problems; see Table 11. These problems have been solved by our proposed SDDS-SABC algorithm and further compared with state-of-art hybrid algorithms (see Table 12).

Table 11: Engineering design problems

| Sr. no. | Problem | n | LI | NI | LE | NE |
|---------|---------------------------|---|----|----|----|----|
| EDP1 | Three-bar truss design | 2 | 0 | 3 | 0 | 0 |
| EDP2 | Compression spring design | 3 | 0 | 4 | 0 | 0 |
| EDP3 | Cantilever beam design | 5 | 0 | 1 | 0 | 0 |
| EDP4 | Pressure vessel design | 4 | 2 | 2 | 0 | 0 |
| EDP5 | Heat exchanger design | 8 | 3 | 3 | 0 | 0 |

8.1 Three-bar truss design problem

A three-bar planar truss structure (see [19]) has been taken into account in this case study. Initially, Nowacki developed this problem to reduce the volume of a statically loaded three-bar truss. Each truss element's stress is subject to limitations. The problem has been defined mathematically as follows:

$$\begin{aligned}
 \text{Min} \quad & f_1(x_1, x_2) = L * (2\sqrt{(2x_1)} + x_2), \\
 \text{s.t.} \quad & g_1(x_1, x_2) = \frac{\sqrt{(2x_1)} + x_2}{\sqrt{(2x_1^2)} + 2x_1x_2} R \leq \sigma, \\
 & g_2(x_1, x_2) = \frac{x_2}{\sqrt{(2x_1^2)} + 2x_1x_2} R \leq \sigma, \\
 & g_3(x_1, x_2) = \frac{1}{x_1 + \sqrt{(2)}x_2} R \leq \sigma, \\
 & 0 \leq x_1, x_2 \leq 1,
 \end{aligned}$$

where $L = 100 \text{ cm.}$, $R = 2 \text{ KN/cm}^2$ and $\sigma = 2 \text{ KN/cm}^2$.

Numerous studies have been published in the literature in an effort to solve this real-life problem. The results of this problem using SDDS-SABC have been compared with other algorithms including SC-GWO, PSO, wPSO, CS, Ray & Saini, Tsai, mGWO, wGWO, m-SCA, OBSCA, SSA, MFO, WOA, ISCA, Chaotic SSA and shown in Table 12.

8.2 Compression spring design problem

To get a minimum weight for a compression spring (see [19]), one needs to find the best values for the variables representing wire diameter (d), mean coil diameter (D), and the number of active coils (N). The following is the mathematical formulation of the problem where the limitations imposed on the objective function are stress, spike frequency, and deflection.

$$\begin{aligned}
 \text{Min } f_2(x) &= (x_3 + 2)x_2x_1^2, \\
 \text{s.t. } g_1(x) &= 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0, \\
 g_2(x) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\
 g_3(x) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\
 g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0, \\
 0.05 &\leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.30, 2 \leq x_3 \leq 15.
 \end{aligned}$$

This problem has been solved by several authors using their proposed algorithms such as; SC-GWO, GWO, PSO, PSO (He & Wang), GSA, SCA, GA, mGWO, wGWO, mSCA, OBSCA, MFO, WOA, SSA, ISCA, and Chaotic SSA. We compare their results with ones obtained by our SDDS-SABC algorithm in Table 12. The results are self-explanatory.

8.3 Cantilever beam design problem

This problem aims to reduce the cantilever beam's overall weight by optimizing the five hollow square cross-section specifications. The thickness of all the cross-sections is the same but has a different length. The problem includes five estimated parameters. One side of the liver is connected to a rigid body, and a load is attached to the other end. The formulation of this has been mathematically expressed as follows:

$$\begin{aligned}
\text{Min } f_3(x) &= 0.6224(x_1 + x_2 + x_3 + x_4 + x_5), \\
\text{s.t. } g(x) &= \frac{60}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0, \\
0.01 &\leq x_1, x_2, x_3, x_4, x_5 \leq 100.
\end{aligned}$$

The solution to this problem using our proposed SDDS-SABC algorithm has been displayed in Table 12. The results obtained by other algorithms such as BASZNN, BAS, BAS-WPT, BSAS, ZNN, ALO, GCA I, GCA II, CS, SOS, and EPO, have been shown in the same table for better comparison.

8.4 Pressure vessel design problem

The objective of this problem is to minimize the overall cost of the cylindrical pressure vessel in terms of material, forming, and welding under the nonlinear constraints of stresses and yield criteria. The decision parameters involve the thickness of the shell (T_{SH}), the thickness of the head (T_{HD}), inner radius (R), and the length of the cylindrical shell (L). Mathematically the problem has been expressed as follows: (see [19])

$$\begin{aligned}
\text{Min } f_4(x) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 19.84x_1^2x_3 + 3.1661x_1^2x_4, \\
x &= (x_1, x_2, x_3, x_4) = (T_{SH}, T_{HD}, R, L), \\
\text{s.t. } g_1(x) &= 0.0193x_3 - x_1 \leq 0, \\
g_2(x) &= 0.00954x_3 - x_1 \leq 0, \\
g_3(x) &= 1296000 - \frac{4}{3}\pi x_3^3 - \pi x_3^2x_4 \leq 0, \\
g_4(x) &= x_4 - 240 \leq 0, \\
1 \times 0.0625 &\leq x_1, x_2 \leq 99 \times 0.0625, \\
10 &\leq x_3, x_4 \leq 200.
\end{aligned}$$

The solution to this problem by our proposed SDDS-SABC algorithm has been shown in Table 12. We compare our results with state-of-the-art algorithms such as SC-GWO, GWO, PSO, SCA, GASA, GA, DE, Branch and Bound, Lagrangian Multiplier, ACO, ES, mGWO, wGWO, mSCA, OBSCA,

MFO, WOA, SSA, ISCA and Chaotic SSA (see Table 12). It has been observed that the proposed SDDS-SABC algorithm outperforms others.

8.5 Heat exchanger design problem

This is a challenging benchmark minimization problem because all the constraints are strictly enforced (Xin-She Yang and Amir H. Gandomi, 2012). It has eight design variables and six inequality restrictions, three linear and three nonlinear. The problem has been stated as follows:

$$\begin{aligned}
 & \text{Min } f_5(x) = x_1 + x_2 + x_3, \\
 & \text{s.t. } g_1(x) = 0.0025(x_1 + x_6) - 1 \leq 0, \\
 & g_2(x) = 0.0025(x_5 + x_7 - x_4) - 1 \leq 0, \\
 & g_3(x) = 0.01(x_8 - x_5) - 1 \leq 0, \\
 & g_4(x) = 833.33252x_4 + 100x_1 - x_1x_6 - 8333.333 \leq 0, \\
 & g_5(x) = 1250x_5 + x_2x_4 - x_2x_7 - 125x_4 \leq 0, \\
 & g_6(x) = x_3x_5 - 2500x_5 - x_3x_8 + 125 \times 10^4 \leq 0.
 \end{aligned}$$

As this is a challenging problem, several authors have not considered it under test except for the BAT algorithm. However, our proposed SDDS-SABC algorithm could solve this tough problem. The results of this problem obtained by both BAT and SDDS-SABC have been presented in Table 12.

Therefore, the results show that SDDS-SABC algorithm is efficient enough to extract the optimal values.

We study the robustness of our proposed dynamic penalty-based constraint handling techniques integrated into the SDDS-SABC method on EDPs (see Table 13).

9 Conclusion

A new and effective hybrid SDDS-SABC method for handling challenging restricted optimization issues was presented in this paper. As far as we are

Table 12: Result's comparison obtained from various algorithms for engineering design problems

| EDP1 | | EDP2 | | EDP3 | | EDP4 | | EDP5 | |
|-------------|-------------------|-------------|-----------------|-------------|------------------|-----------|---------------|-----------|------------------|
| Algorithm | best-found | Algorithm | best-found | Algorithm | best-found | Algorithm | best-found | Algorithm | best-found |
| SC-GWO | 263.8963 | SC-GWO | 0.012672 | SC-GWO | 6059.7179 | BASZNN | 1.3301 | BASZNN | - |
| SCA | 263.9506 | GWO | 0.012675 | GWO | 6136.6600 | BAS | 1.3331 | BAS | - |
| PSO | 263.8986 | PSO | 0.012675 | PSO | 6061.0777 | BAS-WPT | 1.3011 | BAS-WPT | - |
| wPSO | 263.8994 | SCA | 0.012678 | SCA | 6076.3651 | BSAS | 1.3000 | BSAS | - |
| GWO | 263.9497 | GSA | 0.012702 | GSA | 8538.8360 | ZNN | 1.3400 | ZNN | - |
| CS | 263.9716 | RW-GWO | 0.012674 | PSO | 6061.0780 | ALO | 1.3300 | ALO | - |
| Ray & Saini | 264.3000 | GA | 0.012705 | GA | 6288.7450 | GCA I | 1.3400 | GCA I | - |
| Tsai | 263.68 | PSO | 0.012675 | GA | 6410.3810 | GCA II | 1.3400 | GCA II | - |
| mGWO | 263.8967 (IF) | (He & Wang) | 0.012681 | DE | 6059.7340 | CS | 1.3399 | CS | - |
| wGWO | 263.8964 | ES | 0.012678 | ACO | 6059.0888 | SOS | 1.3300 | SOS | - |
| m-SCA | 263.9481 | CC | 0.012833 | ES | 6059.7456 | EPO | 1.1900 | EPO | - |
| OBSCA | 263.9463 | MO | 0.012730 | BB | 8129.1040 | SDDS-SABC | 1.1383 | BAT | 7049.2480 |
| SSA | 263.8958 | mGWO | 0.012676 | LM | 7198.043 | | | SDDS-SABC | 6948.2644 |
| MFO | 267.1922 | wGWO | 0.012672 | mGWO | 6059.7359 | | | | |
| WOA | 263.9858 | m-SCA | 0.012725 | wGWO | 6059.7207 | | | | |
| ISCA | 263.9002 | OBSCA | 0.012874 | m-SCA | 0.012725 | | | | |
| Chaotic SSA | 267.192 | MFO | 0.012758 | OBSCA | 0.012874 | | | | |
| SDDS-SABC | 263.605322 | WOA | 0.012676 | MFO | 6059.7143 | | | | |
| | | SSA | 0.012676 | WOA | 6059.7410 | | | | |
| | | ISCA | 0.01270 | SSA | 6059.7254 | | | | |
| | | Chaotic SSA | 0.012668 | ISCA | 0.01270 | | | | |
| | | SDDS-SABC | 0.018326 | Chaotic SSA | 0.012668 | | | | |
| | | | | SDDS-SABC | 6031.8439 | | | | |

CC - Constraint Correction; MO - Mathematical Optimization; BB - Branch and Bound; LM - Lagrangian Multiplier

Table 13: Result's comparison on performance of different penalty methods

| Problems | (a) | (b) | (c) | (d) |
|----------|-------------------|----------------|-----------------|-------------------|
| EDP1 | 263.6083840317 | 263.7490760091 | 263.6152947853 | 263.6053225360152 |
| EDP2 | Infeasible | Infeasible | Infeasible | 0.018326244566 |
| EDP3 | 5.906791130523017 | 3.46209183752 | 2.880236483910 | 1.138339974684857 |
| EDP4 | 6095.623409850 | 6113.738986468 | 6032.2416268761 | 6031.843954691072 |
| EDP5 | 7913.60005519897 | 8613.847481208 | 8527.930681880 | 6948.26443981868 |
| | (a) [35]; | (b): [34]; | (c): [33] ; | (d): Proposed |

aware, the ABC algorithm performed better at exploration than exploitation due to an imbalance between its exploration and exploitation capabilities. We have modified the ABC algorithm's startup step to maximize exploitation by producing the initial solution using a quasi-random sequence based on the Halton set. Additionally, we have enhanced the scout bee's phase using the sigmoid function and implemented a new search strategy scheme for the bees in use. To meet the constraints, we have designed a new penalty function that is something akin to dynamic penalty logic. The performance of our suggested SDDS-SABC approach is confirmed by the numerical results shown here. Our technique can be applied to other real-life restricted optimization problems in engineering and management, as demonstrated by the exciting experimental findings of real-life engineering design problems.

Acknowledgements

The first author is thankful to DST (Department of Science & Technology, New Delhi, India) for granting DST INSPIRE Research Fellowship (IF 190027) for pursuing Ph.D degree.

Declaration of Competing Interest

The authors declare that they have no known competing nancial interests or personal relationships that could have appeared to inuence the work reported in this paper.

References

- [1] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. and Gandomi, A. *The arithmetic optimization algorithm*. Comput. Methods Appl. Mech. Eng., 376 (2021) 113609.
- [2] Akashah, F. *A review of optimization techniques application for building performance analysis*. Civ. Eng. J., 8(4) (2022) 823–842.
- [3] Bansal, J., Joshi, S. and Sharma, H. *Modified global best artificial bee colony for constrained optimization problems*. Comput. Electr. Eng., 67 (2018) 365–382.
- [4] Bertsimas, D. and Tsitsiklis, J. *Simulated annealing*. Stat. Sci., 8 (1) (1993) 10–15.
- [5] Brajevic, I. *Crossover-based artificial bee colony algorithm for constrained optimization problems*. Neural Comput. Appl., 26 (6) (2015) 1587–1601.
- [6] Brajević, I. and Ignjatović, J. *An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems*. J. Intell. Manuf., 30 (7) (2019) 2545–2574.
- [7] Brajevic, I. and Tuba, M. *An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems*. J. Intell. Manuf., 24 (4) (2013) 729–740.
- [8] Cheng, Z., Song, H., Wang, J., Zhang, H., Chang, T. and Zhang, M. *Hybrid firefly algorithm with grouping attraction for constrained optimization problem*. Knowl.-Based Syst., 220 (2021) 106937.
- [9] Cui, L., Deng, J., Zhang, Y., Tang, G. and Xu, M. *Hybrid differential artificial bee colony algorithm for multi-item replenishment-distribution problem with stochastic lead-time and demands*. J. Clean. Prod., 254 (2020) 119873.
- [10] D’Angelo, G. and Palmieri, F. *GGA: A modified genetic algorithm with gradient-based local search for solving constrained optimization problems*. Inf. Sci., 547 (2021) 136–162.

- [11] Das, R., Das, K. and Mallik, S. *An improved quadratic approximation-based Jaya algorithm for two-echelon fixed-cost transportation problem under uncertain environment*. Soft Comput., 26 (2022) 10301–10320.
- [12] Deb, K. *An efficient constraint handling method for genetic algorithms*. Comput. Methods Appl. Mech. Eng., 186 (2000) 311–338.
- [13] Dorigo, M. and Di Caro, G. *Ant colony optimization: a new meta-heuristic*. Proc. Congr. Evol. Comput. (CEC99), 2 (1999) 1470–1477.
- [14] Duong, H., Nguyen, Q., Nguyen, D. and Van Nguyen, L. *PSO based hybrid PID-FLC Sugeno control for excitation system of large synchronous motor*. Emerg. Sci. J., 6(2) (2022) 201–216.
- [15] Fu, X., Pace, P., Aloï, G., Yang, L. and Fortino, G. *Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm*. Comput. Netw., 177 (2020) 107327.
- [16] Garg, H. *A hybrid GSA-GA algorithm for constrained optimization problems*. Inf. Sci., 478 (2019) 499–523.
- [17] Gu, X. *Application research for multiobjective low-carbon flexible job-shop scheduling problem based on hybrid artificial bee colony algorithm*. IEEE Access, 9 (2021) 135899–135914.
- [18] Guermoui, M., Gairaa, K., Boland, J. and Arrif, T. *A novel hybrid model for solar radiation forecasting using support vector machine and bee colony optimization algorithm: review and case study*. J. Sol. Energy Eng., 143 (2021) 020801.
- [19] Gupta, S. and Deep, K. *Enhanced leadership-inspired grey wolf optimizer for global optimization problems*. Eng. Comput., 36 (2020) 1777–1800.
- [20] Jabeen, S.D. *Vibration optimization of a passive suspension system via genetic algorithm*. Int. J. Model. Simul. Sci. Comput., 4 (2013) 1250022.
- [21] Jabeen, S.D. *Vehicle vibration and passengers comfort*. Int. Conf. Comput. Intell., vol 509. Springer, Singapore (2015) 357–372.

- [22] Javaheri, D., Gilani, A. and Ghaffari, A. *Energy-efficient routing in IoT networks with ABC optimization and machine learning for smart city infrastructure*. Front. Collaborative Res., 2 (2024) 1–13.
- [23] Jiao, L., Li, L., Shang, R., Liu, F. and Stolkin, R. *A novel selection evolutionary strategy for constrained optimization*. Inf. Sci., 239 (2013) 122–141.
- [24] Karaboga, D. *An idea based on honey bee swarm for numerical optimization*. Tech. Rep. TR06, Erciyes Univ., Fac. Eng. Comput., 2005.
- [25] Karaboga, D. and Akay, B. *A modified artificial bee colony (ABC) algorithm for constrained optimization problems*. Appl. Soft Comput., 11 (2011) 3021–3031.
- [26] Karaboga, D. and Basturk, B. *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm*. J. Global Optim., 39 (2007) 459–471.
- [27] Kennedy, J. and Eberhart, R. *Particle swarm optimization*. Proc. Int. Conf. Neural Netw. (ICNN'95), 4 (1995) 1942–1948.
- [28] Li, X. and Yin, M. *Self-adaptive constrained artificial bee colony for constrained numerical optimization*. Neural Comput. Appl., 24 (2014) 723–734.
- [29] Liang, J., Runarsson, T., Mezura-Montes, E., Clerc, M., Suganthan, P., Coello, C. and Deb, K. *Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization*. J. Appl. Mech., 41 (2006) 8–31.
- [30] Liang, R., Wu, C., Chen, Y. and Tseng, W. *Multi-objective dynamic optimal power flow using improved artificial bee colony algorithm based on Pareto optimization*. Int. Trans. Electr. Energy Syst., 26 (2016) 692–712.
- [31] Liu, F., Sun, Y., Wang, G. and Wu, T. *An artificial bee colony algorithm based on dynamic penalty and Lévy flight for constrained optimization problems*. Arab. J. Sci. Eng., 43 (2018) 7189–7208

- [32] Liu, H., Cai, Z. and Wang, Y. *Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization*. Appl. Soft Comput., 10 (2010) 629–640.
- [33] Liu, H., Xu, B., Lu, D. and Zhang, G. *A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm*. Appl. Soft Comput., 68 (2018) 360–376.
- [34] Liu, J., Teo, K., Wang, X. and Wu, C. *An exact penalty function-based differential search algorithm for constrained global optimization*. Soft Comput., 20 (2016) 1305–1313.
- [35] Liu, J., Wu, C., Wu, G. and Wang, X. *A novel differential search algorithm and applications for structure design*. Appl. Math. Comput., 268 (2015) 246–269.
- [36] Liu, M., Yuan, Y., Xu, A., Deng, T. and Jian, L. *A learning-based artificial bee colony algorithm for operation optimization in gas pipelines*. Inf. Sci., 690 (2025) 121593.
- [37] Long, W., Liang, X., Cai, S., Jiao, J. and Zhang, W. *An improved artificial bee colony with modified augmented Lagrangian for constrained optimization*. Soft Comput., 22 (2018) 4789–4810.
- [38] Mani, A. and Patvardhan, C. *A novel hybrid constraint handling technique for evolutionary optimization*. Proc. IEEE Congr. Evol. Comput., (2009) 2577–2583.
- [39] M'Dioud, M., Bannari, A., Er-Rays, Y., Bannari, R. and El Kafazi, I. *A Modified ABC Algorithm For The Best Placement Of DG Units*. Proc. Glob. Power, Energy Commun. Conf. (GPECOM), (2025) 392–399.
- [40] Mezura-Montes, E. and Cetina-Domínguez, O. *Empirical analysis of a modified artificial bee colony for constrained numerical optimization*. Appl. Math. Comput., 218 (2012) 10943–10973.
- [41] Mirjalili, S., Mirjalili, S. and Lewis, A. *Grey wolf optimizer*. Adv. Eng. Softw., 69 (2014) 46–61.

- [42] Mitchell, M., Holland, J. and Forrest, S. *When will a genetic algorithm outperform hill climbing*. Adv. Neural Inf. Process. Syst., 6 (1993) 51–58.
- [43] Oubbati, O., Khan, A. and Liyanage, M. *Blockchain-enhanced secure routing in FANETs: Integrating ABC algorithms and neural networks for attack mitigation*. Synthesis, 2 (2024) 1–11.
- [44] Patra, J., Yadav, A., Verma, R., Pal, N., Samantaray, S., Sahu, K., Singh, P., Parihar, R. and Panda, A. *Efficient multi-objective approach using ABC algorithm for minimizing generation fuel cost and transmission loss through FACTS controllers placement and sizing*. Iran. J. Sci. Technol. Trans. Electr. Eng., 49 (2025) 1313–1335.
- [45] Peng, C., Liu, H. and Gu, F. *A novel constraint-handling technique based on dynamic weights for constrained optimization problems*. Soft Comput., 22 (2018) 3919–3935.
- [46] Phoemphon, S. *Grouping and reflection of the artificial bee colony algorithm for high-dimensional numerical optimization problems*. IEEE Access, 12 (2024) 91426–91446.
- [47] Pramanik, P. and Maiti, M. *An inventory model for deteriorating items with inflation induced variable demand under two level partial trade credit: A hybrid ABC-GA approach*. Eng. Appl. Artif. Intell., 85 (2019) 194–207.
- [48] Price, K. *Differential evolution vs. the functions of the 2/sup nd/ICEO*. Proc. IEEE Int. Conf. Evol. Comput., (1997) 153–157.
- [49] Pu, Q., Xu, C., Wang, H. and Zhao, L. *A novel artificial bee colony clustering algorithm with comprehensive improvement*. Vis. Comput., 38 (2022) 1395–1410.
- [50] Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S. *GSA: a gravitational search algorithm*. Inf. Sci., 179 (2009) 2232–2248.
- [51] Rathod, V., Gumaste, S., Guttula, R., Zade, S. and Singh, R. *Optimization of energy consumption in mobile Ad-Hoc networks with a swarm intelligence-based ABC algorithm*. Discov. Appl. Sci., 7 (2025) 805.

- [52] Rezaee Jordehi, A. *A chaotic-based big bang–big crunch algorithm for solving global optimisation problems*. Neural Comput. Appl., 25 (2014) 1329–1335.
- [53] Runarsson, T. and Yao, X. *Stochastic ranking for constrained evolutionary optimization*. IEEE Trans. Evol. Comput., 4 (2000) 284–294.
- [54] Satapathy, S. and Naik, A. *Data clustering based on teaching-learning-based optimization*. Proc. Int. Conf. Swarm, Evol. Memetic Comput., (2011) 148–156.
- [55] Şenel, F., Gökçe, F., Yüksel, A. and Yiğit, T. *A novel hybrid PSO–GWO algorithm for optimization problems*. Eng. Comput., 35 (2019) 1359–1373.
- [56] Sharma, D. and Jabeen, S. *Hybridizing interval method with a heuristic for solving real-world constrained engineering optimization problems*. Struct., 56 (2023) 104993.
- [57] Shi, Y. *Brain storm optimization algorithm*. Proc. Int. Conf. Swarm Intell., (2011) 303–309.
- [58] Surono, S., Goh, K., Onn, C., Nurraihan, A., Siregar, N., Saeid, A. and Wijaya, T. *Optimization of Markov weighted fuzzy time series forecasting using genetic algorithm (GA) and particle swarm optimization (PSO)*. Emerg. Sci. J., 6 (2022) 1375–1393.
- [59] Takahama, T. and Sakai, S. *Efficient constrained optimization by the ε constrained adaptive differential evolution*. Proc. IEEE Congr. Evol. Comput., (2010) 1–8.
- [60] Tessema, B. and Yen, G. *An adaptive penalty formulation for constrained evolutionary optimization*. IEEE Trans. Syst. Man Cybern. A, 39 (2009) 565–578.
- [61] Tran, S., Vu, H., Pham, T. and Hoang, D. *Constrained Pareto-Based Weighted-Sum ABC Algorithm for Efficient Sensor Networks Deployment*. Proc. Int. Conf. Green Technol. Sustain. Dev., (2024) 310–321.

- [62] Wang, Y., Cai, Z., Guo, G. and Zhou, Y. *Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems*. IEEE Trans. Syst. Man Cybern. B, 37 (2007) 560–575.
- [63] Wang, Z. and Kong, X. *An enhanced artificial bee colony algorithm for constraint optimization*. Eng. Lett., 32 (2024) 276.
- [64] Yeniyay, O. *Penalty function methods for constrained optimization with genetic algorithms*. Math. Comput. Appl., 10 (2005) 45–56.
- [65] Yesodha, K., Krishnamurthy, M., Selvi, M. and Kannan, A. *Intrusion detection system extended CNN and artificial bee colony optimization in wireless sensor networks*. Peer-to-Peer Netw. Appl., 17 (2024) 1237–1262.
- [66] Zhang, X., Lou, Y., Yuen, S., Wu, Z., He, Y. and Zhang, X. *Hybrid artificial bee colony with covariance matrix adaptation evolution strategy for economic load dispatch*. Proc. IEEE Congr. Evol. Comput. (CEC), (2019) 204–209.
- [67] Zhang, Z., Ding, S. and Jia, W. *A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems*. Eng. Appl. Artif. Intell., 85 (2019) 254–268.
- [68] Zhu, G. and Kwong, S. *Gbest-guided artificial bee colony algorithm for numerical function optimization*. Appl. Math. Comput., 217 (2010) 3166–3173.



A quadrature method for Volterra integral equations of the first kind

S.A. Hosseini 

Abstract

This paper introduces a direct quadrature method for the numerical solution of Volterra integral equations of the first kind, utilizing a composite quadrature scheme based on the Floater–Hormann family of linear barycentric rational interpolants. The convergence of the proposed method is rigorously proved, and the order of convergence is explicitly derived in terms of the parameters of the method, thereby providing a clear theoretical framework for its performance. Several numerical experiments are provided to demonstrate both the efficiency and accuracy of the method, as well as to verify the excellent agreement between the implementation results and the theoretically predicted convergence rates.

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

Keywords: Volterra integral equations; Direct quadrature method; Rational interpolation; Barycentric form.

Received 19 August 2025; revised 4 September 2025; accepted 6 September 2025

Seyyed Ahmad Hosseini

Department of Mathematics, Faculty of Sciences, Golestan University, Gorgan, Iran.

email: a.hosseini@gu.ac.ir

How to cite this article

Hosseini, S.A., A quadrature method for Volterra integral equations of the first kind. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1589-1606.
<https://doi.org/10.22067/ijnao.2025.95004.1712>

1 Introduction

This paper is concerned with the numerical solution of the classical Volterra integral equations (VIEs) of the first kind

$$\int_{t_0}^t k(t, s, y(s)) \, ds = g(t), \quad t \in I = [t_0, T], \quad (1)$$

where $y : I \rightarrow \mathbb{R}$ is an unknown function, and $k : S \times \mathbb{R} \rightarrow \mathbb{R}$ with $S = \{(t, s) : t_0 \leq s \leq t \leq T\}$, represents the kernel of the equation. In practice, the functions g and k (except for its third variable) are typically used only at equispaced values of the variables. In what follows, we assume that $g(t_0) = 0$, and the functions g and k are smooth enough such that VIE (1) has a unique solution [21].

Volterra-type equations play a crucial role in modeling various dynamic systems where the current state of the system depends not only on its current conditions, but also on the accumulated effects of past interactions. In these equations, the related quantity varies in time and simultaneously depends on its past values. VIEs appear in various scientific and engineering disciplines, including physics, biology, engineering, and finance. For example, in viscoelasticity, they can describe how materials respond to stress over time, considering past deformations. In population dynamics, they help model species interactions by incorporating the influence of previous population levels. One of the notable applications of the first kind VIEs is in the field of epidemiology, particularly in the modeling of population dynamics during the spread of infectious diseases. Using VIEs in this context helps capture more realistically the interaction between susceptible, infected, and recovered individuals over time, incorporating the history of infection events and their cumulative effect on the population.

Most real-world problems are so complicated that there is no hope of finding an analytical solution. As a result, numerical methods are often needed to obtain solutions. Specifically, Volterra-type equations also necessitate numerical methods that yield approximations to the exact solutions. The development of a numerical solver for VIEs is a wide and mature area of research; see for instance [9, 10, 21] and the references therein.

A broad range of numerical methods across various classes has been developed for VIEs of the first kind (see, e.g., [13, 15, 26]). For smooth kernels, first-kind VIEs can be transformed into equivalent second-kind equations. Where the kernels are explicitly defined and differentiable, suitable numerical techniques tailored for second-kind VIEs can be applied, ensuring more efficient and accurate solutions.

Direct quadrature methods are among the simplest and most traditional schemes for solving VIEs. These methods commonly employ composite Newton–Cotes formulas, Gregory’s rules, and hybrid schemes, providing a clear and efficient strategy for the numerical solution of VIEs. Nevertheless, their effectiveness diminishes when higher accuracy is required. They also often exhibit instability and loss of precision, particularly when applied to problems with smooth kernels over long intervals or under conditions of increasing mesh density. Moreover, it is important to note that, as the degree of the method increases, these schemes can suffer from Runge’s phenomenon, wherein oscillations lead to significant numerical errors, and making them impractical [12].

A more robust alternative is to replace polynomial interpolation, which forms the foundation of many traditional numerical methods, with linear barycentric rational interpolation, which is characterized by barycentric weights, one for every node. The weights are chosen in such a way that bad properties of the polynomial such as ill-conditioning and Runge’s phenomenon are avoided, and convergence, well-conditioning and absence of poles are guaranteed. Berrut [5] presented a very simple choice of the barycentric weights that successfully avoids poles in the interpolation interval. However, despite the excellent conditioning of the resulting linear barycentric rational interpolants (LBRI), their convergence rate remains slow for general node distributions. The situation changed significantly with the introduction of a new family of LBRI by Floater and Hormann in [11], a family of barycentric rational interpolants based on a blend of the local polynomial interpolants, which depends on a parameter d , and including the previously introduced interpolants. This family of LBRI presents a favorable comparison to more classical polynomial interpolants, for interpolation of univariate data, especially in the equispaced setting. Indeed,

the Lebesgue constant associated with these interpolants exhibits logarithmic growth in this situation, a stark contrast to the exponential growth experienced by polynomials. This logarithmic growth implies that the error increases at a much slower rate, and making them more advantageous for accurate interpolation in such settings compared to classical polynomial interpolants [7, 8]. Moreover, the flexibility, robustness, and favorable convergence rate, make these tools a state-of-the-art method for interpolation at equispaced nodes. Due to these attractive features, this family of LBRIs has recently gained popularity and has been used in the construction of various numerical methods for solving different classes of time-dependent problems [1, 2, 3, 4, 6, 18, 19, 20, 22, 23]. The spirit of this paper is that of deriving a highly accurate and stable scheme based on the composite barycentric rational quadrature (CBRQ) introduced in [6] for the numerical solution of VIEs of the first kind (1).

After briefly reviewing the LBRIs in Section 2, a method for solving VIEs (1) based on the CBRQ rule will be introduced in Section 3. This section further provides a rigorous convergence analysis of the method and its order of accuracy. The robustness and efficiency of the method and the theoretical results on its order of convergence are illustrated by some numerical experiments in Section 4.

2 Linear barycentric rational quadrature

Quadrature formulas constitute a fundamental component of simulation, data analysis, and numerical modeling, playing a vital role in addressing practical problems across many fields of computational sciences and engineering. A natural and widely adopted approach for approximating the definite integral of a function over a bounded interval is to replace the integrand with a suitable interpolant and apply the integration operator to the resulting approximation. In particular, linear interpolation schemes trivially lead to quadrature rules through this process. Prior to reviewing the idea underlying the CBRQ rule, we first give a short introduction to the Floater–Hormann family of LBRIs.

Let f be a real-valued and continuously differentiable function over the interval $[a, b]$, and consider the $n + 1$ distinct interpolation nodes

$$a = t_0 < t_1 < \cdots < t_n = b.$$

The Floater–Hormann family of the LBRIs to interpolate the given $n + 1$ pairs $(t_j, f(t_j))$, with distinct nodes t_j , $j = 0, 1, \dots, n$, for every fixed nonnegative integer $d \leq n$, takes the barycentric form [11]

$$r_{n,d}[f](t) = \sum_{k=0}^n b_k^{(n,d)}(t) f(t_k), \quad b_k^{(n,d)}(t) = \frac{\beta_k^{(n,d)}}{t - t_k} \bigg/ \sum_{j=0}^n \frac{\beta_j^{(n,d)}}{t - t_j}, \quad (2)$$

with the barycentric weights

$$\beta_k^{(n,d)} = \sum_{i=\max(0, k-d)}^{\min(k, n-d)} (-1)^i \prod_{j=i, j \neq k}^{i+d} \frac{1}{t_k - t_j}, \quad (3)$$

where $0 \leq d \leq n$. We denote the function to be interpolated by f , to avoid confusion with the functions y and g in VIE (1). The following theorem from [11] gives the rate of convergence of this family of the LBRIs via a bound on the interpolation error in the maximum norm.

Theorem 1. For any $f \in C^{d+2}[a, b]$, we have

$$\|r_{n,d}[f] - f\| \leq Ch^{d+1},$$

where $h = \max_{0 \leq k \leq n-1} (t_{k+1} - t_k)$ is the global mesh size and the constant C depends only on d , the derivatives of f , the interval length $b - a$, and, only in the case $d = 0$, on the maximal local mesh ratio

$$\rho = \max_{1 \leq k \leq n-2} \min \left\{ \frac{t_{k+1} - t_k}{t_k - t_{k-1}}, \frac{t_{k+1} - t_k}{t_{k+2} - t_{k+1}} \right\}.$$

Moreover, according to [11, Theorem 2], in the case of odd $n - d$, the bound on the interpolation error involves an additional factor, nh , so the order of convergence is one unit larger than the stated above, that is, $d + 1 + \delta$, where $\delta = 1$ for odd $n - d$ and $\delta = 0$ for even $n - d$.

Throughout this work, we are mainly interested in the case of uniformly spaced nodes, when the weights in (3) can be replaced by

$$\bar{\beta}_k^{(n,d)} = (-1)^d d! h^d \beta_k^{(n,d)} = (-1)^k \sum_{i=d}^n \binom{d}{i-k}, \quad k = 0, 1, \dots, n. \quad (4)$$

The linearity of barycentric rational interpolation in data renders it well-suited for applications. Klein and Berrut [17] introduced a global quadrature formula based on integrating the LBRI (2) corresponding to the real integrable function f over the integration interval $[a, b]$ of the form

$$\begin{aligned} \int_a^b f(t) dt &\approx \int_a^b r_{n,d}[f](t) dt \\ &= h \sum_{k=0}^n w_{n,k} f_k = Q_n^G, \end{aligned} \quad (5)$$

with quadrature weights

$$w_{n,k} = h^{-1} \int_a^b b_k^{(n,d)}(t) dt = \int_0^n \phi_k^{(n,d)}(x) dx, \quad (6)$$

where

$$\phi_k^{(n,d)}(x) = \frac{\bar{\beta}_k^{(n,d)}}{x-k} \bigg/ \sum_{j=0}^n \frac{\bar{\beta}_j^{(n,d)}}{x-j},$$

with $\bar{\beta}_k^{(n,d)}$ as in (4).

The integrands in (6) are rational functions that, in general, cannot be integrated analytically without additional knowledge of their properties, such as the locations of their poles. Furthermore, algebraic methods typically require the polynomials in the numerator and denominator of the integrand to be in canonical form, a representation often impaired by stability issues. As a result, these integrals must be computed numerically up to machine precision, using, for instance, the routines available in the Chebfun system [24] or, alternatively, with Gauss–Legendre or Clenshaw–Curtis quadrature rules [14, 25]. Notably, the barycentric form exhibits greater flexibility compared to Gauss–Legendre quadrature, as it allows for arbitrary node distributions rather than restricting nodes to the roots of Legendre polynomials. Moreover,

barycentric rational quadrature is particularly effective at handling endpoint singularities and functions with steep gradients, owing to its basis in rational approximation.

The convergence and stability of the resulting quadrature rule directly inherit the corresponding properties of the underlying interpolant. It was proved in [17] that for nonnegative integers n and d , $d \leq n/2 - 1$, and $f \in C^{d+3}[a, b]$, the quadrature formula (5) with quadrature weights (6) converges at the rate $O(h^{d+2})$ as the global mesh size h tends to zero if the quadrature weights given by (6) are approximated by a quadrature rule converging at least at the rate $O(h^{d+2})$.

Direct application of the quadrature formula in discretization schemes for time-dependent problems, specifically VIEs, implies significant computational cost due to the necessity of computing quadrature weights at each time step as the partition size grows. However, since the barycentric weights (6) do not depend on the nodes and are translation invariant, it is possible to develop a composite version of the quadrature rule that addresses this issue efficiently.

Consider the interval $[a, b]$ partitioned uniformly by points $a = t_0 < t_1 < \dots < t_N = b$ with step size $h = \frac{b-a}{N}$, so that $t_k = a + kh$ for $k = 0, 1, \dots, N$. Let d and n satisfy $0 \leq d \leq n \leq N/2$, and define $p = \lfloor \frac{N}{n} \rfloor - 1$. Under these conditions, the CBRQ rule can be formulated as

$$\begin{aligned} \int_{t_0}^{t_N} f(t) dt &= \sum_{j=0}^{p-1} \int_{t_{jn}}^{t_{(j+1)n}} f(t) dt + \int_{t_{pn}}^{t_N} f(t) dt \\ &\approx h \sum_{j=0}^{p-1} \sum_{k=0}^n w_{n,k} f_{jn+k} + h \sum_{k=0}^{N-pn} w_{N-pn,k} f_{pn+k} = Q_N^C, \end{aligned} \quad (7)$$

where

$$w_{i,k} = h^{-1} \int_{t_0}^{t_i} b_k^{(i,d)}(t) dt = \int_0^i \phi_k^{(i,d)}(x) dx, \quad (8)$$

for $i = n, n+1, \dots, 2n-1$ and $k = 0, \dots, i$. Note that for $n \leq N \leq 2n$, the only contributing term in (7) is the last one, which is precisely the global quadrature formula given in (5).

Based on this construction, and noting that each local quadrature formula converges at the rate of $O(h^{d+2})$, and that n is fixed and there are $p+1 =$

$O(n) = O(1/h)$ integrals to be computed, the order of the CBRQ rule (7) behaves as follows.

Theorem 2. Suppose that N and n are positive integers with $n \leq N$, that d is a nonnegative integer with $d \leq n \leq N/2$, and that $f \in C^{d+2}[a, b]$. Then the absolute error in the approximation of the integral of f with the composite quadrature rule (7) goes to zero as $O(h^{d+1+\delta})$, where $\delta = 0$ if $n - d$ is even and $\delta = 1$ if $n - d$ is odd.

3 Description of the method for VIEs of the first kind

In this section, we utilize the favorable properties of the introduced CBRQ rule, including smoothness, high accuracy, and an arbitrarily high rate of convergence, to construct a direct quadrature-based scheme for numerically solving the classical VIEs of the form (1). Differentiating (1) yields VIEs of the second kind

$$k(t, t, y(t)) + \int_{t_0}^t k_t(t, s, y(s)) \, ds = g'(t), \quad t \in I, \quad (9)$$

where k_t is the partial derivative of the kernel k with respect to t .

Let $T_N = \{t_0, t_1, \dots, t_N = T\}$ be a uniform partition of the given interval I with the fixed stepsize $h = t_{i+1} - t_i = (T - t_0)/N$, $i = 0, 1, \dots, N - 1$ and assume that d and n are as introduced in section 2, and that $p = \lfloor m/n \rfloor - 1$. Applying the CBRQ rule (7) to the integral part of (9) at the mesh point t_m yields

$$\begin{aligned} k(t_m, t_m, y_m) + h \sum_{j=0}^{p-1} \sum_{k=0}^n w_{n,k} k_t(t_m, t_{jn+k}, y_{jn+k}) \\ + h \sum_{k=0}^{m-pn} w_{m-pn,k} k_t(t_m, t_{pn+k}, y_{pn+k}) = g'(t_m), \end{aligned} \quad (10)$$

for $m = n + 1, \dots, N$. The quadrature weights $w_{i,k}$ are given by (8) for $i = n, n + 1, \dots, 2n - 1$ and $k = 0, \dots, i$. Here y_m denotes the approximation to the exact solution y of (1) at $t = t_m$. This approach will be referred to as

the CBRQM, which stands for “composite barycentric rational quadrature method”.

It is clear that a set of starting values y_m , $m = 1, 2, \dots, n$, is necessary to prevent deterioration in the order of convergence of the method, as any loss of precision will be carried over through the whole interval of integration. To supply them, we employ the quadrature formula (5) to approximate the integral part of (9) over the interval $[t_0, t_m]$, $m = 1, 2, \dots, n$, which gives

$$k(t_m, t_m, y_m) + h \sum_{k=0}^n \bar{w}_{m,k} k_t(t_m, t_k, y_k) = g'(t_m), \quad (11)$$

where the quadrature weights required for the starting procedure are given by

$$\bar{w}_{m,k} = \int_0^m \frac{\bar{\beta}_k^{(n,d)}}{x-k} \bigg/ \sum_{j=0}^n \frac{\bar{\beta}_j^{(n,d)}}{x-j} dx, \quad k = 0, 1, \dots, n,$$

wherein the barycentric weights $\bar{\beta}_j^{(n,d)}$ depend on n , not on m . This starting procedure is fully implicit and specifically designed to provide sufficiently accurate starting values. It is essential to emphasize that the nonlinear system of equations represented by (11) contains n equations in the n unknowns y_m , $m = 1, 2, \dots, n$. This system must be solved simultaneously to obtain the starting values required for the subsequent implementation of the method. The following theorem rigorously establishes the convergence rate of the starting values derived from (11) in terms of the parameters of the method.

Theorem 3. Assume that $f \in C^{d+2}(I)$ and $k \in C^{d+2}(S \times \mathbb{R})$, let $\mathbf{y}_n = (y_1, y_2, \dots, y_n)^T$ be the approximate values obtained by the starting procedure (11) with $d \leq n$, and let $\mathbf{e}_n = (e_1, e_2, \dots, e_n)^T$, where $e_i = y(t_i) - y_i$, $i = 1, 2, \dots, n$, are the starting errors. Then, $\|\mathbf{e}_n\|_\infty$ goes to zero as $O(h^{d+2+\delta})$, where $\delta = 0$ for even values of $n - d$ and $\delta = 1$ for odd values of $n - d$.

Proof. Substituting the exact values for y in the starting procedure (11) and incorporating their consistency error $R_m(h)$ gives

$$k(t_m, t_m, y(t_m)) + h \sum_{k=0}^n \bar{w}_{m,k} k_t(t_m, t_k, y(t_k)) + R_m(h) = g'(t_m). \quad (12)$$

Subtracting equation (9), in which the variable t is replaced by t_m , from (12) for each $m = 1, 2, \dots, n$, yields

$$R_m(h) = \int_{t_0}^{t_m} k_t(t_m, s, y(s)) \, ds - h \sum_{k=0}^n \bar{w}_{m,k} k_t(t_m, t_k, y(t_k)).$$

Since $m \leq n$, n is constant, and h shrinks, it readily follows from the convergence rate of the global quadrature (5) that for each $m = 1, 2, \dots, n$, $R_m(h) = O(h^{d+2})$. Considering the arguments mentioned just after Theorem 1, it can be deduced that when $n - d$ is odd, the convergence order of the starting values increases by one, yielding $d + 2 + \delta$, where δ is the same quantity as stated in Theorem 2.

Subtracting the starting values in (11) from (12) and using the mean value theorem gives

$$k_y(t_m, t_m, \xi_m) e_m + h \sum_{k=0}^n \bar{w}_{m,k} k_{ty}(t_m, t_k, \eta_k) e_k = R_m(h), \quad m = 1, 2, \dots, n,$$

where k_y and k_{ty} denote the partial derivatives of k with respect to y and t , y , respectively, and where ξ_m and η_k lie within the interior of the line segments connecting the exact value of y and its approximations at the corresponding functions. Introducing the matrix \mathcal{D}_n as the diagonal matrix with entries $k_y(t_m, t_m, \xi_m)$ for $m = 1, 2, \dots, n$, the matrix \mathcal{W}_n with the (m, k) th element given by $\bar{w}_{m,k} k_{ty}(t_m, t_k, \eta_k)$, and the consistency error vector $\mathbf{R}_n(h) := [R_1(h), R_2(h), \dots, R_n(h)]^T$, the last equation can be written in matrix form as

$$(\mathcal{D}_n + h\mathcal{W}_n)\mathbf{e}_n = \mathbf{R}_n(h).$$

Due to the differentiability assumption on the kernel k , the partial derivatives involved in the matrices have a maximum absolute value, and since n is fixed in the starting procedure (11) and only the stepsize h varies, the corresponding starting quadrature weights remain bounded as well. Consequently, the norm of the matrix \mathcal{W}_n is bounded so that $h \|\mathcal{W}_n\|_\infty$ may be made as small

as necessary by diminishing h . Therefore, with small enough stepsize h , there exists a positive constant C such that

$$\begin{aligned}\|\mathbf{e}_n\|_\infty &\leq \|(\mathcal{D}_n + h\mathcal{W}_n)^{-1}\|_\infty \|\mathbf{R}_n(h)\|_\infty \\ &\leq \frac{1}{\|\mathcal{D}_n\|_\infty - h\|\mathcal{W}_n\|_\infty} \|\mathbf{R}_n(h)\|_\infty \\ &\leq C \|\mathbf{R}_n(h)\|_\infty,\end{aligned}$$

which implies $\|\mathbf{e}_n\|_\infty = O(h^{d+2+\delta})$. \square

We are now in a position to state our main theorem about the order of convergence of the method (10), which can easily be deduced with the same ingredients as in [21, Theorem 7.2], and the help of Theorems 2 and 3.

Theorem 4. Let $g \in C^{d+2}(I)$ and let $k \in C^{d+2}(S \times \mathbb{R})$, where the kernel k satisfies a Lipschitz condition with respect to its third argument. Assume further that n and d with $d \leq n$ are, respectively positive and nonnegative integers, and let the nodes be equispaced. Then, the CBRQM (10) is convergent of order $d + 1 + \delta$ if the order of the utilized starting procedure is at least $d + \delta$, where $\delta = 0$ for even values of $n - d$, and $\delta = 1$ for odd values of $n - d$.

4 Numerical experiments

In this section, we apply the proposed method with various choices of n , d , and d_s (for the starting procedure), to several linear and nonlinear VIEs of the first kind to illustrate the efficiency and accuracy of the method and verify the theoretical convergence estimates established in section 3. To this end, the approximation quality in each numerical experiment is measured by

$$e_h^S = \max_{1 \leq m \leq n} \|y(t_m) - y_m\|_\infty,$$

the maximum norm of the starting errors, and

$$e_h(T) = \|y(t_N) - y_N\|_\infty,$$

the maximum norm of the error at the endpoint $t_N = T$ of the integration interval. Additionally, to validate the theoretical convergence order, the experimental estimate of the order of accuracy for the starting procedure (11) and the CBRQM (10) are computed by

$$O_S = \log_2(e_h^S/e_{h/2}^S),$$

and

$$O_C = \log_2(e_h^C(T)/e_{h/2}^C(T)).$$

As a first example, consider the linear convolution VIE of the first kind [13]

$$\int_0^t (c^2 + 1) \cos(t-s)y(s) ds = ce^{ct} + \sin t - c \cos t, \quad t \in [0, 4], \quad (13)$$

where $c = \pm 1$, and the exact solution is given by $y(t) = e^t$. Tables 1 and 2 list numerical results for the starting procedure and the CBRQM with various choices of the parameters (n, d, d_s) and different values of the stepsize h . For both cases of the parameter c , the errors decrease with decreasing stepsize h . As to be expected from Theorem 3, the error of the starting values decreases at the rate of $d_s + 2 + \delta$, with $\delta = 1$ for odd $n - d_s$ and $\delta = 0$ for even $n - d_s$, and the errors of the CBRQM (10) decrease at the rate of $d + 1 + \delta$, with $\delta = 1$ for odd $n - d$ and $\delta = 0$ for even $n - d$, as established by Theorem 4.

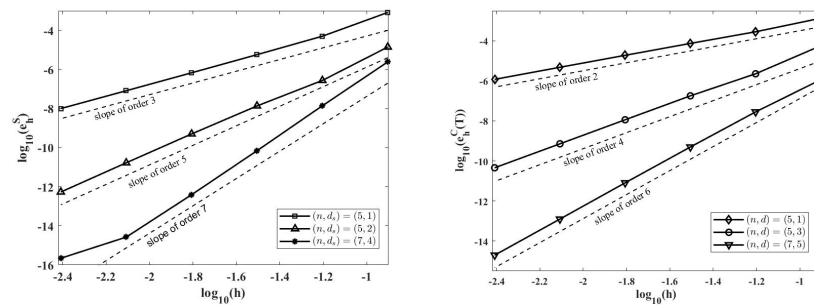
Table 1: Numerical results of the CBRQM applied to the VIE in (13) with $c = 1$.

| h | | 2^{-2} | 2^{-3} | 2^{-4} | 2^{-5} | 2^{-6} | 2^{-7} | 2^{-8} |
|---------------------------|------------|----------|----------|----------|----------|----------|----------|----------|
| $(n, d, d_s) = (5, 2, 1)$ | | | | | | | | |
| Starting procedure | e_h^S | 5.90e-3 | 4.63e-4 | 4.76e-5 | 5.46e-6 | 6.54e-7 | 8.02e-8 | 9.92e-9 |
| | O_S | | 3.67 | 3.28 | 3.12 | 3.06 | 3.03 | 3.02 |
| CBRQM | $e_h^C(T)$ | 1.72e-2 | 1.26e-3 | 7.73e-5 | 4.69e-6 | 2.94e-7 | 1.84e-8 | 1.15e-9 |
| | O_C | | 3.77 | 4.03 | 4.04 | 4.00 | 4.00 | 4.00 |
| $(n, d, d_s) = (6, 3, 2)$ | | | | | | | | |
| Starting procedure | e_h^S | 8.28e-4 | 3.18e-5 | 1.48e-6 | 7.96e-8 | 4.60e-9 | 2.77e-10 | 1.70e-11 |
| | O_S | | 4.70 | 4.43 | 4.22 | 4.11 | 4.05 | 4.03 |
| CBRQM | $e_h^C(T)$ | 1.02e-3 | 2.39e-5 | 7.55e-7 | 2.78e-8 | 1.03e-9 | 3.52e-11 | 1.15e-12 |
| | O_C | | 5.42 | 4.98 | 4.76 | 4.75 | 4.87 | 4.94 |

As the second test problem, consider the classical VIE of the form [16]

Table 2: Numerical results of the CBRQM applied to the VIE in (13) with $c = -1$.

| h | | 2^{-2} | 2^{-3} | 2^{-4} | 2^{-5} | 2^{-6} | 2^{-7} | 2^{-8} |
|---------------------------|------------|----------|----------|----------|----------|----------|----------|----------|
| $(n, d, d_s) = (5, 2, 1)$ | | | | | | | | |
| Starting procedure | e_h^S | 1.50e-3 | 2.43e-4 | 3.48e-5 | 4.66e-6 | 6.05e-7 | 7.71e-8 | 9.73e-9 |
| | O_S | | 2.63 | 2.80 | 2.90 | 2.95 | 2.97 | 2.99 |
| CBRQM | $e_h^C(T)$ | 3.10e-3 | 2.91e-4 | 2.18e-5 | 1.49e-6 | 9.74e-8 | 6.22e-9 | 3.93e-10 |
| | O_C | | 3.41 | 3.74 | 3.87 | 3.94 | 3.97 | 3.98 |
| $(n, d, d_s) = (6, 3, 2)$ | | | | | | | | |
| Starting procedure | e_h^S | 1.75e-4 | 1.43e-5 | 1.00e-6 | 6.57e-8 | 4.19e-9 | 2.64e-10 | 1.66e-11 |
| | O_S | | 3.61 | 3.84 | 3.93 | 3.97 | 3.99 | 3.99 |
| CBRQM | $e_h^C(T)$ | 2.21e-4 | 1.91e-5 | 9.06e-7 | 3.39e-8 | 1.15e-9 | 3.75e-11 | 1.19e-12 |
| | O_C | | 3.53 | 4.40 | 4.74 | 4.88 | 4.94 | 4.98 |

Figure 1: Log-log-plots of the approximation error of the starting procedure and the CBRQM applied to the VIE in (14) with $(n, d_s) = (5, 1), (5, 2), (7, 4)$ (left) and $(n, d) = (5, 1), (5, 3), (7, 5)$ (right).

$$\int_0^t e^{-ts} y(s) ds = g(t), \quad t \in [0, 1], \quad (14)$$

where the function g is chosen such that the exact solution is $y(t) = e^{-t} \cos t$. Figure 1 shows the logarithmic errors $\log_{10}(e_h^S)$ and $\log_{10}(e_h^C(T))$ of the starting procedure and the CBRQM (10) for this equation, plotted versus $\log_{10}(h)$, together with the slope lines corresponding to the expected convergence rates. The numerical results confirm the expected orders of convergence for various choices of (n, d, d_s) as predicted by Theorems 3 and 4.

To demonstrate the efficiency and accuracy of the CBRQM (10), consider the following nonlinear first-kind VIE

$$\int_0^t \frac{1}{t+s+9+e^{y(s)}} ds = \frac{1}{2} \log\left(\frac{3t+10}{t+10}\right), \quad t \in [0, 1], \quad (15)$$

with the exact solution $y(t) = \log(t + 1)$. The numerical results for this equation are presented in Table 3 for the parameter $(n, d, d_s) = (3, 1, 1)$ and $(n, d, d_s) = (8, 3, 2)$. These results clearly validate the theoretical convergence order of the proposed method.

Table 3: Numerical results of the CBRQM applied to the VIE in (15).

| h | | 2^{-3} | 2^{-4} | 2^{-5} | 2^{-6} | 2^{-7} | 2^{-8} |
|---------------------------|------------|----------|----------|----------|----------|----------|----------|
| $(n, d, d_s) = (3, 1, 1)$ | | | | | | | |
| Starting procedure | e_h^S | 2.71e-5 | 3.97e-6 | 5.44e-7 | 7.13e-8 | 9.14e-9 | 1.16e-9 |
| | O_S | | 2.77 | 2.87 | 2.93 | 2.96 | 2.98 |
| CBRQM | $e_h^C(T)$ | 3.66e-5 | 9.51e-6 | 2.79e-6 | 7.02e-7 | 1.82e-7 | 4.55e-8 |
| | O_C | | 1.94 | 1.77 | 1.99 | 1.95 | 2.00 |
| $(n, d, d_s) = (8, 3, 2)$ | | | | | | | |
| Starting procedure | e_h^S | 1.12e-6 | 8.42e-8 | 5.83e-9 | 3.85e-10 | 2.48e-11 | 1.57e-12 |
| | O_S | | 3.73 | 3.85 | 3.92 | 3.96 | 3.98 |
| CBRQM | $e_h^C(T)$ | 1.88e-7 | 4.57e-9 | 1.68e-10 | 5.74e-12 | 1.87e-13 | 5.33e-15 |
| | O_C | | 5.36 | 4.77 | 4.87 | 4.94 | 5.13 |

Implementation of numerical methods becomes increasingly challenging when dealing with VIEs over long integration intervals. To demonstrate the efficiency of the CBRQM in such cases, consider the nonlinear VIE

$$\int_0^t \sin(t - sy(s)) \, ds = 1 - \cos t, \quad t \in [0, 100], \quad (16)$$

where the exact solution is $y(t) = 1$. The numerical results for this equation with parameters $(n, d, d_s) = (10, 3, 2)$ and $(n, d, d_s) = (12, 5, 4)$ are given in Table 4 and confirm once more the theoretical results and the capability of the method in solving VIEs over long intervals.

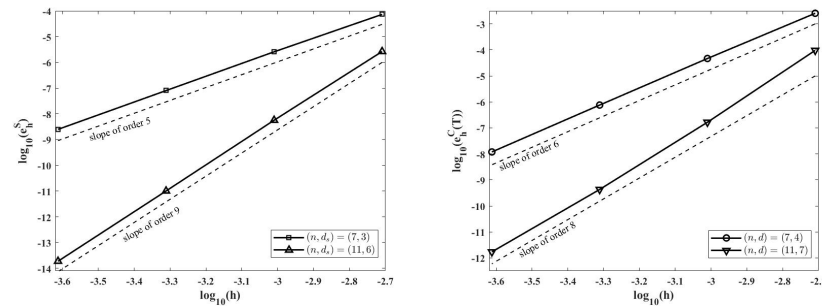
Finally, consider the highly oscillatory first-kind VIE

$$\int_0^t e^{-\alpha(t-s)} \cos(\omega(t-s))y(s) \, ds = g(t), \quad t \in [0, 1], \quad (17)$$

where the function g is chosen such that the exact solution is $y(t) = e^{-\alpha t} \sin(\omega t)$. For $\alpha > 0$ and $\omega \gg 1$, the VIE in (17) becomes highly oscillatory due to the cosine term. Figure 2 shows the logarithmic errors $\log_{10}(e_h^S)$ and $\log_{10}(e_h^C(T))$, corresponding to the starting procedure and the CBRQM (10), applied to the VIE in (17) for $\alpha = 1$ and $\omega = 100$, plotted versus

Table 4: Numerical results of the CBRQM applied to the VIE in (16).

| h | | 2^{-2} | 2^{-3} | 2^{-4} | 2^{-5} | 2^{-6} | 2^{-7} |
|----------------------------|------------|----------|----------|----------|----------|----------|----------|
| $(n, d, d_s) = (10, 3, 2)$ | | | | | | | |
| Starting procedure | e_h^S | 2.20e-4 | 2.18e-5 | 1.51e-6 | 9.67e-8 | 6.08e-9 | 3.80e-10 |
| | O_S | | 3.34 | 3.85 | 3.96 | 3.99 | 4.00 |
| CBRQM | $e_h^C(T)$ | 7.36e-5 | 1.48e-5 | 9.03e-7 | 3.32e-8 | 1.08e-9 | 3.41e-11 |
| | O_C | | 2.31 | 4.03 | 4.77 | 4.94 | 4.99 |
| $(n, d, d_s) = (12, 5, 4)$ | | | | | | | |
| Starting procedure | e_h^S | 8.01e-6 | 2.26e-7 | 4.01e-9 | 6.46e-11 | 1.02e-12 | 1.55e-14 |
| | O_S | | 5.15 | 5.82 | 5.96 | 5.98 | 6.04 |
| CBRQM | $e_h^C(T)$ | 4.89e-6 | 1.01e-7 | 2.34e-9 | 2.44e-11 | 2.12e-13 | 1.55e-15 |
| | O_C | | 5.60 | 5.43 | 6.58 | 6.85 | 7.10 |

Figure 2: Log-log-plots of the approximation error of the starting procedure and the CBRQM applied to the VIE in (17) with $(n, d_s) = (7, 3), (11, 6)$ (left) and $(n, d) = (7, 4), (11, 7)$ (right).

$\log_{10}(h)$, together with the expected convergence rates. As expected, the experimental orders of convergence for both the starting procedure and the CBRQM (10) exhibit excellent agreement with the theoretical results.

References

- [1] Abdi, A., Arnold, M. and Podhaisky, H. *The barycentric rational numerical differentiation formulas for stiff ODEs and DAEs*, Numer. Algorithms 97 (2024), 431–451.
- [2] Abdi, A., Berrut, J.-P. and Podhaisky, H. *The barycentric rational predictor–corrector schemes for Volterra integral equations*, J. Comput.

- Appl. Math. 440 (2024), 115611.
- [3] Abdi, A. and Hosseini, S.A. *The barycentric rational difference-quadrature scheme for systems of Volterra integro-differential equations*, SIAM J. Sci. Comput. 40 (2018), A1936–A1960.
- [4] Abdi, A., Hosseini, S.A. and Podhaisky, H. *The linear barycentric rational backward differentiation formulae for stiff ODEs on nonuniform grids*, Numer. Algorithms 98 (2025), 877–902.
- [5] Berrut, J.-P. *Rational functions for guaranteed and experimentally well-conditioned global interpolation*, Comput. Math. Appl. 15 (1998), 1–16.
- [6] Berrut, J.-P., Hosseini, S.A. and Klein, G. *The linear barycentric rational quadrature method for Volterra integral equations*, SIAM J. Sci. Comput. 36 (2014), A105–A123.
- [7] Bos, L., De Marchi, S. and Hormann, K. *On the Lebesgue constant of Berrut’s rational interpolant at equidistant nodes*, J. Comput. Appl. Math. 236 (2011), 504–510.
- [8] Bos, L., De Marchi, S., Hormann, K. and Klein, G. *On the Lebesgue constant of barycentric rational interpolation at equidistant nodes*, Numer. Math. 121 (2012), 461–471.
- [9] Brunner, H. *Collocation Methods for Volterra Integral and Related Functional Equations*, Cambridge University Press, Cambridge, 2004.
- [10] Brunner, H. and van der Houwen, P.J. *The Numerical Solution of Volterra Equations*, in: CWI Monogr., North-Holland, Amsterdam, 1986.
- [11] Floater, M.S. and Hormann, K. *Barycentric rational interpolation with no poles and high rates of approximation*, Numer. Math. 107 (2007), 315–331.
- [12] Fornberg, B. and Reeger, J.A. *An improved Gregory-like method for 1-D quadrature*, Numer. Math. 141 (2019), 1–19.

- [13] Gladwin, C.J. *Quadrature rule methods for Volterra integral equations of the first kind*, Math. Comput. 33 (1979), 705–716.
- [14] Hale, N. and Townsend, A. *Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weights*, SIAM J. Sci. Comput. 35 (2013), A652–A674.
- [15] Holyhead, P.A.W., McKee, S. and Taylor, P.J. *Multistep methods for solving linear Volterra integral equations of the first kind*, SIAM J. Numer. Anal. 12 (1975), 698–711.
- [16] Kauthen, J.-P. and Brunner, H. *Continuous collocation approximations to solution of first kind Volterra equations*, Math. Comput. 66 (1997), 1441–1459.
- [17] Klein, G. and Berrut, J.-P. *Linear barycentric rational quadrature*, BIT 52 (2012), 407–424.
- [18] Li, J. and Cheng, Y. *Linear barycentric rational collocation method for solving heat conduction equation*, Numer. Methods Partial Differ. Equ. 37 (2021), 533–545.
- [19] Li, J. and Cheng, Y. *Barycentric rational method for solving biharmonic equation by depression of order*, Numer. Methods Partial Differ. Equ. 37 (2021), 1993–2007.
- [20] Li, M. and Huang, C. *The linear barycentric rational quadrature method for auto-convolution Volterra integral equations*, J. Sci. Comput. 78 (2019), 549–564.
- [21] Linz, P. *Analytical and Numerical Methods for Volterra Equations*, SIAM, Philadelphia, 1985.
- [22] Liu, H., Huang J. and He, X. *Bivariate barycentric rational interpolation method for two dimensional fractional Volterra integral equations*, J. Comput. Appl. Math. 389 (2021), 113339.
- [23] Luo, W.-H., Huang, T.-Z., Gu, X.-M. and Liu, Y. *Barycentric rational collocation methods for a class of nonlinear parabolic partial differential equations*, Appl. Math. Lett. 68 (2017), 13–19.

- [24] Trefethen, L.N., et al. *Chebfun Version 5.6.0, The Chebfun Development Team*, <http://www.chebfun.org> (2016)
- [25] Trefethen, L.N. *Is Gauss quadrature better than Clenshaw–Curtis?* SIAM Rev. 50 (2008), 67–87.
- [26] Zhang, T. and Liang, H. *Multistep collocation approximations to solutions of first-kind Volterra integral equations*, Appl. Numer. Math. 130 (2018), 171–183.



Nonlinear optimization of revenue per unit of time in discrete Dutch auctions with risk-aware bidders

R.A. Shamim^{} and M.K. Majahar Ali*,^{}

Abstract

This study develops a computational framework to optimize the auctioneer's revenue per unit of time in modified discrete Dutch auction by incorporating bidders' risk preferences through the constant absolute risk aversion utility function. Bidders are categorized into three distinct risk profiles—risk-loving, risk-neutral, and risk-averse—allowing for a comprehensive analysis of how risk attitudes influence auction outcomes. A nonlinear programming methodology is utilized to ascertain the optimal revenue per unit time while incorporating discrete bid levels. The findings

*Corresponding author

Received 28 May 2025; revised 31 August 2025; accepted 13 September 2025

Raja Aqib Shamim

School of Mathematical Sciences, Universiti Sains Malaysia, 11800, Pulau Penang, Malaysia.

Department of Mathematics, University of Kotli, 11100, Azad Jammu and Kashmir, Pakistan. e-mail: raja.aqib5@student.usm.my

Majid Khan Majahar Ali

School of Mathematical Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia. e-mail: majidkhanmajaharali@usm.my

How to cite this article

Shamim, R.A. and Majahar Ali, M.K., Nonlinear optimization of revenue per unit of time in discrete Dutch auctions with risk-aware bidders. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1607-1638. <https://doi.org/10.22067/ijnao.2025.93750.1656>

demonstrate that, at the outset, an increase in the number of bidders substantially boosts the revenue per unit time; nevertheless, after reaching a specific point, the incremental benefits decrease, resulting in a plateau. Additionally, the analysis suggests that, in auctions featuring larger pools of bidders, achieving maximum revenue per unit time necessitates fewer bid levels, as surplus bid levels do not yield further revenue improvements. Bidders exhibiting risk-averse tendencies tend to generate lower returns due to their cautious bidding patterns, whereas risk-seeking participants contribute to higher revenue per unit time by engaging in more assertive bidding. Collectively, these results highlight the significant influence of bidders' risk preferences on auction design and establish a comprehensive mathematical framework that can be readily adapted to various algorithmic auction mechanisms. Behavioral interpretation via the prospect theory and alignment with published field evidence support the model's external validity.

AMS subject classifications (2020): Primary 90C30; Secondary 91B26.

Keywords: Auctions; Constant absolute risk aversion; Discrete Dutch auction; Nonlinear programming; Revenue per unit of time.

1 Introduction

An auction constitutes a competitive bidding mechanism wherein an item of uncertain value is awarded to the participant prepared to offer the highest price. Auctions represent one of the three primary methods of trade, alongside fixed-price sales and negotiation-based transactions [39]. They hold significant importance in the contemporary global economy, enabling the exchange of assets ranging from real estate and agricultural commodities to mineral rights and spectrum licenses [22, 56, 29, 12]. Among the diverse auction types, the Dutch auction (DA), also referred to as the descending-price or clock auction, stands out for its swift transaction process and particular suitability for the sale of perishable goods and time-sensitive assets [20, 48].

In a DA, the auctioneer initiates the process by setting an initially high asking price, which is then systematically reduced following a specified schedule until a participant agrees to the prevailing price [48]. Unlike ascending-

price English auctions that favor unique items, such as antiques, DAs excel in markets for goods with diminishing value over time, such as fresh produce, concert tickets, and container space [1, 32]. Practical applications extend to cash management [2], stock repurchases [6], and airline overbooking [23].

Traditional auction models often assume continuous bidding and risk-neutral bidders, which may not align with real-world dynamics [43, 46, 40]. The introduction of discrete Dutch auction (DDA) has addressed practical constraints by limiting bid levels to a discrete set of values. Early studies by Li and Kuo [32, 33] explored revenue maximization of DDA through optimal bid level design, demonstrating that revenue increases with the number of bid levels and bidders. However, these models ignored the variability in bidder risk preferences and emotional attachments. Li, Yue, and Kuo [34] extended DDA models by incorporating time as a critical parameter, examining trade-offs between auction duration and revenue. Their findings revealed that optimizing revenue per unit of time could significantly enhance auctioneer profitability, particularly in high-frequency auction environments. Despite these advancements, their models also remained limited to emotional attachment of the bidders with the item to be sold and bidders' risk preferences. Addressing some of these limitations, Shamim and Ali [48] integrated bidders' emotional attachments using the log-normal valuation distribution along with the consideration of time in DDA frameworks. By accounting for the emotional attachments, their research demonstrated the significant impact of emotions on auction outcomes and bidding strategies. However, they did not discuss the impact of bidders' risk preferences on the auction outcomes.

This study builds upon the aforementioned foundational works by examining the influence of bidders' risk preferences through the constant absolute risk aversion (CARA) utility function while incorporating the critical role of time in auction profitability. Recognizing that an auctioneer seeks to maximize revenue not only per auction but also per unit of time, this research integrates risk-sensitive bidder behavior with time-optimized revenue strategies. By formulating a computational framework that captures the complexities of real-world auctions, this study aims to enhance auction theory and offer practical insights for designing more efficient DDAs.

This study also extends the standard DDA model by explicitly addressing bidder risk asymmetry, clarifying the behavioral interpretation of the CARA parameter, and situating the framework alongside the prospect theory. While this study relies on simulation-based analysis, it also demonstrates that our results are consistent with published empirical studies of fish and flower markets, thereby reinforcing the practical relevance of its findings even in the absence of new transaction-level data.

The rest of this paper is structured as follows. Section 2 provides a comprehensive review of existing literature, highlighting the research gap addressed in this study. In Section 3, a mathematical revenue model for DDAs is developed, incorporating bidders' risk preferences and time considerations. Section 4 presents and analyzes the key results obtained by solving the proposed model using the R software. Finally, Section 5 concludes the study by summarizing the findings, discussing its limitations, and suggesting potential directions for future research.

2 Literature review

Auction theory has been a central theme in economic research, with considerable emphasis placed on analyzing the dynamics of different auction formats, such as English auctions, sealed-bid auctions, and DAs. Notably, DAs, distinguished by their descending-price structure, have become particularly valued for their effectiveness in facilitating the sale of perishable and time-sensitive goods [20, 48]. Nevertheless, much of the existing scholarship presumes continuous bidding and risk-neutral behavior among participants, assumptions that often do not align with the practical realities of auction environments [43, 46, 41]. This section provides an overview of the current literature on DAs, focusing specifically on discrete bidding and strategies for maximizing auctioneer revenue per unit of time, while also drawing attention to the insufficient consideration of bidders' risk preferences within the field.

Traditional auction models often assume that bid prices are continuous variables, allowing bidders to outbid each other by infinitesimally small increments [43, 46, 40, 14, 47]. While this assumption is suitable for unique items such as antiques, it is less applicable to fast DAs, where perishable goods

or services are sold rapidly. For instance, Royal Flora Holland auctions last approximately four seconds per transaction [28], and fish markets in Italy complete 15 transactions per minute using simultaneous clocks [17, 19]. The inefficiency of continuous bidding in such contexts has led researchers to explore discrete bidding mechanisms, where bid levels are restricted to a finite set of values.

Discrete bidding is not uncommon in English auctions [14, 47], sealed-bid auctions [10, 37], and hybrid auctions [25]. However, its application in DAs has received limited attention. Early work by Yu [54] demonstrated the existence of a symmetric pure-strategy equilibrium in DAs with fixed bid decrements but did not explore the optimization of bid levels or their impact on closing prices. Yuen, Sung, and Wong [55] extended this research by analyzing DAs conducted via wireless networks, introducing a communication cost factor. While their iterative numerical approach provided insights into optimal bid decrements, their model was constrained by its focus on communication costs and did not address revenue maximization directly.

The optimization of auctioneer revenue has been a central theme in auction theory. Cramton et al. [13] and Sujarittanonta [49] examined DAs with discrete bid levels, focusing on efficiency maximization rather than revenue optimization. In contrast, Li and Kuo [32, 33] explored revenue-maximizing DAs with unequal bid decrements, demonstrating that revenue increases with the number of bid levels and bidders. The assumption of a deterministic number of bidders was challenged by McAfee and McMillan [38], who introduced probabilistic models to account for uncertain bidder participation. This line of research gained traction with the rise of e-commerce, as online auctions necessitated models that could accommodate fluctuating bidder arrivals. Studies by Bajari and Hortaçsu [5], Etzion, Pinker, and Seidmann [15], and Caldentey and Vulcano [9] approximated bidder arrivals using Poisson processes, a modeling approach validated by empirical studies [50, 24]. Despite these advancements, the focus remained on English and sealed-bid auctions, leaving only a limited number of studies in DAs [48, 33, 34].

A significant limitation of traditional auction models is their neglect of time as a critical factor in auction profitability. While increasing the number of bid levels can enhance revenue per auction, it also prolongs auction

duration, potentially reducing the total number of transactions conducted within a given timeframe. Li, Yue, and Kuo [34] and Shamim and Ali [48] tackled this challenge by integrating time considerations into DDA models, illustrating that maximizing revenue per unit of time can substantially improve the auctioneer's profitability. Their research highlighted the critical need to balance the number of bid levels with the overall auction duration, especially within high-frequency auction settings.

Although previous studies have advanced the optimization of auction design, they have largely neglected the influence of bidders' risk preferences and emotional attachments. Conventional models typically assume bidders to be risk-neutral for the sake of analytical tractability; however, this assumption does not necessarily reflect the diversity of risk attitudes observed in actual auction settings. Shamim and Ali [48] contributed to this area by incorporating bidders' emotional attachments into DDA frameworks through the use of lognormal valuation distributions. Their work demonstrated the considerable effect of emotions on both auction outcomes and bidding behavior. Nevertheless, their approach did not consider the risk preferences of bidders, thereby leaving a significant gap in the existing body of literature.

This research extends the foundational contributions of Li and Kuo [32, 33], Li, Yue, and Kuo [34], and Shamim and Ali [48] by addressing two significant gaps identified in the current literature. First, it incorporates bidders' risk preferences-encompassing risk-neutral, risk-seeking, and risk-averse behaviors-within the DDA framework through the application of the CARA utility function. Second, it treats time as a pivotal parameter, with the objective of maximizing the auctioneer's revenue both per auction and per unit of time. Through the development of a computational framework that reflects the intricacies of real-world auction environments, this study aims to advance auction theory and offer actionable guidance for the design of more effective DAs. This article reports simulation-based evidence; due to the unavailability of public transaction-level DA data, empirical alignment is provided through published field studies.

In conclusion, although substantial advancements have been achieved in the study of DDAs and revenue optimization, the incorporation of bidders' risk preferences and strategies for maximizing time-sensitive revenue has not

been thoroughly investigated. This research seeks to address these deficiencies by presenting a comprehensive framework designed to improve both the efficiency and profitability of auctions.

3 Model development

This research investigates the impact of bidders' risk preferences on the revenue per unit of time in a DA featuring discrete bidding increments, within an independent private value (IPV) framework characterized by symmetric information. Under this setting, each bidder possesses knowledge solely of their own valuation for the auctioned item, which is independently drawn from a uniform distribution, and this information remains private and uninfluenced by the valuations of other participants [40, 34, 30]. The study considers scenarios in which bidders exhibit risk aversion, risk neutrality, or risk-loving behavior. In each case, a bidder is expected to place a bid when the asking price first drops to or below their valuation.

The discrete bid levels taken in this setting are $b_1 < b_2 < \dots < b_m$, where $m \geq 1$. Initially, the auctioneer opens the bidding process at a very high bid level b_{m+1} where nobody is willing to bid, and then the price decreases to $b_m, b_{m-1}, \dots, b_2, b_1$ after each preset interval of time until a bidder bids to buy the item at bid level b_i for any $i \in \{1, 2, \dots, m\}$. In the DA setting, the item is sold at a price b_i if and only if there exist q number of bidders having their valuations in the interval $[b_i, b_{i+1})$ and nobody is willing to buy it for the price higher than b_{i+1} . Also, the remaining $n - q$ bidders' valuations lie below b_i , $i = 1, 2, \dots, m$. If only one bidder has the valuation in the interval $[b_i, b_{i+1})$, then the object is sold to him/her and if there are two or more such bidders, the one who stops the clock first or calls out "mine" first will get the item.

If $n \geq 2$ participants take part in the auction, then the probability that the item is sold at the price level b_i , $i = 1, 2, \dots, m$ is $P(b_i)$, which is given by [32, 34]:

$$\begin{aligned}
 P(b_i) &= \sum_{q=1}^n \binom{n}{q} F(b_i)^{n-q} [F(b_{i+1}) - F(b_i)]^q, \\
 &= F(b_{i+1})^n - F(b_i)^n.
 \end{aligned} \tag{1}$$

To account for the risk preferences of the bidders, whether they are risk-loving, risk-neutral, or risk-averse, their utility of accepting a bid at the price level b_i is represented using the CARA utility function $U(b_i) = \frac{1-e^{-\alpha b_i}}{\alpha}$, where α is the constant of absolute risk aversion [30, 3, 42, 35, 8]. Therefore, the expected revenue per unit of time by the auctioneer in a DDA considering the risk preferences is given by

$$\mathcal{R} = \frac{\sum_{i=1}^m U(b_i) P(b_i)}{\mathcal{D}}, \tag{2}$$

where \mathcal{D} is the auction duration given as follows:

$$\begin{aligned}
 \mathcal{D} &= sE(m), \\
 &= s \left[\sum_{i=1}^m (m+2-i)P(b_i) + (m+1) \left(1 - \sum_{i=1}^m P(b_i) \right) \right], \\
 &= s \left[\sum_{i=1}^m (m+2-i) [F(b_{i+1})^n - F(b_i)^n] + (m+1) \left(1 - \sum_{i=1}^m [F(b_{i+1})^n - F(b_i)^n] \right) \right], \\
 &= s \left[(1+m)(1 + F(b_1)^n - F(b_{m+1})^n) + \sum_{i=1}^m (2-i+m)(F(b_{i+1})^n - F(b_i)^n) \right].
 \end{aligned} \tag{3}$$

In light of (1) and (3), (2) becomes

$$\mathcal{R} = \frac{\sum_{i=1}^m \frac{1-e^{-\alpha b_i}}{\alpha} [F(b_{i+1})^n - F(b_i)^n]}{s \left[(1+m)(1 + F(b_1)^n - F(b_{m+1})^n) + \sum_{i=1}^m (2-i+m)(F(b_{i+1})^n - F(b_i)^n) \right]}, \tag{4}$$

where α is the coefficient of constant absolute risk aversion, determining each bidder's risk attitude.

Here, a symmetric IPV setting is assumed; that is, the valuation of each bidder j is v_j , $j = 1, 2, \dots, n$, which is drawn from a uniform distribution defined on $[0, \bar{v}]$ with cumulative distribution function (c.d.f.) $F(\cdot)$ and probability distribution function (p.d.f.) $f(\cdot)$. In other words, all bidders share the same valuation distribution and private information. We also consider a DDA with a fixed number of price drop levels. The bid levels

$b_1 < b_2 < \dots < b_m$ span from a minimum price $b_1 = 0$ (no reserve price) to a maximum $b_{m+1} = \bar{v}$ (starting price) partitioning the $[0, \bar{v}]$ valuation range. It follows that $F(b_1) = 0$, $F(b_{m+1}) = \bar{v}$ and $F(b_i) = \frac{b_i}{\bar{v}}$, $i = 1, 2, \dots, m$ without any loss of generality. These modeling assumptions, consistent with prior literature [33, 36, 44], provide a tractable framework for our analysis. Hence, the seller's expected revenue per unit of time \mathcal{Z} can be expressed as follows:

$$\begin{aligned} \mathcal{Z} &= \frac{\sum_{i=1}^m \frac{1-e^{-\alpha b_i}}{\alpha} \left[\left(\frac{b_{i+1}}{\bar{v}} \right)^n - \left(\frac{b_i}{\bar{v}} \right)^n \right]}{s \left[(1+m) \left(1 + \left(\frac{b_1}{\bar{v}} \right)^n - \left(\frac{b_{m+1}}{\bar{v}} \right)^n \right) + \sum_{i=1}^m (2-i+m) \left(\left(\frac{b_{i+1}}{\bar{v}} \right)^n - \left(\frac{b_i}{\bar{v}} \right)^n \right) \right]}, \\ &= \frac{\sum_{i=1}^m (1-e^{-\alpha b_i}) (b_{i+1}^n - b_i^n)}{\alpha s \left[(1+m)(\bar{v}^n + b_1^n - b_{m+1}^n) + \sum_{i=1}^m (2-i+m)(b_{i+1}^n - b_i^n) \right]}. \end{aligned} \quad (5)$$

Therefore, the formulated model as a nonlinear program (NLP) in decision variables b_1, b_2, \dots, b_m and the parameters α , m , n , s , and \bar{v} is given below:

Maximize

$$\mathcal{Z} = \frac{\sum_{i=1}^m (1-e^{-\alpha b_i}) (b_{i+1}^n - b_i^n)}{\alpha s \left[(1+m)(\bar{v}^n + b_1^n - b_{m+1}^n) + \sum_{i=1}^m (2-i+m)(b_{i+1}^n - b_i^n) \right]},$$

subject to:

$$\begin{aligned} b_{i+1} &\geq b_i, \quad i = 1, 2, \dots, m, \\ b_1 &\geq 0, \\ b_{m+1} &= \bar{v}. \end{aligned} \quad (6)$$

In the above NLP (6), it is crucial to recognize that as α approaches 0, it signifies the risk-neutral case. This is due to the fact that $\lim_{\alpha \rightarrow 0} \frac{1-e^{-\alpha b_i}}{\alpha} = b_i$, which leads to the reduction of our NLP (6) to the model described by Li and Kuo [32], which does not account for the risk preferences of the bidders despite their claim, as that model lacks any parameters to define risk behaviors. Moreover, positive α indicates risk-averse bidders and negative α indicates risk-seeking behavior of the bidders [30, 3, 8].

This paper focuses on solving the optimization model (6) to find the revenue-maximizing set of bid levels and optimal revenue per unit of time under the given constraints. In mathematical terms, we tackle an NLP with m decision variables b_1, b_2, \dots, b_m (the bid levels). The objective function $\mathcal{Z}(b_1, \dots, b_m)$ is continuously differentiable but nonlinear and generally nonconvex, due to the combination of exponential utility terms and polynomial probability terms in (6). However, the structure of the problem offers some advantages: The feasible region is defined by simple linear inequalities $0 \leq b_1 \leq b_2 \leq \dots \leq b_m \leq \bar{v}$, and we observed that increasing a bid level beyond its optimal point yields diminishing returns (suggesting a single prominent optimum in practice). This NLP is solved using a numerical optimization approach. Specifically, a program in R (using the `nloptr` package) is implemented to maximize (6) subject to the constraints. This solver employs an augmented Lagrangian method to handle the monotonicity constraints effectively, ensuring that the solution respects $b_1 \leq \dots \leq b_m$. Each function evaluation of \mathcal{Z} involves summing over m terms and computing probabilities raised to the power n , which is an $O(m)$ computation. Thus, the computational complexity scales primarily with the number of bid levels m . In our study, we considered m up to 7, for which the solver finds solutions within a few hours on a standard PC. The number of bidders n influences the shape of the objective (larger n makes the revenue curve steeper) but does not increase the number of decision variables, so it has a minor impact on computation time. We also note that our model reduces to the known risk-neutral case when $\alpha \rightarrow 0$, for which analytical solution methods exist (see Li and Kuo [32]); but for arbitrary α , an analytical solution is intractable, validating our choice of a numerical solver. The use of a modern NLP solver is sufficient and efficient for the problem sizes in this study.

3.1 Risk preference asymmetry and utility curvature

We now formalize the distinction between risk-averse and risk-seeking bidders in our model. In the CARA utility framework $U(b_i) = \frac{1-e^{-\alpha b_i}}{\alpha}$, the sign of α governs the utility function's curvature and thereby the bidder's risk attitude. If $\alpha > 0$, then the second derivative $U''(b_i) < 0$, meaning $U(b_i)$ is concave, that is, the hallmark of risk aversion. The bidder derives diminishing marginal utility from monetary gains, preferring certain outcomes over gambles with the same expectation. Conversely, if $\alpha < 0$, then $U''(b_i) > 0$, making the utility convex. This corresponds to risk-seeking (risk-loving) behavior, where the bidder is inclined to gamble for higher returns, as the marginal utility of payoff increases with b_i . The boundary case $\alpha \rightarrow 0$ yields $U(b_i) = b_i$ (by L'Hopital's rule), a linear utility indicating risk neutrality. Thus, $\alpha \rightarrow 0$ is the cutoff point between two qualitatively different regimes of bidder behavior. We emphasize that positive α and negative α are not symmetric cases, rather they produce fundamentally different bidding incentives. A risk-averse bidder (positive α) is primarily concerned with avoiding high payments (losses), whereas a risk-loving bidder (negative α) focuses on the potential for paying very low prices (gains), even at the risk of possibly leaving empty-handed.

This asymmetry manifests in bidding strategies. A risk-averse bidder will tend to bid (stop the clock) earlier, at a higher price, to secure the item before the price falls too low and uncertainty increases. Their concave utility implies a high disutility for the "loss" incurred if the auction is lost or if the price drops further and someone else wins, hence they exit the auction sooner to minimize regret. In contrast, a risk-loving bidder gains extra utility from pushing their luck; the convex utility means the incremental utility of a lower price is high. Such a bidder is more willing to wait until the price has dropped significantly before bidding, even though waiting carries the risk of losing to a competitor. They effectively treat the prospect of getting a very cheap price as a gamble worth taking. Our model captures these tendencies via the parameter α . For example, in our simulations, a moderately risk-averse bidder ($\alpha = 0.2$) might stop the auction at a price around 80% of their private value, whereas a similarly strong risk-seeker ($\alpha = -0.2$) might hold

out until the price is 50–60% of their value, dramatically increasing variance in outcomes. Indeed, our computational results confirm this: Holding other parameters fixed, higher α leads to earlier bids and lower revenue, while more negative α leads to prolonged bidding and can raise revenue (see Tables 2 and 3). Formally, for any given number of bidders n and bid levels m , we find $\mathcal{Z}_{rl} > \mathcal{Z}_{rn} > \mathcal{Z}_{ra}$, where \mathcal{Z}_{rl} , \mathcal{Z}_{rn} , and \mathcal{Z}_{ra} represent the auctioneer's expected revenue per unit of time for risk-loving, risk-neutral, and risk-averse bidders, respectively, underscoring how the auctioneer's expected revenue per unit time improves as bidders become more risk-seeking (refer to Tables 1–3). This is intuitive from the model: cautious bidders “quit” early, yielding higher prices but fewer active bidders at low price levels, whereas risk-seeking bidders stay longer in the game, driving the price lower and intensifying competition, which paradoxically can increase the auctioneer's time-adjusted revenue by shortening auction duration. The key point is that risk aversion versus risk seeking are asymmetrical in effect, they do not simply cancel out or mirror each other. The analysis and results of this study reflect the asymmetry clearly.

4 Results and discussion

In this section, a series of problem instances is examined to analyze the behavior of the proposed model under varying parameter configurations. The number of bid levels is represented as $m \in \{2, 3, \dots, 7\}$, the number of bidders as $n \in \{2, 5, 10, 15, 20, 25, 30, 40, 60, 80, 100\}$, and the risk parameter as $\alpha \in \{-0.5, -0.4, \dots, 0.5\}$ with $\bar{v} = 1$ and $s = 1$ based on [34]. Using (6), NLPs are formulated and solved for different combinations of m , n , and α for $\bar{v} = 1$ and $s = 1$ by implementing a program in RStudio. In the subsequent discussion, $\mathcal{Z}_{m=\gamma}^*$ is used to denote the auctioneer's maximum expected revenue per unit of time when the number of bid levels is γ .

To facilitate further discussion, Table 1 provides a summary of the auctioneer's expected revenues per unit of time for all values of m specified above, under the assumption of risk-neutral bidders ($\alpha \rightarrow 0$). When bidders exhibit risk-neutral behavior ($\alpha \rightarrow 0$), the proposed model (6) reduces to the revenue model outlined by Li, Yue, and Kuo [34] for cases with zero salvage

value. The results presented in Table 1 align with those reported by Li, Yue, and Kuo [34] when identical parameter values are used.

Table 1: Auctioneer's maximum expected revenue per unit of time for risk-neutral bidders (i.e., $\alpha \rightarrow 0$) for $\bar{v} = 1$, $s = 1$, $n \in \{2, 5, 10, 15, 20, 25, 30, 40, 60, 80, 100\}$ and $m \in \{2, 3, \dots, 7\}$.

| n | $\mathcal{Z}_{m=2}^*$ | $\mathcal{Z}_{m=3}^*$ | $\mathcal{Z}_{m=4}^*$ | $\mathcal{Z}_{m=5}^*$ | $\mathcal{Z}_{m=6}^*$ | $\mathcal{Z}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1671 | 0.1900 | 0.1932 | 0.1934 | 0.1934 | 0.1934 |
| 5 | 0.2717 | 0.2983 | 0.3016 | 0.3019 | 0.3019 | 0.3019 |
| 10 | 0.3445 | 0.3671 | 0.3691 | 0.3693 | 0.3693 | 0.3693 |
| 15 | 0.3798 | 0.3986 | 0.3999 | 0.4000 | 0.4000 | 0.4000 |
| 20 | 0.4011 | 0.4170 | 0.4180 | 0.4180 | 0.4180 | 0.4180 |
| 25 | 0.4155 | 0.4293 | 0.4301 | 0.4301 | 0.4301 | 0.4301 |
| 30 | 0.4259 | 0.4382 | 0.4387 | 0.4387 | 0.4387 | 0.4387 |
| 40 | 0.4402 | 0.4501 | 0.4505 | 0.4505 | 0.4505 | 0.4505 |
| 60 | 0.4562 | 0.4634 | 0.4636 | 0.4636 | 0.4636 | 0.4636 |
| 80 | 0.4651 | 0.4708 | 0.4709 | 0.4709 | 0.4709 | 0.4709 |
| 100 | 0.4708 | 0.4755 | 0.4756 | 0.4756 | 0.4756 | 0.4756 |

As shown in Table 1, the expected revenue per unit of time consistently increases with the number of bidders n for each value of bid levels m ranging from 2 to 7. Similarly, for any given n , the expected revenue per unit of time rises with an increasing number of bid levels m . However, this growth halts when m reaches 4 for $n \geq 20$ or 5 for $n < 20$. The highest optimal values for expected revenue per unit of time for each n are highlighted in bold in Table 1. This finding aligns with previously established results in the literature [32, 33, 34, 55]. For instance, Figure 1 illustrates the case where $n = 40$, showing that the optimum revenue per unit of time \mathcal{Z}^* increases with m and reaches a peak value of 0.4505 when m is 4. These results suggest that the auctioneer can achieve maximum expected revenue per unit of time with no more than 5 bid levels, regardless of the value of n , consistent with the trends reported by Li, Yue, and Kuo [34].

It is important to note that this plateauing behavior occurs under our model's assumptions of symmetric bidders and uniform valuations. Intuitively, an auction's revenue per unit time cannot increase indefinitely with more competition, there is an upper limit. As n becomes very large, the highest bidder's valuation will likely be very close to the maximum value \bar{v} . (For instance, with valuations Uniform $[0, \bar{v}]$, the expected highest valuation

among n bidders is $\frac{n}{n+1}\bar{v}$, which approaches \bar{v} as $n \rightarrow \infty$.) This means adding more bidders beyond a certain point yields diminishing returns in expected revenue, causing the revenue curve to flatten out. Similarly, increasing the number of bid levels beyond about four or five yields negligible benefit because the auction outcome is then approaching that of a continuous DA. Additional intermediate price drops (bid levels) past this threshold do not significantly raise the winning price or reduce the selling time. Therefore, our model indicates that under standard conditions (uniform i.i.d. values and no reserve price) the optimal auction design need not exceed five price levels, a result that aligns with economic intuition and prior findings in the literature.

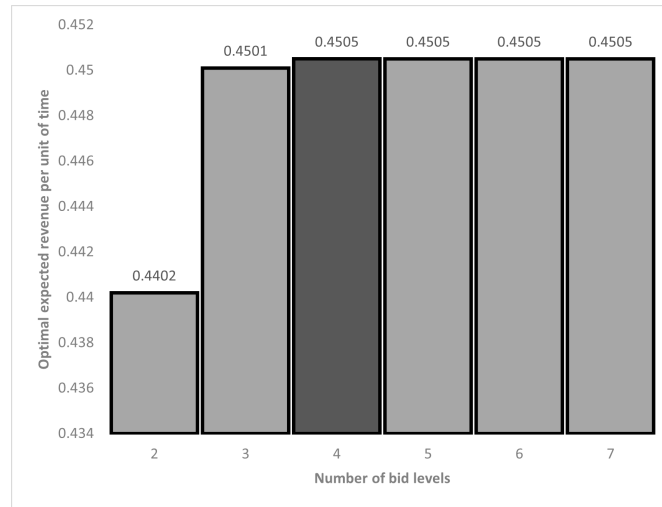


Figure 1: The auctioneer's maximum expected revenue per unit of time (\mathcal{Z}^*) versus number of bid levels m for $n = 40$ and risk-neutral bidders, that is, ($\alpha \rightarrow 0$).

Table 2 outlines the auctioneer's maximum expected revenue per unit of time across various bid levels (m) and numbers of bidders (n) under the condition of risk-averse bidders, characterized by $\alpha \in \{0.1, 0.2, \dots, 0.5\}$. A higher value of α indicates greater risk aversion [45, 11, 4]. The results demonstrate that as α increases, reflecting heightened risk aversion, the expected revenue per unit of time declines for every combination of m and n . This occurs because higher risk aversion leads bidders to adopt less aggressive bidding strategies, opting to wait for lower prices to mitigate potential

losses. Consequently, the auctioneer's maximum expected revenue decreases with increasing α .

When comparing the results from Table 1 (risk-neutral bidders) and Table 2 (risk-averse bidders), it is evident that optimal expected revenue is higher under risk-neutral conditions. Risk-averse bidders, prioritizing loss minimization over potential gains, tend to bid conservatively, which negatively impacts the auctioneer's revenue [45, 53]. This strategic shift underscores how risk preferences influence auction dynamics, leading to lower bids and reduced revenue for the auctioneer [52, 7].

Furthermore, Tables 2a–2e highlight that for larger bidder groups, fewer bid levels are required to maximize the auctioneer's expected revenue per unit of time. Specifically, the optimal number of bid levels is typically fewer than five, dropping to four or even three in certain cases when the number of bidders is sufficiently high. For instance, in the scenario where $n = 100$ and $\alpha = 0.1$, four bid levels are sufficient to achieve maximum expected revenue. However, for the same bidding population and $\alpha = 0.5$, only three bid levels are required.

Table 3 highlights the auctioneer's maximum expected revenue per unit of time across various bid levels (m) and numbers of bidders (n) under the influence of risk-loving (or risk-seeking) bidders. Specifically, this analysis considers $\alpha \in \{-0.1, -0.2, \dots, -0.5\}$, where more negative values of α indicate stronger risk-seeking behavior [45, 11, 4]. The table reveals that as α decreases (becomes more negative), reflecting heightened risk-seeking tendencies, the expected revenue per unit of time for the auctioneer increases consistently for all values of m and n . This behavior stems from the aggressive bidding strategies of risk-loving participants, who avoid delaying their bids for potential price drops, driven by their preference for higher risks. As a result, the auctioneer's maximum expected revenue increases as risk-seeking behavior intensifies.

A comparison between Table 1 (risk-neutral bidders) and Table 3 (risk-loving bidders) shows that the optimal revenue per unit of time generated from risk-neutral bidders is lower than that from risk-loving bidders. The propensity of risk-loving bidders to take bold risks results in more aggressive bidding behavior, which translates to higher maximum expected revenue per

Table 2: Auctioneer's maximum expected revenue per unit of time for risk-averse bidders (i.e., $\alpha \in \{0.1, 0.2, \dots, 0.5\}$) for $\bar{v} = 1$, $s = 1$, $n \in \{2, 5, 10, 15, 20, 25, 30, 40, 60, 80, 100\}$ and $m \in \{2, 3, \dots, 7\}$.

(a) For $\alpha = 0.1$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1628 | 0.1848 | 0.1879 | 0.1881 | 0.1881 | 0.1881 |
| 5 | 0.2629 | 0.2882 | 0.2912 | 0.2915 | 0.2916 | 0.2916 |
| 10 | 0.3318 | 0.3529 | 0.3548 | 0.3549 | 0.3549 | 0.3549 |
| 15 | 0.3648 | 0.3823 | 0.3835 | 0.3836 | 0.3836 | 0.3836 |
| 20 | 0.3846 | 0.3994 | 0.4003 | 0.4004 | 0.4004 | 0.4004 |
| 25 | 0.398 | 0.4108 | 0.4115 | 0.4115 | 0.4115 | 0.4115 |
| 30 | 0.4077 | 0.4190 | 0.4195 | 0.4195 | 0.4195 | 0.4195 |
| 40 | 0.4209 | 0.4300 | 0.4304 | 0.4304 | 0.4304 | 0.4304 |
| 60 | 0.4357 | 0.4423 | 0.4425 | 0.4425 | 0.4425 | 0.4425 |
| 80 | 0.4439 | 0.4491 | 0.4492 | 0.4492 | 0.4492 | 0.4492 |
| 100 | 0.4491 | 0.4534 | 0.4535 | 0.4535 | 0.4535 | 0.4535 |

(b) For $\alpha = 0.2$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1587 | 0.1799 | 0.1828 | 0.1830 | 0.1830 | 0.1830 |
| 5 | 0.2546 | 0.2786 | 0.2814 | 0.2817 | 0.2817 | 0.2817 |
| 10 | 0.3197 | 0.3395 | 0.3412 | 0.3413 | 0.3413 | 0.3413 |
| 15 | 0.3507 | 0.3669 | 0.3680 | 0.3681 | 0.3681 | 0.3681 |
| 20 | 0.3691 | 0.3829 | 0.3836 | 0.3837 | 0.3837 | 0.3837 |
| 25 | 0.3816 | 0.3934 | 0.3940 | 0.3940 | 0.3940 | 0.3940 |
| 30 | 0.3906 | 0.4010 | 0.4014 | 0.4014 | 0.4014 | 0.4014 |
| 40 | 0.4028 | 0.4112 | 0.4115 | 0.4115 | 0.4115 | 0.4115 |
| 60 | 0.4164 | 0.4225 | 0.4226 | 0.4226 | 0.4226 | 0.4226 |
| 80 | 0.4239 | 0.4287 | 0.4288 | 0.4288 | 0.4288 | 0.4288 |
| 100 | 0.4288 | 0.4327 | 0.4327 | 0.4327 | 0.4327 | 0.4327 |

(c) For $\alpha = 0.3$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1548 | 0.1752 | 0.1780 | 0.1782 | 0.1782 | 0.1782 |
| 5 | 0.2467 | 0.2694 | 0.2721 | 0.2724 | 0.2724 | 0.2724 |
| 10 | 0.3082 | 0.3268 | 0.3284 | 0.3285 | 0.3285 | 0.3285 |
| 15 | 0.3373 | 0.3524 | 0.3534 | 0.3535 | 0.3535 | 0.3535 |
| 20 | 0.3545 | 0.3672 | 0.3679 | 0.3680 | 0.3680 | 0.3680 |
| 25 | 0.3661 | 0.3770 | 0.3775 | 0.3775 | 0.3775 | 0.3775 |
| 30 | 0.3744 | 0.3840 | 0.3844 | 0.3844 | 0.3844 | 0.3844 |
| 40 | 0.3857 | 0.3934 | 0.3937 | 0.3937 | 0.3937 | 0.3937 |
| 60 | 0.3983 | 0.4038 | 0.4040 | 0.4040 | 0.4040 | 0.4040 |
| 80 | 0.4052 | 0.4096 | 0.4096 | 0.4096 | 0.4096 | 0.4096 |
| 100 | 0.4097 | 0.4132 | 0.4133 | 0.4133 | 0.4133 | 0.4133 |

(d) For $\alpha = 0.4$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1511 | 0.1706 | 0.1733 | 0.1735 | 0.1735 | 0.1735 |
| 5 | 0.2392 | 0.2607 | 0.2632 | 0.2635 | 0.2635 | 0.2635 |
| 10 | 0.2974 | 0.3147 | 0.3162 | 0.3163 | 0.3163 | 0.3163 |
| 15 | 0.3246 | 0.3387 | 0.3396 | 0.3396 | 0.3397 | 0.3397 |
| 20 | 0.3407 | 0.3525 | 0.3531 | 0.3531 | 0.3531 | 0.3531 |
| 25 | 0.3514 | 0.3615 | 0.3620 | 0.3620 | 0.3620 | 0.3620 |
| 30 | 0.3591 | 0.3680 | 0.3684 | 0.3684 | 0.3684 | 0.3684 |
| 40 | 0.3696 | 0.3767 | 0.3769 | 0.3769 | 0.3769 | 0.3769 |
| 60 | 0.3812 | 0.3863 | 0.3864 | 0.3864 | 0.3864 | 0.3864 |
| 80 | 0.3876 | 0.3916 | 0.3916 | 0.3916 | 0.3916 | 0.3916 |
| 100 | 0.3917 | 0.3949 | 0.3950 | 0.3950 | 0.3950 | 0.3950 |

(e) For $\alpha = 0.5$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1475 | 0.1663 | 0.1689 | 0.1691 | 0.1691 | 0.1691 |
| 5 | 0.232 | 0.2524 | 0.2548 | 0.2550 | 0.2550 | 0.2550 |
| 10 | 0.2871 | 0.3034 | 0.3047 | 0.3048 | 0.3048 | 0.3048 |
| 15 | 0.3126 | 0.3257 | 0.3265 | 0.3266 | 0.3266 | 0.3266 |
| 20 | 0.3276 | 0.3385 | 0.3391 | 0.3391 | 0.3391 | 0.3391 |
| 25 | 0.3376 | 0.3469 | 0.3474 | 0.3474 | 0.3474 | 0.3474 |
| 30 | 0.3448 | 0.3529 | 0.3533 | 0.3533 | 0.3533 | 0.3533 |
| 40 | 0.3544 | 0.3610 | 0.3612 | 0.3612 | 0.3612 | 0.3612 |
| 60 | 0.3652 | 0.3698 | 0.3699 | 0.3699 | 0.3699 | 0.3699 |
| 80 | 0.371 | 0.3747 | 0.3747 | 0.3747 | 0.3747 | 0.3747 |
| 100 | 0.3748 | 0.3778 | 0.3778 | 0.3778 | 0.3778 | 0.3778 |

unit of time for the auctioneer compared to their risk-neutral counterparts [45, 53]. In essence, risk-loving bidders focus on maximizing potential gains rather than minimizing losses. This leads to higher bids, directly enhancing the auctioneer's expected revenue per unit of time [52, 7].

Furthermore, Tables 3a–3e emphasize that as the number of bidders (n) increases, the number of bid levels required to maximize the auctioneer's expected revenue per unit of time generally decreases. The maximum number of bid levels required remains five or fewer in most cases, although it can reach six bid levels when n is relatively small. For instance, when $n = 100$, only four bid levels are sufficient to maximize expected revenue per unit of time, irrespective of the value of α .

Table 3: Auctioneer's maximum expected revenue per unit of time for risk-loving bidders (i.e., $\alpha \in \{-0.1, -0.2, \dots, -0.5\}$) for $\bar{v} = 1$, $s = 1$, $n \in \{2, 5, 10, 15, 20, 25, 30, 40, 60, 80, 100\}$ and $m \in \{2, 3, \dots, 7\}$.

(a) For $\alpha = 0.1$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1716 | 0.1954 | 0.1987 | 0.1990 | 0.1990 | 0.1990 |
| 5 | 0.2808 | 0.309 | 0.3124 | 0.3128 | 0.3128 | 0.3128 |
| 10 | 0.358 | 0.3820 | 0.3843 | 0.3844 | 0.3844 | 0.3844 |
| 15 | 0.3956 | 0.4158 | 0.4173 | 0.4174 | 0.4174 | 0.4174 |
| 20 | 0.4185 | 0.4357 | 0.4368 | 0.4368 | 0.4368 | 0.4368 |
| 25 | 0.434 | 0.4490 | 0.4498 | 0.4498 | 0.4498 | 0.4498 |
| 30 | 0.4452 | 0.4585 | 0.4591 | 0.4592 | 0.4592 | 0.4592 |
| 40 | 0.4606 | 0.4715 | 0.4719 | 0.4719 | 0.4719 | 0.4719 |
| 60 | 0.478 | 0.4859 | 0.4861 | 0.4861 | 0.4861 | 0.4861 |
| 80 | 0.4877 | 0.4939 | 0.4941 | 0.4941 | 0.4941 | 0.4941 |
| 100 | 0.4939 | 0.4991 | 0.4992 | 0.4992 | 0.4992 | 0.4992 |

(b) For $\alpha = 0.2$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1763 | 0.201 | 0.2045 | 0.2048 | 0.2048 | 0.2048 |
| 5 | 0.2905 | 0.3202 | 0.3239 | 0.3242 | 0.3243 | 0.3243 |
| 10 | 0.3722 | 0.3979 | 0.4003 | 0.4005 | 0.4005 | 0.4005 |
| 15 | 0.4124 | 0.4341 | 0.4357 | 0.4358 | 0.4359 | 0.4359 |
| 20 | 0.4369 | 0.4555 | 0.4567 | 0.4568 | 0.4568 | 0.4568 |
| 25 | 0.4536 | 0.4699 | 0.4707 | 0.4708 | 0.4708 | 0.4708 |
| 30 | 0.4658 | 0.4802 | 0.4809 | 0.4809 | 0.4809 | 0.4809 |
| 40 | 0.4824 | 0.4942 | 0.4947 | 0.4947 | 0.4947 | 0.4947 |
| 60 | 0.5013 | 0.5099 | 0.5102 | 0.5102 | 0.5102 | 0.5102 |
| 80 | 0.5118 | 0.5186 | 0.5188 | 0.5188 | 0.5188 | 0.5188 |
| 100 | 0.5186 | 0.5243 | 0.5244 | 0.5244 | 0.5244 | 0.5244 |

(c) For $\alpha = 0.3$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1812 | 0.2070 | 0.2106 | 0.2109 | 0.2109 | 0.2109 |
| 5 | 0.3007 | 0.3320 | 0.3359 | 0.3363 | 0.3364 | 0.3364 |
| 10 | 0.3872 | 0.4147 | 0.4173 | 0.4175 | 0.4175 | 0.4175 |
| 15 | 0.4302 | 0.4535 | 0.4553 | 0.4554 | 0.4554 | 0.4554 |
| 20 | 0.4565 | 0.4766 | 0.4779 | 0.4780 | 0.4780 | 0.4780 |
| 25 | 0.4745 | 0.4921 | 0.4930 | 0.4931 | 0.4931 | 0.4931 |
| 30 | 0.4876 | 0.5032 | 0.5040 | 0.5041 | 0.5041 | 0.5041 |
| 40 | 0.5056 | 0.5185 | 0.5190 | 0.5190 | 0.5190 | 0.5190 |
| 60 | 0.5261 | 0.5355 | 0.5358 | 0.5358 | 0.5358 | 0.5358 |
| 80 | 0.5375 | 0.5450 | 0.5452 | 0.5452 | 0.5452 | 0.5452 |
| 100 | 0.5449 | 0.5511 | 0.5512 | 0.5512 | 0.5512 | 0.5512 |

(d) For $\alpha = 0.4$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1864 | 0.2132 | 0.2170 | 0.2173 | 0.2173 | 0.2173 |
| 5 | 0.3113 | 0.3445 | 0.3487 | 0.3491 | 0.3491 | 0.3491 |
| 10 | 0.4031 | 0.4324 | 0.4353 | 0.4355 | 0.4355 | 0.4355 |
| 15 | 0.4491 | 0.4741 | 0.4761 | 0.4762 | 0.4762 | 0.4762 |
| 20 | 0.4773 | 0.499 | 0.5004 | 0.5005 | 0.5005 | 0.5005 |
| 25 | 0.4966 | 0.5157 | 0.5168 | 0.5168 | 0.5168 | 0.5168 |
| 30 | 0.5108 | 0.5278 | 0.5287 | 0.5287 | 0.5287 | 0.5287 |
| 40 | 0.5303 | 0.5443 | 0.5449 | 0.5449 | 0.5449 | 0.5449 |
| 60 | 0.5525 | 0.5628 | 0.5631 | 0.5631 | 0.5631 | 0.5631 |
| 80 | 0.565 | 0.5731 | 0.5733 | 0.5733 | 0.5733 | 0.5733 |
| 100 | 0.573 | 0.5798 | 0.5799 | 0.5799 | 0.5799 | 0.5799 |

(e) For $\alpha = 0.5$

| n | $\mathcal{R}_{m=2}^*$ | $\mathcal{R}_{m=3}^*$ | $\mathcal{R}_{m=4}^*$ | $\mathcal{R}_{m=5}^*$ | $\mathcal{R}_{m=6}^*$ | $\mathcal{R}_{m=7}^*$ |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 2 | 0.1918 | 0.2197 | 0.2237 | 0.2240 | 0.2240 | 0.2240 |
| 5 | 0.3226 | 0.3576 | 0.3621 | 0.3625 | 0.3626 | 0.3626 |
| 10 | 0.4199 | 0.4513 | 0.4543 | 0.4546 | 0.4546 | 0.4546 |
| 15 | 0.4691 | 0.496 | 0.4981 | 0.4983 | 0.4983 | 0.4983 |
| 20 | 0.4994 | 0.5228 | 0.5244 | 0.5245 | 0.5245 | 0.5245 |
| 25 | 0.5202 | 0.5408 | 0.5421 | 0.5421 | 0.5421 | 0.5421 |
| 30 | 0.5355 | 0.5539 | 0.5549 | 0.5550 | 0.5550 | 0.5550 |
| 40 | 0.5566 | 0.5718 | 0.5725 | 0.5725 | 0.5725 | 0.5725 |
| 60 | 0.5807 | 0.5920 | 0.5923 | 0.5923 | 0.5923 | 0.5923 |
| 80 | 0.5942 | 0.6032 | 0.6034 | 0.6034 | 0.6034 | 0.6034 |
| 100 | 0.603 | 0.6105 | 0.6106 | 0.6106 | 0.6106 | 0.6106 |

From Tables 1–3, it is evident that the inequality $\mathcal{R}_l > \mathcal{R}_{rn} > \mathcal{R}_a$ consistently holds for all values of m and n . Here, \mathcal{R}_l , \mathcal{R}_{rn} , and \mathcal{R}_a represent the auctioneer's expected revenue per unit of time for risk-loving, risk-neutral, and risk-averse bidders, respectively. This relationship is further illustrated for $m = 5$ in Figure 2, which shows that the expected revenue per unit of time $\mathcal{R}_{m=5}^*$ increases steadily as α decreases, thereby corroborating the stated inequality.

Moreover, Figure 2 reveals that as the number of bidders n increases, the revenue initially grows rapidly, but the rate of growth gradually slows down beyond a certain point for higher values of n . A similar trend can be observed for other values of m , indicating a consistent pattern across the

auction's various configurations. The findings indicate that an increase in the number of bidders leads to a higher revenue per unit of time, as expected. However, beyond a certain threshold, adding more bidders has a diminishing impact on auction outcomes. This is due to the fact that while additional bidders contribute to increased competition, the marginal revenue gains become negligible. Moreover, excessively increasing the number of bidders results in significantly higher operational costs, including administrative expenses and auction management overhead, which may offset the benefits of increased participation.

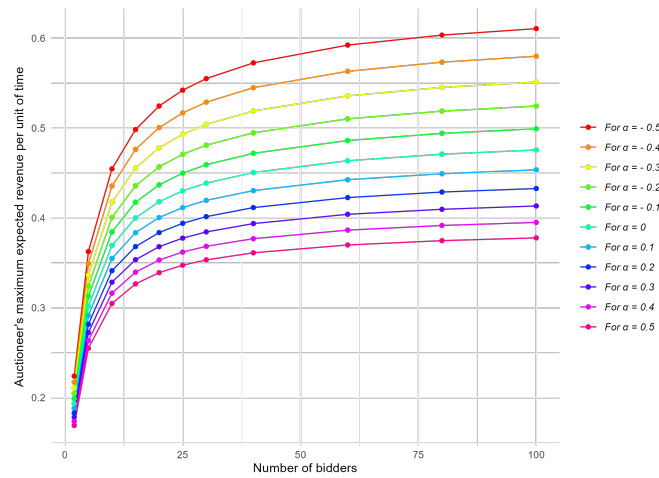


Figure 2: The auctioneer's maximum expected revenue per unit of time ($\mathcal{R}_{m=5}^*$) versus number of bidders (n) where $m = 5$, $s = 1$, $\bar{v} = 1$ and $\alpha \in \{-0.5, 0.4, \dots, 0.5\}$.

Tables 4, 5, and 6 present the optimal bid levels for $m = 6$ with $\bar{v} = 1$, considering risk-neutral ($\alpha \rightarrow 0$), risk-averse ($\alpha > 0$), and risk-loving ($\alpha < 0$) bidders, respectively. In all cases, $b_1 = 0$ implies that the lowest bid level is zero, meaning that the item is given away for free if unsold by that point (as in [32]), an assumption in our developed model. Additionally, $b_{m+1} = 1$ represents the highest asking price, with all intermediate bid levels optimized using the NLP (6).

Figure 3 illustrates the relationship between the constant of absolute risk aversion α and the optimal bid levels b_i from Tables 4–6. Specifically, Figures 3a and 3b represent $n = 10$ and $n = 30$, showing that for a smaller number

of bidders, the auctioneer must set distinct bid levels to maximize expected revenue per unit of time for each value of α . Conversely, Figures 3c and 3d demonstrate that as the number of bidders increases, the gap between the bid level curves b_i^* decreases, allowing the auctioneer to skip several intermediate bid levels while still maximizing the expected revenue per unit of time. These graphs confirm that fewer bid levels suffice to maximize expected revenue per unit of time as the number of bidders grows significantly.

Although the optimal solutions of the NLP (6) for Li and Kuo's parameters [32] are not explicitly presented here, replicating their conditions with $\alpha \rightarrow 0$ validates our model against their findings. This validation underscores the robustness of our approach, which extends the existing literature by incorporating the impact of bidders' risk preferences on the auctioneer's expected revenue per unit of time in DDAs—a previously unexplored aspect.

Table 4: Risk-neutral (i.e., $\alpha \rightarrow 0$) optimal bid levels for $m = 6$, $\bar{v} = 1$, $s = 1$ and $n \in \{2, 5, 10, 15, 20, 25, 30, 40, 60, 80, 100\}$.

| n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 |
|-----|-------|---------|---------|---------|---------|---------|
| 2 | 0 | 0.00637 | 0.05086 | 0.16367 | 0.3564 | 0.63423 |
| 5 | 0 | 0.12656 | 0.22543 | 0.35649 | 0.52718 | 0.74089 |
| 10 | 0 | 0.29469 | 0.40417 | 0.52172 | 0.65895 | 0.81797 |
| 15 | 0 | 0.4047 | 0.50874 | 0.61167 | 0.72696 | 0.85617 |
| 20 | 0 | 0.43077 | 0.5754 | 0.66937 | 0.76961 | 0.87961 |
| 25 | 0 | 0.13506 | 0.61354 | 0.70967 | 0.79927 | 0.89571 |
| 30 | 0 | 0.03807 | 0.65014 | 0.74045 | 0.82134 | 0.90756 |
| 40 | 0 | 0.0032 | 0.4345 | 0.77603 | 0.85195 | 0.92399 |
| 60 | 0 | 0.00037 | 0.08696 | 0.82408 | 0.88799 | 0.94292 |
| 80 | 0 | 0.01177 | 0.0637 | 0.8562 | 0.90889 | 0.95373 |
| 100 | 0 | 0.02018 | 0.03763 | 0.87752 | 0.92267 | 0.96081 |

Beyond the numerical results, bidders' risk attitudes are shaped by behavioral factors. To capture this dimension, we complement the quantitative analysis with insights from the prospect theory (Section 4.1) and further relate our findings to published field evidence (Section 4.2).

Table 5: Risk-averse (i.e., $\alpha \in \{0.1, 0.2, \dots, 0.5\}$) optimal bid levels for $m = 6$, $\bar{v} = 1$, $s = 1$, $n \in \{2, 5, 10, 15, 20, 25, 30, 40, 60, 80, 100\}$.

| (a) For $\alpha = 0.1$ | | | | | | | (b) For $\alpha = 0.2$ | | | | | | |
|------------------------|-------|----------|----------|----------|----------|----------|------------------------|-------|----------|----------|----------|----------|----------|
| n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 | n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 |
| 2 | 0 | 0.006369 | 0.05019 | 0.160943 | 0.351108 | 0.628404 | 2 | 0 | 0.006361 | 0.049527 | 0.15829 | 0.34593 | 0.622612 |
| 5 | 0 | 0.125464 | 0.223157 | 0.352824 | 0.522604 | 0.736984 | 5 | 0 | 0.124371 | 0.220902 | 0.349199 | 0.518056 | 0.733072 |
| 10 | 0 | 0.292227 | 0.401038 | 0.518059 | 0.655308 | 0.815349 | 10 | 0 | 0.289806 | 0.397913 | 0.5144 | 0.651656 | 0.812708 |
| 15 | 0 | 0.401718 | 0.505522 | 0.608329 | 0.723948 | 0.854164 | 15 | 0 | 0.398847 | 0.502304 | 0.604973 | 0.720908 | 0.852139 |
| 20 | 0 | 0.217861 | 0.562987 | 0.665763 | 0.766988 | 0.87798 | 20 | 0 | 0.242129 | 0.560997 | 0.662805 | 0.764391 | 0.876328 |
| 25 | 0 | 0.089728 | 0.609424 | 0.706885 | 0.79701 | 0.894332 | 25 | 0 | 0.110955 | 0.607069 | 0.704176 | 0.794732 | 0.892931 |
| 30 | 0 | 0.031427 | 0.647208 | 0.737953 | 0.819334 | 0.906362 | 30 | 0 | 0.049844 | 0.644846 | 0.735459 | 0.817304 | 0.905144 |
| 40 | 0 | 0.007614 | 0.457394 | 0.774618 | 0.850336 | 0.923038 | 40 | 0 | 0.008437 | 0.478437 | 0.773193 | 0.848701 | 0.92207 |
| 60 | 0 | 0.041936 | 0.125849 | 0.822922 | 0.886796 | 0.942243 | 60 | 0 | 0.013095 | 0.100278 | 0.820966 | 0.88556 | 0.941551 |
| 80 | 0 | 0.007262 | 0.062786 | 0.854846 | 0.90793 | 0.953198 | 80 | 0 | 0.000198 | 0.020311 | 0.853103 | 0.906942 | 0.952658 |
| 100 | 0 | 0.006374 | 0.053867 | 0.87651 | 0.921869 | 0.960376 | 100 | 0 | 0.002268 | 0.037323 | 0.875249 | 0.921051 | 0.959933 |

| (c) For $\alpha = 0.3$ | | | | | | | (d) For $\alpha = 0.4$ | | | | | | |
|------------------------|-------|----------|----------|----------|----------|----------|------------------------|-------|----------|----------|----------|----------|----------|
| n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 | n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 |
| 2 | 0 | 0.006352 | 0.048875 | 0.155706 | 0.340861 | 0.61686 | 2 | 0 | 0.00634 | 0.048234 | 0.153189 | 0.335902 | 0.611153 |
| 5 | 0 | 0.123276 | 0.218666 | 0.34561 | 0.513537 | 0.72915 | 5 | 0 | 0.122181 | 0.216451 | 0.342061 | 0.50905 | 0.725222 |
| 10 | 0 | 0.287384 | 0.394792 | 0.510742 | 0.647993 | 0.810045 | 10 | 0 | 0.284961 | 0.391676 | 0.507087 | 0.644322 | 0.807361 |
| 15 | 0 | 0.395954 | 0.499075 | 0.601603 | 0.717848 | 0.850091 | 15 | 0 | 0.39306 | 0.49584 | 0.598221 | 0.714767 | 0.848021 |
| 20 | 0 | 0.249816 | 0.558328 | 0.659782 | 0.761765 | 0.874654 | 20 | 0 | 0.21983 | 0.554073 | 0.656643 | 0.759109 | 0.872959 |
| 25 | 0 | 0.094469 | 0.603605 | 0.701375 | 0.792425 | 0.89151 | 25 | 0 | 0.074269 | 0.599807 | 0.698539 | 0.790095 | 0.890071 |
| 30 | 0 | 0.054418 | 0.642144 | 0.732927 | 0.815248 | 0.903908 | 30 | 0 | 0.087762 | 0.64014 | 0.730399 | 0.81317 | 0.902654 |
| 40 | 0 | 0.058763 | 0.5332 | 0.772527 | 0.847074 | 0.921089 | 40 | 0 | 0.032483 | 0.505141 | 0.769579 | 0.845334 | 0.920089 |
| 60 | 0 | 0.009706 | 0.11239 | 0.819471 | 0.884319 | 0.940849 | 60 | 0 | 0.012192 | 0.134555 | 0.818088 | 0.883061 | 0.940135 |
| 80 | 0 | 0.001578 | 0.03352 | 0.851851 | 0.90595 | 0.95211 | 80 | 0 | 0.006956 | 0.025687 | 0.850382 | 0.904939 | 0.951553 |
| 100 | 0 | 0.023409 | 0.043642 | 0.874137 | 0.920222 | 0.959483 | 100 | 0 | 0.031534 | 0.037506 | 0.872913 | 0.91938 | 0.959026 |

| (e) For $\alpha = 0.5$ | | | | | | |
|------------------------|-------|----------|----------|----------|----------|----------|
| n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 |
| 2 | 0 | 0.006326 | 0.047604 | 0.150739 | 0.33105 | 0.605494 |
| 5 | 0 | 0.121087 | 0.214256 | 0.33855 | 0.504596 | 0.721288 |
| 10 | 0 | 0.282548 | 0.388568 | 0.503437 | 0.640644 | 0.804655 |
| 15 | 0 | 0.390155 | 0.492596 | 0.594826 | 0.711667 | 0.845929 |
| 20 | 0 | 0.235182 | 0.551674 | 0.653599 | 0.756438 | 0.871244 |
| 25 | 0 | 0.145968 | 0.599328 | 0.695842 | 0.787747 | 0.888613 |
| 30 | 0 | 0.033733 | 0.635972 | 0.727758 | 0.811065 | 0.901384 |
| 40 | 0 | 0.005836 | 0.490001 | 0.767377 | 0.843606 | 0.919077 |
| 60 | 0 | 0.003413 | 0.122017 | 0.816238 | 0.881776 | 0.939411 |
| 80 | 0 | 0.016607 | 0.03418 | 0.849054 | 0.903916 | 0.950987 |
| 100 | 0 | 0.000002 | 0.000859 | 0.871422 | 0.918521 | 0.958561 |

4.1 Behavioral perspective: Prospect theory and risk behavior

Growing research in behavioral economics shows that individuals' risk preferences are reference-dependent. In particular, Kahneman and Tversky's prospect theory posits that people evaluate outcomes relative to a reference point (status quo) and exhibit risk aversion for gains and risk seeking for losses, rather than a uniform risk attitude [27]. This behavior is captured by an S-shaped value function (refer to Figure 4): Concave in the gains region (implying diminishing sensitivity and risk-averse behavior) but convex in the losses region, reflecting risk-seeking tendencies to avoid sure losses. In the context of auctions, this means a bidder's inclination to take risks

Table 6: Risk-loving (i.e., $\alpha \in \{-0.1, -0.2, \dots, -0.5\}$) optimal bid levels for $m = 6$, $\bar{v} = 1$, $s = 1$, $n \in \{2, 5, 10, 15, 20, 25, 30, 40, 60, 80, 100\}$.

| (a) For $\alpha = 0.1$ | | | | | | | (b) For $\alpha = 0.2$ | | | | | | |
|------------------------|-------|----------|----------|----------|----------|----------|------------------------|-------|----------|----------|----------|----------|----------|
| n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 | n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 |
| 2 | 0 | 0.006375 | 0.051546 | 0.166462 | 0.361795 | 0.640093 | 2 | 0 | 0.006373 | 0.052238 | 0.169329 | 0.367303 | 0.64598 |
| 5 | 0 | 0.127647 | 0.227719 | 0.360184 | 0.531781 | 0.744772 | 5 | 0 | 0.128736 | 0.230026 | 0.363916 | 0.536405 | 0.748642 |
| 10 | 0 | 0.29708 | 0.407293 | 0.525376 | 0.662573 | 0.820556 | 10 | 0 | 0.299522 | 0.410421 | 0.52903 | 0.666183 | 0.823122 |
| 15 | 0 | 0.407425 | 0.511922 | 0.614991 | 0.729956 | 0.858142 | 15 | 0 | 0.410409 | 0.515113 | 0.618296 | 0.732924 | 0.860094 |
| 20 | 0 | 0.482237 | 0.581117 | 0.672531 | 0.772176 | 0.881224 | 20 | 0 | 0.444641 | 0.581898 | 0.675348 | 0.77469 | 0.88281 |
| 25 | 0 | 0.088777 | 0.615174 | 0.712301 | 0.801493 | 0.897075 | 25 | 0 | 0.105909 | 0.618475 | 0.714982 | 0.803695 | 0.898416 |
| 30 | 0 | 0.063925 | 0.653424 | 0.742928 | 0.823323 | 0.908745 | 30 | 0 | 0.036579 | 0.655578 | 0.745332 | 0.825277 | 0.90991 |
| 40 | 0 | 0.038886 | 0.554843 | 0.781179 | 0.853688 | 0.924935 | 40 | 0 | 0.000879 | 0.438112 | 0.77999 | 0.855147 | 0.925853 |
| 60 | 0 | 0.012076 | 0.124245 | 0.826104 | 0.889195 | 0.943592 | 60 | 0 | 0.010616 | 0.098259 | 0.827382 | 0.890358 | 0.94425 |
| 80 | 0 | 4.26E-05 | 0.026447 | 0.85719 | 0.909836 | 0.95425 | 80 | 0 | 0.00337 | 0.054132 | 0.85871 | 0.910774 | 0.954763 |
| 100 | 0 | 0.030176 | 0.089937 | 0.878983 | 0.923466 | 0.96124 | 100 | 0 | 0.004399 | 0.005239 | 0.879533 | 0.924233 | 0.961661 |

| (c) For $\alpha = 0.3$ | | | | | | | (d) For $\alpha = 0.4$ | | | | | | |
|------------------------|-------|----------|----------|----------|----------|----------|------------------------|-------|----------|----------|----------|----------|----------|
| n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 | n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 |
| 2 | 0 | 0.006368 | 0.052938 | 0.172269 | 0.37292 | 0.651889 | 2 | 0 | 0.006358 | 0.053645 | 0.175284 | 0.378645 | 0.657814 |
| 5 | 0 | 0.129817 | 0.232346 | 0.36768 | 0.541051 | 0.752494 | 5 | 0 | 0.130889 | 0.234679 | 0.371476 | 0.545715 | 0.756325 |
| 10 | 0 | 0.301964 | 0.413547 | 0.532678 | 0.669776 | 0.82566 | 10 | 0 | 0.304386 | 0.416668 | 0.536319 | 0.673349 | 0.82817 |
| 15 | 0 | 0.413297 | 0.518282 | 0.621579 | 0.735865 | 0.862021 | 15 | 0 | 0.416156 | 0.521432 | 0.624841 | 0.738779 | 0.863921 |
| 20 | 0 | 0.437732 | 0.58437 | 0.678252 | 0.777185 | 0.884374 | 20 | 0 | 0.411012 | 0.585903 | 0.681063 | 0.779646 | 0.885914 |
| 25 | 0 | 0.104765 | 0.621276 | 0.717632 | 0.80587 | 0.899737 | 25 | 0 | 0.094328 | 0.623848 | 0.720226 | 0.808014 | 0.901037 |
| 30 | 0 | 0.082239 | 0.659187 | 0.747779 | 0.827208 | 0.911056 | 30 | 0 | 0.041685 | 0.66105 | 0.750112 | 0.829108 | 0.912183 |
| 40 | 0 | 0.014346 | 0.477744 | 0.782869 | 0.856752 | 0.926763 | 40 | 0 | 0.001515 | 0.477023 | 0.784702 | 0.85829 | 0.927656 |
| 60 | 0 | 0.013075 | 0.107012 | 0.829033 | 0.891514 | 0.944897 | 60 | 0 | 0.023113 | 0.116925 | 0.830673 | 0.892651 | 0.945533 |
| 80 | 0 | 0.000748 | 0.034652 | 0.859841 | 0.911689 | 0.955267 | 80 | 0 | 0.018905 | 0.072682 | 0.861382 | 0.912598 | 0.955763 |
| 100 | 0 | 0.001334 | 0.004129 | 0.880615 | 0.924997 | 0.962075 | 100 | 0 | 0.020446 | 0.04199 | 0.881898 | 0.925752 | 0.962482 |

| (e) For $\alpha = 0.5$ | | | | | | |
|------------------------|-------|----------|----------|----------|----------|----------|
| n | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 |
| 2 | 0 | 0.006345 | 0.054358 | 0.178373 | 0.384477 | 0.66375 |
| 5 | 0 | 0.131952 | 0.237024 | 0.375302 | 0.550396 | 0.760132 |
| 10 | 0 | 0.306795 | 0.419782 | 0.539951 | 0.676902 | 0.830652 |
| 15 | 0 | 0.419002 | 0.524563 | 0.62808 | 0.741664 | 0.865796 |
| 20 | 0 | 0.421518 | 0.589118 | 0.683961 | 0.782086 | 0.887431 |
| 25 | 0 | 0.128293 | 0.627457 | 0.722856 | 0.810134 | 0.902317 |
| 30 | 0 | 0.048893 | 0.663809 | 0.75246 | 0.830983 | 0.913292 |
| 40 | 0 | 0.001763 | 0.495012 | 0.787023 | 0.859826 | 0.928535 |
| 60 | 0 | 0.000512 | 0.087395 | 0.831879 | 0.893756 | 0.946157 |
| 80 | 0 | 0.005219 | 0.029883 | 0.862319 | 0.913477 | 0.956249 |
| 100 | 0 | 0.011548 | 0.026896 | 0.882862 | 0.926487 | 0.962881 |

may increase if they perceive themselves as “in the losses”—for example, if the current auction price exceeds their internal reference point (perhaps the price they initially hoped to pay). Conversely, if a bidder stands to obtain a item at a price well below their value (a perceived gain), then the prospect theory predicts more risk-averse behavior, that is, locking in the win rather than gambling further [31].

The Dutch auction format may interact with these behavioral tendencies. Bidders often set a mental reference price; dropping below it turns the potential purchase into a “gain” scenario, where they might become cautious and clinch the deal. If the price stays above that reference (a potential loss relative to their target), bidders might hold off (risk-averse) longer than standard risk-neutral models predict, hoping the price drops further. Empirical evi-

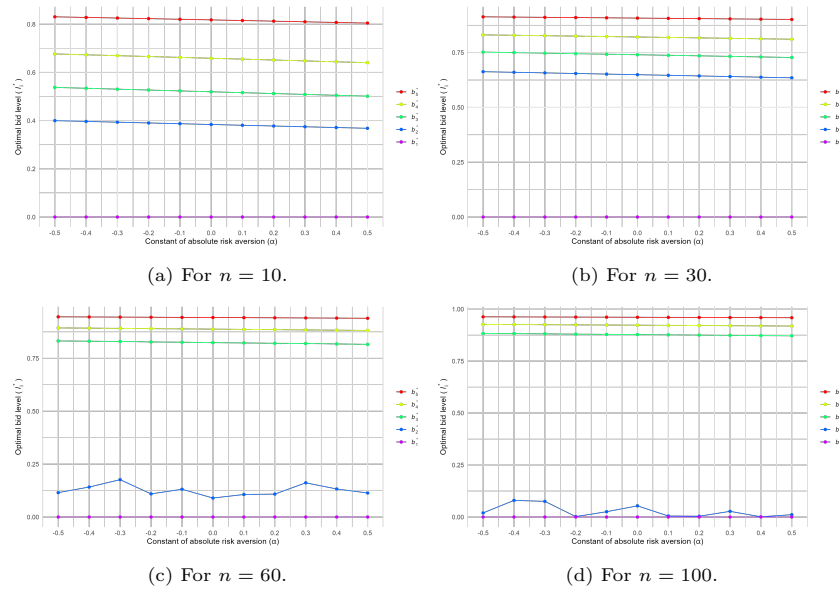


Figure 3: Constant of absolute risk aversion (α) versus optimal bid levels (b_i^*) when $m = 5$.

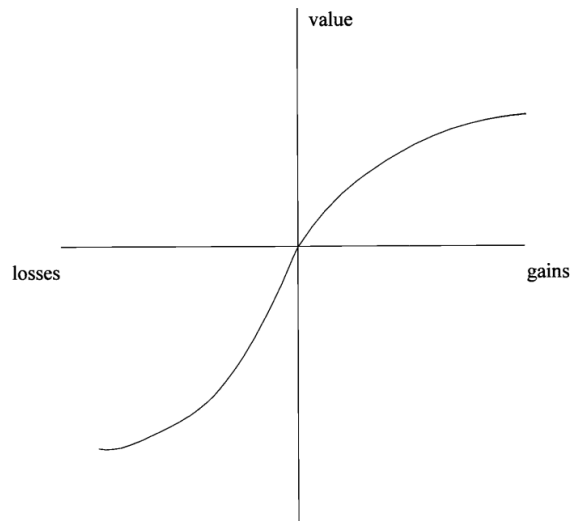


Figure 4: Value function (as in [31]).

dence of such behavior is noted in auction experiments and field data [18, 27]. For instance, experienced bidders sometimes “ride the clock” longer when

they feel they are “behind” (a form of loss-chasing), a behavior consistent with the prospect theory’s loss-domain risk seeking. This subsection bridges our model with real-world behavior: While our optimization assumes consistent risk preferences via CARA utilities, in practice a bidder’s risk posture might dynamically shift from conservative to bold depending on whether the current price is viewed as a gain or a loss. Incorporating such reference-dependent preferences formally is an interesting avenue for future extensions of our model.

4.2 Validation through published empirical evidence

Although this study is based on simulation results, it is important to consider whether the findings are consistent with empirical evidence. Access to detailed, transaction-level Dutch auction data remains highly restricted, since most fish and flower auction houses do not make such records publicly available. As a result, we validate our framework by drawing on published studies that have analyzed real auction data.

Fluvià et al. [16], using approximately 179,000 transactions from the Ancona fish market, reported the patterns that resonate strongly with our model: Prices decline substantially over the course of an auction, and auctioneer revenue per unit time reaches a plateau once bidder participation exceeds a certain threshold. Both results mirror the dynamics predicted in our simulations. Likewise, empirical evidence from the Dutch flower auctions (Royal FloraHolland) shows that transactions are typically concluded within seconds [51, 34], meaning that only a small number of bid decrements are actually employed. This observation supports our result that, beyond an optimal number of bid levels, additional increments contribute little to revenue performance. Taken together, these studies confirm that the main dynamics captured by our simulations are also observed in practice: (i) revenue per unit time increases with more bidders but eventually exhibits diminishing returns, and (ii) auction efficiency is achieved with only a limited number of bid decrements. While no new empirical dataset is introduced here, the con-

sistency between the findings of the study and published evidence provides external validation of this research.

Another central contribution of this work is the explicit integration of bidders' risk preferences, modeled with the CARA utility function, into a framework for revenue per unit of time optimization. The simulations clearly indicate that risk-seeking behavior enhances auctioneer revenue, whereas risk aversion reduces it. This asymmetry follows from the curvature of the utility function: concavity (risk aversion) leads to cautious bidding, while convexity (risk seeking) encourages prolonged participation and more aggressive bids. These insights are consistent with prior laboratory and field studies. Kagel and Levin [26] demonstrated experimentally that risk-averse bidders shade their bids, lowering seller revenue. Hu, Matthews, and Zou [21] showed both analytically and experimentally that risk-averse participants produce lower expected revenues in Dutch auctions, while risk-loving bidders generate more aggressive competition and higher revenues. Evidence from fish auctions aligns with this as well: Fluvà et al. [16] found that cautious buyer strategies depressed prices, whereas aggressive bidding accelerated sales at higher prices, reflecting the same outcomes as our risk-seeking simulations.

By aligning these CARA-based results with established experimental and field evidence, this study demonstrates that bidder risk attitudes are not only theoretically significant but also observable in real markets. This strengthens the external validity of the proposed framework, even in the absence of new transaction-level data.

5 Conclusion

This study introduced a novel framework for modeling the DDA using a non-linear programming approach to maximize the auctioneer's expected revenue per unit of time while explicitly accounting for bidders' risk preferences. By integrating the CARA utility function, the model extended existing research by incorporating α as a measure of bidders' risk attitudes. Results derived from extensive numerical experiments revealed several significant insights that enhance understanding of optimal auction design.

The findings demonstrated that the auctioneer's expected revenue per unit of time increases as the number of bidders n grows. Initially, the revenue per unit of time experiences a sharp rise with an increasing number of participants; however, as the number of bidders becomes sufficiently large, the rate of growth slows. This suggested diminishing returns in terms of revenue gains with further increases in bidder population. For auctions with smaller numbers of bidders, the auctioneer must set each bid level distinctly to achieve maximum revenue per unit of time. However, as the number of bidders increases, some bid levels can be omitted without affecting the optimal outcome. This observation implied that the auction design can be simplified for larger bidder populations without compromising efficiency of revenue per unit of time.

Furthermore, this study emphasized the influence of the number of bid levels (m) on the optimization of revenue per unit of time. The results indicated that, although an increase in bid levels initially enhances the auctioneer's maximum expected revenue per unit of time, this improvement eventually plateaus, suggesting that additional bid levels beyond a certain threshold do not yield further benefits. These observations substantiated that employing five or fewer bid levels is sufficient for optimizing revenue per unit of time, regardless of the risk preferences of the bidders.

The implications of these findings are significant for auctioneers seeking to forecast potential revenues per unit of time and enhance the efficiency of their auction operations. By taking into account both the number of participants and the optimal structuring of bid levels, auctioneers can effectively balance operational complexity with the goal of maximizing revenue per unit of time. This research not only corroborates previous studies but also deepens the understanding of the role that risk preferences play in shaping auction outcomes, thereby providing valuable, practical recommendations for the implementation of auctions in real-world settings.

As the risk aversion coefficient (α) increases, there is a corresponding decrease in the auctioneer's expected revenue per unit of time. This reduction is primarily due to the conservative bidding behavior exhibited by risk-averse participants, who prioritize minimizing potential losses over seeking additional gains. In contrast, when α assumes more negative values, reflecting

stronger risk-loving tendencies, the auctioneer's expected revenue per unit of time rises. This outcome is attributable to the assertive bidding strategies adopted by risk-loving bidders, who focus on maximizing potential gains rather than limiting losses. Throughout this study, it is consistently observed that $\mathcal{Z}_{rl} > \mathcal{Z}_{rn} > \mathcal{Z}_{ra}$, where \mathcal{Z}_{rl} , \mathcal{Z}_{rn} , and \mathcal{Z}_{ra} denote the auctioneer's expected revenue for risk-loving, risk-neutral, and risk-averse bidders, respectively. These findings underscored the significant impact of bidders' risk preferences on auction outcomes.

Although the model developed in this study advanced prior research by explicitly integrating risk preferences, it also demonstrated the capacity to reproduce earlier results when $\alpha \rightarrow 0$, thus affirming its validity and wider applicability. This dual functionality highlighted the model's robustness and adaptability, establishing it as a noteworthy contribution to the literature on DDA.

Beyond the numerical optimization results, this work underscored the fundamental role of risk attitudes in shaping auctioneer revenue per unit time. The consistent ordering of revenues across risk profiles, the behavioral justification provided by the prospect theory, and the alignment with empirical patterns reported in the literature together validated the robustness of the proposed CARA-based approach. While direct access to proprietary datasets remains limited, the convergence of our simulation outcomes with documented field evidence ensured that the framework not only advances theoretical auction design but also offers insights that are credible and transferable to real-world market settings.

The results of this study contributed to a deeper understanding of DDA and provided valuable guidance for enhancing auction design. Nevertheless, several limitations should be recognized. This research does not utilize real-world data for empirical validation, presumes a zero minimum selling price, and relies exclusively on the CARA utility function in conjunction with uniformly distributed bidder valuations. These simplifying assumptions suggested avenues for future inquiry. Subsequent studies could address these constraints by investigating the impact of nonzero minimum selling prices, employing alternative probability distributions for bidder valuations, and exploring different utility functions to model risk preferences. Furthermore,

empirical validation of the model, where feasible, would strengthen its practical significance and applicability.

In summary, this study enhances the comprehension of DDA by incorporating bidders' risk preferences into a computational optimization framework. Through the application of nonlinear programming methods to examine the effects of these preferences, the research advanced auction theory and illustrated the utility of mathematical computing in the formulation of effective auction mechanisms. The findings presented herein established a basis for the development of more efficient and practical auction models, with relevance extending across a variety of economic and computational contexts.

Acknowledgements

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

References

- [1] Adam, M.T. , Eidels, A., Lux, E. and Teubner, T. *Bidding behavior in dutch auctions: Insights from a structured literature review*, Int. J. Electron. Commer., 21 (2017) 363–397.
- [2] Alderson, M.J., Brown, K.C. and Lummer, S.L. *Dutch auction rate preferred stock*, Financ. Manag., 16 (1987) 68–73.
- [3] Alvarez, F. and Mazon, C. *Comparing the spanish and the discriminatory auction formats: A discrete model with private information*, Eur. J. Oper. Res., 179 (2007) 256–266.
- [4] Babcock, B.A., Choi, E.K. and Feinerman, E. *Risk and probability premiums for cara utility functions*, J. Agric. Resour. Econ., 18 (1993) 17–24.
- [5] Bajari, P. and Hortacsu, A. *The winner's curse, reserve prices, and endogenous entry: Empirical insights from ebay auctions*, RAND J. Econ., 34 (2003) 329–355.

- [6] Barger, L. and Farrell, M. *The price effect of stock repurchases: Evidence from dual class firms*, Manag. Sci., 67 (2021) 6568–6580.
- [7] Bos, O., Gomez-Martinez, F. and Onderstal, S. *Signalling in auctions for risk-averse bidders*, PLOS One, 17 (2022) p. e0275709.
- [8] Brunner, C., Hu, A. and Oechssler, J. *Premium auctions and risk preferences: An experimental study*, (Heidelberg Discussion Paper Series, No. 544). Heidelberg University, 2013.
- [9] Caldentey, R. and Vulcano, G. *Online auction and list price revenue management*, Manag. Sci., 53 (2007) 795–813.
- [10] Chwe, M.S.-Y. *The discrete bid first auction*, Econ. Lett., 31 (1989) 303–306.
- [11] Colell, A.M., Whinston, M. and Green, J. *Microeconomic theory*, Oxford University Press, New York, 1995.
- [12] Cramton, P. *Spectrum auction design*, Rev. Ind. Organ., 42 (2013) 161–190.
- [13] Cramton, P., Filiz-Ozbay E., Ozbay E.Y. and Sujarittanonta, P. *Discrete clock auctions: an experimental study*, Exp. Econ., 15 (2012) 309–322.
- [14] David, E., Rogers, A, Jennings, N.R. , Schiff, J., Kraus, S. and Rothkopf, M.H. *Optimal design of english auctions with discrete bid levels*, ACM Trans. Internet Technol., 7 (2007) 12–es.
- [15] Etzion, H., Pinker, E. and Seidmann, A. *Analyzing the simultaneous use of auctions and posted prices for online selling*, Manuf. Serv. Oper. Manag., 8 (2006) 68–91.
- [16] Fluvia, M., Garriga, A., Rigall-I-Torrent, R., Rodríguez-Carámbula, E. and Saló, A. *Buyer and seller behavior in fish markets organized as Dutch auctions: Evidence from a wholesale fish market in Southern Europe*, Fish. Res., 127 (2012) 18–25.
- [17] Gallegati, M., Giulioni, G., Kirman, A. and Palestini, A. *What's that got to do with the price of fish? buyers behavior on the ancona fish market*, J. Econ. Behav. Organ., 80 (2011) 20–33.

- [18] Giannikos, C.I., Kakolyris, A. and Suen, T.S. *Prospect theory and a manager's decision to trade a blind principal bid basket*, Glob. Finance J., 55 (2023) 100806.
- [19] Guerci, E., Kirman, A. and Moulet, S. *Learning to bid in sequential dutch auctions*, J. Econ. Dyn. Control, 48 (2014) 374–393.
- [20] Hafalir, I., Kesten, O., Sherstyuk, K. and Tao, C. *When speed is of essence: Perishable goods auctions*, University of Hawaii at Mānoa, Department of Economics, Hawaii, US, 2023.
- [21] Hu, A., Matthews, S.A. and Zou, L. *Risk aversion and optimal reserve prices in first-and second-price auctions*, J. Econ. Theory, 3 (2010) 1188–1202.
- [22] Hungria-Gunnelin, R. *An analysis of auction strategies in apartment sales*, J. Eur. Real Estate Res., 11 (2018) 202–223.
- [23] Ivaldi, M., Petrova, M. and Urdanoz, M. *Airline cooperation effects on airfare distribution: An auction-model-based approach*, Transp. Policy, 115 (2022) 239–250.
- [24] Jiang, A. X. and Leyton-Brown, K. *Estimating bidders' valuation distributions in online auctions*, in Proceedings of IJCAI-05 workshop on game theoretic and decision theoretic agents, (2005) 98–107.
- [25] Junmin, S. and Chang, A.-C. *Revenue and duration of oral auction*, Int. J. Ind. Eng. Manag., 2 (2009) 368–377.
- [26] Kagel, J.H. and Levin, D. *Auctions: A survey of experimental research*, In J.H. Kagel & A.E. Roth (Eds.), Handb. Exp. Econ. (Vol. 2, pp. 563–637). Princeton University Press, 2016.
- [27] Kahneman, D. and Tversky, A. *Prospect theory: An analysis of decision under risk*, In Handbook of the fundamentals of financial decision making: Part I, World Scientific, (2013) 99–127.
- [28] Kambil, A. and Van Heck, E. *Reengineering the dutch flower auctions: A framework for analyzing exchange organizations*, Inf. Sys.Res., 9 (1998) 1–19.

- [29] Klemperer, P. *Auctions: Theory and Practice*, Princeton University Press, United Kingdom, 2004.
- [30] Krishna, V. *Auction theory*, 2nd Ed., Academic press, Elsevier, USA, 2009.
- [31] Levy, J.S. *Applications of prospect theory to political science*, Synthese, 135 (2003) 215–241.
- [32] Li, Z. and Kuo, C.-C. *Revenue-maximizing dutch auctions with discrete bid levels*, Eur. J. Oper. Res., 215 (2011) 721–729.
- [33] Li, Z. and Kuo, C.-C. *Design of discrete dutch auctions with an uncertain number of bidders*, Ann. Oper. Res., 211 (2013) 255–272.
- [34] Li, Z., Yue, J. and Kuo, C.-C. *Design of discrete dutch auctions with consideration of time*, Eur. J. Oper. Res., 265 (2018) 1159–1171.
- [35] Makui, A., Naboureh, K. and Sadjadi, S.J. *An experimental study of auctions behavior with risk preferences*, Int. J. Ind. Eng. Manag. Sci., 8 (2021) 1–7.
- [36] Mathews, T. *Bidder welfare in an auction with a buyout option*, Int. Game Theory Rev., 8 (2006) 595–612.
- [37] Mathews, T. and Sengupta, A. *Sealed bid second price auctions with discrete bidding*, Appl. Econ. Res. Bull., 1 (2008) 31–52.
- [38] McAfee R.P. and McMillan, J. *Auctions with a stochastic number of bidders*, J. Econ. Theory, 43 (1987) 1–19.
- [39] Menezes, F.M. and Monteiro, P.K. *An introduction to auction theory*, Oxford University Press, United Kingdom, 2005.
- [40] Milgrom, P.R. *Putting auction theory to work*, Cambridge University Press, United Kingdom, 2004.
- [41] Milgrom, P.R. and Weber, R.J. *A theory of auctions and competitive bidding*, Econometrica, 50 (1982) 1089–1122.

- [42] Murto, P. and Valimaki, J. *Large common value auctions with risk-averse bidders*, Games Econ. Behav., 91 (2015) 60–74.
- [43] Myerson, R.B. *Optimal auction design*, Math. Oper. Res., 6 (1981) 58–73.
- [44] Pekec, A.S. and Tsetlin, I. *Revenue ranking of discriminatory and uniform auctions with an unknown number of bidders*, Manag. Sci., 54 (2008) 1610–1623.
- [45] Rabin, M. *Risk aversion and expected-utility theory: A calibration theorem*, in Handbook of the fundamentals of financial decision making: Part I, World Scientific (2013) 241–252.
- [46] Riley, J.G. and Samuelson, W.F. *Optimal auctions*, Am. Econ. Rev., 71 (1981) 381–392.
- [47] Rothkopf, M.H. and Harstad, R.M. *On the role of discrete bid levels in oral auctions*, Eur. J. Oper. Res., 74 (74) 572–581.
- [48] Shamim, R.A. and Ali, M.K.M. *Optimizing discrete dutch auctions with time considerations: a strategic approach for lognormal valuation distributions*, J. Niger. Soc. Phys. Sci. 7 (2025) 2291–2291.
- [49] Sujarittanonta, P. *Design of discrete auction*, University of Maryland, College Park, 2010.
- [50] Vakrat, Y. and Seidmann, A. *Implications of the bidders' arrival process on the design of online auctions*, in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, IEEE (2000) 7–pp.
- [51] Van den Berg, G.J., Van Ours, J.C. and Pradhan, M.P. *The declining price anomaly in Dutch Dutch rose auctions*, Am. Econ. Rev., 4,(2001) 1055–1062.
- [52] Vasserman, S. and Watt, M. *Risk aversion and auction design: Theoretical and empirical evidence*, Int. J. Ind. Organ., 79 (2021) 102758.

- [53] Yong, L. and Shulin, L. *Effects of risk aversion on all-pay auction with reimbursement*, Econ. Lett., 185 (2019) 108751.
- [54] Yu, J. *Discrete approximation of continuous allocation mechanisms*, California Institute of Technology, United States, 1999.
- [55] Yuen, W.H., Sung, C.W. and Wong, W.S. *Optimal price decremental strategy for dutch auctions*, Commun. Inf. Syst., 2 (2002) 411–434.
- [56] Zimik, C. and Mishra, P.P. *Auction game of agro-product under cob-web phenomenon of supply and demand*, Int. J. Bus. Forecast. Mark. Intell., 10 (2025) 119–139.



On overcoming Dahlquist's second barrier for A -stable linear multistep methods

G. Hojjati*, , S. Fazeli and A. Moradi

Abstract

Dahlquist's second barrier limits the order of A -stable linear multistep methods to at most two, posing significant challenges for achieving higher accuracy in the numerical solution of stiff ordinary differential equations. Leveraging various successful techniques, many efforts have been made to develop efficient methods that overcome this fundamental obstacle through different approaches. In this paper, we survey these techniques and analyze their impact on enhancing the stability and accuracy of the resulting methods. A comprehensive understanding of these advances can assist researchers in designing more effective algorithms for stiff problems.

*Corresponding author

Received 28 June 2025; revised 10 September 2025; accepted 13 September 2025

Gholamreza Hojjati

Faculty of Mathematics, Statistics and Computer Science, University of Tabriz, Tabriz, Iran. e-mail: ghojjati@tabrizu.ac.ir

Somayyeh Fazeli

Marand Technical College, University of Tabriz, Tabriz, Iran. e-mail: fazeli@tabrizu.ac.ir

Afsaneh Moradi

Institute of Analysis and Numerics, Otto von Guericke University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany. e-mail: afsaneh.moradi@ovgu.de

How to cite this article

Hojjati, G., Fazeli, S. and Moradi, A., On overcoming Dahlquist's second barrier for A -stable linear multistep methods. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1639–1657. <https://doi.org/10.22067/ijnao.2025.94194.1673>

AMS subject classifications (2020): 65L05.

Keywords: Initial value problem; Stiff system; Linear multistep methods; Dahlquist's second barrier; convergence; Stability.

1 Introduction

Stiff ordinary differential equations (ODEs) in the form

$$\begin{aligned}y'(x) &= f(x, y(x)), & x \in [x_0, X], \\ y(x_0) &= y_0,\end{aligned}\tag{1}$$

arise frequently in scientific and engineering applications where the solution exhibits components with widely varying time scales. Numerical solution of such systems requires methods that remain stable even when large step sizes are used for the rapidly decaying components. Explicit methods generally fail in this regard due to severe stability restrictions, making implicit methods the preferred choice for stiff problems. Among implicit methods, A -stable methods play a crucial role. A numerical method is said to be A -stable if its region of absolute stability contains the entire left half of the complex plane. This means that when applied to the standard test problem of Dahlquist [9]

$$y' = \lambda y, \quad \lambda \in \mathbb{C},$$

with $\operatorname{Re}(\lambda) < 0$, the numerical solution decays to zero for any stepsize $h > 0$, mirroring the behavior of the exact solution. This property ensures numerical stability for stiff problems without requiring small step sizes. To relax the stringent requirement of A -stability, the concept of $A(\alpha)$ -stability is introduced. A method is $A(\alpha)$ -stable if its region of absolute stability contains a sector of the left half-plane bounded by two rays forming an angle 2α with the negative real axis. While not fully A -stable, such methods maintain strong stability properties for many stiff problems and can achieve higher order accuracy.

Implicit Runge–Kutta (IRK) methods can be constructed without theoretical limitations on order while preserving A -stability. For example, IRKs, such as those based on Gauss, Radau, and Lobatto quadratures, can attain arbitrarily high order while preserving A -stability [14, 20, 7]. However, these methods require solving nonlinear systems of equations involving multiple implicit stages at each time step which leads to significantly higher computational cost.

Linear multistep methods (LMMs) as a class of multivalue and one-stage methods, by incorporating past solution values and their derivatives, construct higher-order polynomial approximations that increase the order of accuracy without requiring additional function evaluations at intermediate stages within each step. A classical k -step LMM for solving (1) is given by

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j},$$

where α_j and β_j are parameters to be determined, $y_{n+j} \approx y(x_{n+j})$, h is the stepsize, and $f_{n+j} = f(x_{n+j}, y_{n+j})$. LMMs despite generally having lower computational cost than Runge–Kutta methods, suffer severe degradation of stability as their order increases. In particular, the requirement of A -stability puts a severe limitation on LMMs, which limits their applicability to stiff problems when high order accuracy is required. This pessimistic restriction is known as Dahlquist's second barrier.

Theorem 1 (Dahlquist's second barrier [9]). The maximal order of an A -stable LMM is two, and the trapezoidal rule is the unique method achieving this order with the minimal error constant.

Circumventing Dahlquist's second barrier poses challenges for designing efficient A - or $A(\alpha)$ -stable methods of high orders for stiff ODEs within the multistep framework. Developing such methods has been carried out by equipping traditional LMMs with various advanced techniques. A comprehensive understanding of the strategies involved in developing techniques to circumvent Dahlquist's second barrier is essential, as it enables researchers to design more effective and stable numerical algorithms tailored for stiff differential equations. Drawing on the authors' experience with methods over-

coming Dahlquist's second barrier, this paper surveys the successful research directions. This survey fills an existing gap in the literature by providing a unified overview of methods that overcome Dahlquist's second barrier. It highlights and compares various advanced techniques and their combinations that have been proposed to enhance the stability and accuracy of LMMs for stiff ODEs. By doing so, it offers researchers a comprehensive understanding of the strengths and limitations of each approach and fosters the generation of novel ideas for further advancements.

The paper is organized along the following lines. Section 2 introduces the advanced step-point strategy, reviewing several efficient methods based on backward differentiation formulas (BDF) that utilize this technique. In section 3, adaptive methods are discussed with a presentation of methods that incorporate adaptivity to enhance stability and accuracy. Section 4 focuses on second derivative methods as a successful strategy for improving both accuracy and stability. It demonstrates how LMMs have been enhanced using this approach and surveys several proposed methods that surpass Dahlquist's second barrier. Finally, section 5 concludes the paper with a summary of the main findings and remarks on future research directions.

2 Advanced step-point strategy

BDF methods constitute a widely used family of implicit LMMs for the numerical solution of ODEs, particularly effective for stiff problems. Initially developed by Curtiss and Hirschfelder [8] and later formalized by Gear [13], the k -step BDF method is given by

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \beta_k f_{n+k}. \quad (2)$$

Here, $\alpha_k = 1$ and the other coefficients are chosen so that the method has order $p = k$. A k -step BDF is A -stable for $k = p = 2$ and $A(\alpha)$ -stable for $k = p = 3, 4, 5, 6$; orders beyond six lose zero-stability and are generally not used in practice. Due to their favorable balance of stability and accuracy, BDF methods serve as the foundation for many robust stiff ODE solvers

such as LSODE and VODE [24, 19]. However, as a subclass of LMMs, they inherit the drawback that they cannot be A -stable for orders greater than two. Using the advanced step-point technique is one of the efficient strategies to overcome this drawback. In this way, some implicit advanced step-point (IAS) methods based on BDF methods have been introduced.

2.1 EBDF methods

Cash [4] enhanced BDF methods by incorporating the advanced step-point strategy, leading to the development of extended BDF (EBDF). The k -step EBDF method takes the form [4]

$$y_{n+k} + \sum_{j=0}^{k-1} \bar{\alpha}_j y_{n+j} = h \left(\bar{\beta}_k f_{n+k} + \bar{\beta}_{k+1} f_{n+k+1} \right), \quad (3)$$

where the coefficients are chosen to achieve order $p = k + 1$. Knowing the solutions y_{n+j} at the past nodes x_{n+j} , for $j = 0, 1, \dots, k - 1$, the EBDF algorithm proceeds as follows:

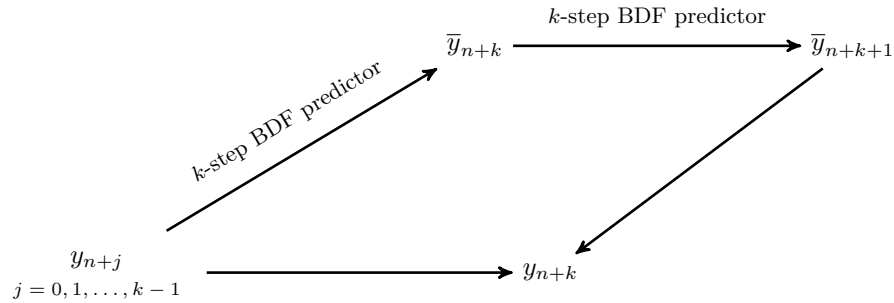
- The k -step BDF method predicts \bar{y}_{n+k} using y_{n+j} , $j = 0, 1, \dots, k - 1$.
- The k -step BDF method predicts \bar{y}_{n+k+1} using y_{n+j} , $j = 1, 2, \dots, k - 1$ and the predicted \bar{y}_{n+k} .
- Finally, the solution y_{n+k} is corrected using y_{n+j} , $j = 0, 1, \dots, k - 1$, and the predicted \bar{y}_{n+k+1} from (3) written in the form

$$y_{n+k} - h \bar{\beta}_k f_{n+k} = - \sum_{j=0}^{k-1} \bar{\alpha}_j y_{n+j} + h \bar{\beta}_{k+1} \bar{f}_{n+k+1},$$

where $\bar{f}_{n+k+1} = f(x_{n+k+1}, \bar{y}_{n+k+1})$.

The diagram of overall procedure of the EBDF methods has been plotted in Figure 1.

The EBDF methods are A -stable up to order *four* and $A(\alpha)$ -stable up to order *nine*, significantly improving the stability properties while achieving a higher order of convergence compared to classical BDF methods.

Figure 1: Diagram illustrating the k -step EBDf methods.

2.2 MEBDF methods

To avoid the need for computing and factorizing the two iteration matrices arising in the application of a modified Newton iteration at each stage—which leads to higher computational costs—EBDF approach was modified by Cash [6]. This modified method, known as the modified EBDf (MEBDF), replaces the corrector formula (3) with

$$y_{n+k} + \sum_{j=0}^{k-1} \bar{\alpha}_j y_{n+j} = h v_k \bar{f}_{n+k} + h(\bar{\beta}_k - v_k) f_{n+k} + h \bar{\beta}_{k+1} \bar{f}_{n+k+1}.$$

Here, the order of MEBDF is independent of the choice of v_k . Selecting $v_k = \bar{\beta}_k - \beta_k$, ensures that the coefficient matrix used in the modified Newton iteration scheme is the same for both the predictor and the corrector. This choice not only improves computational efficiency by requiring only one **LU** decomposition per step but also enlarges the $A(\alpha)$ -stability region compared to the original EBDf methods. The coefficients of the methods can be found in [4].

IAS methods have also been parallelized (so-called PIAS) aiming for significant efficiency gains and speed-ups, as shown by Psihoyios [22].

2.3 TIAS methods

The two implicit advanced step-point (TIAS) method, introduced by Psihoyios [22], extends the BDF family by incorporating two future points to improve accuracy and stability. The algorithm uses three predictor steps based on BDF and a corrector defined by

$$y_{n+k} + \sum_{j=0}^{k-1} \hat{\alpha}_j y_{n+j} = h \left(\hat{\beta}_k f_{n+k} + \hat{\beta}_{k+1} f_{n+k+1} + \hat{\beta}_{k+2} f_{n+k+2} \right). \quad (4)$$

Knowing the solutions y_{n+j} at the past nodes x_{n+j} , for $j = 0, 1, \dots, k-1$, the TIAS algorithm proceeds as follows:

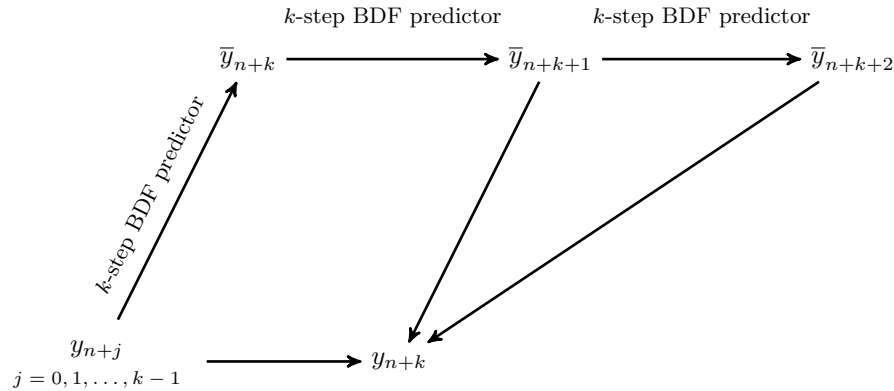
- The k -step BDF method predicts \bar{y}_{n+k} using y_{n+j} , $j = 0, 1, \dots, k-1$.
- The k -step BDF method predicts \bar{y}_{n+k+1} using y_{n+j} , $j = 1, 2, \dots, k-1$ and the computed \bar{y}_{n+k} .
- The k -step BDF method predicts \bar{y}_{n+k+2} using y_{n+j} , $j = 2, 3, \dots, k-1$ and the computed \bar{y}_{n+k} and \bar{y}_{n+k+1} .
- Finally, the TIAS corrector (4) computes the corrected solution y_{n+k} using y_{n+j} , $j = 0, 1, \dots, k-1$, and the predicted solutions \bar{y}_{n+k+1} and \bar{y}_{n+k+2} as

$$y_{n+k} - h\hat{\beta}_k f_{n+k} = - \sum_{j=0}^{k-1} \hat{\alpha}_j y_{n+j} + h\hat{\beta}_{k+1} \bar{f}_{n+k+1} + h\hat{\beta}_{k+2} \bar{f}_{n+k+2}.$$

The diagram of overall procedure of the TIAS methods has been plotted in Figure 2.

Using this approach, A -stable methods have been developed up to order *six*. However, this stability improvement was not achieved with the same level of optimization as in the MEBDF methods.

Considering the stability results of the classical BDF method (without advanced step-point) as well as those of methods with one and two advanced step-points aligns with the conjecture that the maximal order p of A -stable

Figure 2: Diagram illustrating the k -step TIAS methods.

methods increases with the number of advanced step-points, potentially following the relation:

$$p \leq 2q + 2,$$

where q is the number of advanced step-points. Based on the complexity involved in constructing A -stable methods of order six with $q = 2$, this conjecture has not yet been fully investigated [22].

A general formula was introduced in [23] that generates the stability functions of the methods BDF, EBDF, MEBDF, IAS, TIAS, and PIAS. This formula can substantially facilitate stability analysis and further computational manipulation of these and analogous schemes.

The features of the advanced step-point strategy have led to its application in the construction of other methods aimed at improving accuracy and stability properties. For example, Fazeli, Hojjati, and Shahmorad [11] introduced a class of multistep collocation methods for solving nonlinear Volterra integral equations, in which collocation points in the future interval, as well as in the current interval, are used. This technique results in high-order methods with an extensive absolute stability region.

3 Adaptive methods

Adaptive methods (also known as blended methods in some contexts) represent another effective technique for overcoming Dahlquist's second barrier. In this strategy, by incorporating adjustable parameters into the algorithms and tuning these to optimal values, the stability properties of the numerical methods can be significantly enhanced, enabling the construction of higher-order methods with improved absolute stability regions. This flexibility, when applied to LMMs, enables circumventing Dahlquist's second barrier.

3.1 AMF-BDF method

This strategy was first introduced by Skeel and Kong [25] by blending the k -step Adams–Moulton formula (AMF_k) and the k -step BDF (BDF_k) as

$$\text{AMF}_k - t h J \text{BDF}_k = 0,$$

in which $J = \frac{\partial f}{\partial y}$ is the Jacobian matrix of f with respect to y . This method is of order $p = k + 1$ for all values of t . The optimum values of t are given in [25]; see also [14], for which the method is A -stable up to order *four* and $A(\alpha)$ -stable up to order *twelve*, with larger values of α compared to the BDF method.

3.2 A-BDF method

The adaptive BDF (A-BDF), introduced by Fredebeul [12], generalizes the classical BDF methods by incorporating a parameter that can be optimized to improve stability properties. The k -step A-BDF method is a blended method of implicit and explicit BDF that can be expressed as

$$\text{A} - \text{BDF}_k(t) := \text{BDF}_k^{(i)} - t \text{BDF}_k^{(e)} = 0,$$

in which $\text{BDF}_k^{(i)}$ is the classical implicit k -step BDF (2), and $\text{BDF}_k^{(e)}$ is an explicit k -step BDF-type method defined by

$$\sum_{j=0}^k \alpha_j^* y_{n+j} = h\beta_{k-1}^* f_{n+k-1},$$

where $\alpha_k^* = 1$ and the other coefficients are chosen so that $\text{BDF}_k^{(e)}$ has order k . Therefore, a k -step A-BDF takes the form

$$\sum_{j=0}^k (\alpha_j - t\alpha_j^*) y_{n+j} = h\beta_k f_{n+k} - t\beta_{k-1}^* f_{n+k-1}.$$

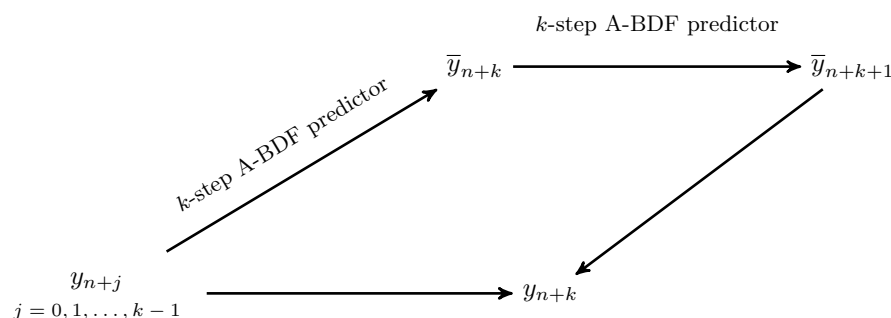
By finding the optimum values of the parameter t for each step number k , the maximum values of the angle α in $A(\alpha)$ -stability of A-BDF methods are achieved. The results reported in [12] show that the k -step A-BDF method is A -stable up to order *two* and $A(\alpha)$ -stable up to order *seven*, with larger values of α compared to the underlying classical k -step BDF.

3.3 A-EBDF method

The adaptive EBDF (A-EBDF), introduced by Hojjati, Rahimi Ardabili, and Hosseini [16], extends the A-BDF method to improve the stability properties of BDF, EBDF, and A-BDF. It combines two strategies—advanced step-point and adaptive methods—applied to the BDF algorithm. Knowing the solutions y_{n+j} at the past nodes x_{n+j} , $j = 0, 1, \dots, k-1$, the A-EBDF algorithm proceeds as follows:

- The k -step A-BDF method predicts \bar{y}_{n+k} using y_{n+j} , $j = 0, 1, \dots, k-1$.
- The k -step A-BDF method predicts \bar{y}_{n+k+1} using y_{n+j} , $j = 1, 2, \dots, k-1$ and the predicted \bar{y}_{n+k} .
- Finally, the k -step EBDF method (3) computes the solution y_{n+k} using y_{n+j} , $j = 0, 1, \dots, k-1$, and the predicted \bar{y}_{n+k+1} .

The diagram of overall procedure of the A-EBDF methods has been plotted in Figure 3.

Figure 3: Diagram illustrating the k -step A-EBDF methods.

It is proven that this scheme achieves order $k + 1$ for all values of the parameters $t \in \mathbb{R} \setminus \{1\}$. The optimum values of t resulting in the maximum angle α of $A(\alpha)$ -stability in A-EBDF are given in [16]. Stability analysis shows that the A-EBDF method is A -stable up to order *four* and $A(\alpha)$ -stable up to order *nine* with a larger angle α compared to the BDF, EBDF, and A-BDF methods. This improvement in stability properties results from the combination of the two aforementioned strategies applied to the BDF algorithm.

4 Second derivative methods

Incorporating the second derivative of the solution into numerical algorithms is an effective strategy to enhance both the accuracy and stability of the methods. Notably, in implicit methods, the use of the second derivative often incurs no additional computational cost. Specifically, for an autonomous problem of the form $y' = f(y)$, the second derivative can be expressed as $y'' = g = \frac{\partial f}{\partial y} f$, where $\frac{\partial f}{\partial y}$ is the Jacobian matrix of f . Also, for the Jacobian of g , a piecewise constant approximation of $(\frac{\partial f}{\partial y})^2$ is typically used.

The LMMs have been extended using this technique in many particular cases. A k -step second derivative linear multistep methods (SDMM) for the solution of the initial value problem (1) takes the general form

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j} + h^2 \sum_{j=0}^k \gamma_j g_{n+j},$$

where $g := y'' = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f$ and $g_{n+j} = g(x_{n+j}, y_{n+j})$. In this section, we survey some efficient SDMMs and those combined with other techniques mentioned in previous sections.

4.1 SDBDF methods

Second derivative BDF (SDBDF) methods extend the classical BDF methods by incorporating the second derivative of the solution to improve accuracy and stability, particularly for stiff problems [14]. A k -step SDBDF method takes the form

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \beta_k f_{n+k} + h^2 \gamma_k g_{n+k}, \quad (5)$$

where $\alpha_k = 1$ and the other coefficients are chosen so that the method has order $p = k + 1$.

The inclusion of the second derivative term $h^2 \gamma_k g_{n+k}$ enhances the stability region of the method beyond that of classical BDF schemes. Notably, SDBDF methods are A -stable up to order *four* and $A(\alpha)$ -stable up to order *eleven* thereby surpassing Dahlquist's second barrier. Beyond serving as efficient solvers in their own right, SDBDF schemes play a foundational role—analogous to classical BDF methods—in the development of advanced numerical methods discussed later in this section.

4.2 Enright methods

These methods were introduced by Enright [10] by enhancing the Adams methods through incorporating the second derivative of the solution into the algorithm. The general form of a k -step Enright method is

$$y_{n+k} - y_{n+k-1} = h \sum_{j=0}^k \beta_j f_{n+j} + h^2 \gamma_k g_{n+k}, \quad (6)$$

where the coefficients are chosen so that the method has order $p = k + 2$.

Inheriting from Adams' methods, the zero-stability of these methods is guaranteed for all values of the step number k . The methods are A -stable up to order *four* ($k = 2$) and $A(\alpha)$ -stable up to order *nine* ($k = 7$), while for $k = 8$ the stability region becomes disconnected. It is worth noting that the underlying Adams–Moulton methods are A -stable only up to order two for $k = 0, 1$, and for other $k \geq 2$ the stability region is bounded.

4.3 E2BD methods

The second derivative extended backward differentiation formulas (E2BD) were introduced by Cash [5] as an enhancement of Adams-type methods by incorporating two key techniques: The use of an advanced step-point and the inclusion of the second derivative of the solution. These methods are typically implemented in a predictor-corrector mode and are classified into two main classes:

E2BD methods – Class 1

Predictor: The Enright method (6).

$$\text{Corrector: } y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k+1} \bar{\beta}_j f_{n+j} + h^2 (\bar{\gamma}_k g_{n+k} + \bar{\gamma}_{k+1} g_{n+k+1}). \quad (7)$$

In this class, the corrector extends the Enright method (6) by incorporating the first and second derivatives of the solution at the future point x_{n+k+1} . The coefficients in (7) are chosen to achieve order $p = k + 4$. A k -step E2BD method of Class 1, considering the predictor's order, attains overall

order $p = k + 3$. These methods exhibit superior stability properties, being A -stable up to order *eight*.

E2BD methods – Class 2

Predictor: The Enright method (6).

$$\text{Corrector: } y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k+1} \bar{\beta}_j f_{n+j} + h^2 \bar{\gamma}_k g_{n+k}. \quad (8)$$

In this class, only the second derivative of the solution at the future point x_{n+k+1} is incorporated in the corrector. The coefficients in (8) are chosen so that the method has order $p = k + 3$. Considering the predictor's order, a k -step E2BD method of Class 2 also has order $p = k + 3$. While these methods are computationally more efficient than those of Class 1, they exhibit slightly weaker stability properties, being A -stable up to order *six*.

Class 1 methods for $k \geq 6$ and Class 2 methods for $k \geq 4$, are $A(\alpha)$ -stable with large stability angles α . For example, the 6-step E2BD method of Class 1 has $\alpha > 89^\circ$. This makes these methods well-suited for integrating stiff differential systems whose Jacobians have eigenvalues with large imaginary components close to the imaginary axis.

4.4 ESDMMs

The extended SDMMs (ESDMMs) were introduced by Hojjati, Rahimi Ardabili, and Hossein [17] as an enhancement of the SDBDF methods by incorporating the second derivative of the solution at the future point into the algorithm. These methods can be also considered as an extension of BDF schemes employing two key strategies: The use of an advanced step-point and the inclusion of the second derivative of the solution. A k -step ESDMM has the general form

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h\hat{\beta}_k f_{n+k} + h^2(\hat{\gamma}_k g_{n+k} - \hat{\gamma}_{k+1} g_{n+k+1}), \quad (9)$$

where $\hat{\alpha}_k = 1$ and the remaining coefficients are chosen to ensure the method attains order $p = k + 2$. Given the known solutions y_{n+j} at previous nodes x_{n+j} for $j = 0, 1, \dots, k - 1$, the ESDMM algorithm proceeds as follows:

- The k -step SDBDF (5) predicts \bar{y}_{n+k} using y_{n+j} , $j = 0, 1, \dots, k - 1$.
- The k -step SDBDF (5) predicts \bar{y}_{n+k+1} using y_{n+j} , $j = 1, 2, \dots, k - 1$ and the predicted \bar{y}_{n+k} .
- Finally, the k -step ESDMM (9) computes the solution y_{n+k} using y_{n+j} , $j = 0, 1, \dots, k - 1$, and the predicted \bar{y}_{n+k+1} as

$$y_{n+k} - h\hat{\beta}_k f_{n+k} - h^2\hat{\gamma}_k g_{n+k} = - \sum_{j=0}^{k-1} \hat{\alpha}_j y_{n+j} - h^2\hat{\gamma}_{k+1} \bar{y}_{n+k+1}.$$

The diagram of overall procedure of the ESDMMs has been plotted in Figure 4.

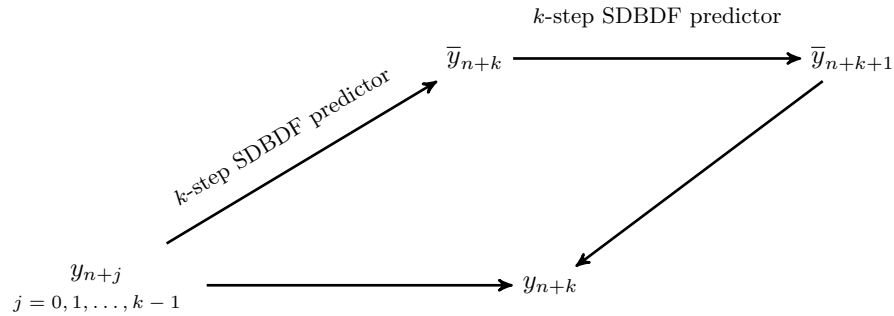


Figure 4: Diagram illustrating the k -step ESDMM methods.

ESDMMs exhibit A -stability up to order *six* and $A(\alpha)$ -stability up to order *fourteen*, with larger stability angles α compared to those of BDF and SDBDF methods.

In analogy with the motivation behind MEBDF methods, ESDMMs have been further refined into modified ESDMMs (MESDMMs) by replacing the corrector (9) with the following form [17]:

$$\sum_{j=0}^k \hat{\alpha}_j y_{n+j} = h(\hat{\beta}_k - \beta_k) \bar{f}_{n+k} + h\beta_k f_{n+k} + h^2(\hat{\gamma}_k - \gamma_k) \bar{g}_{n+k} \\ - h^2 \hat{\gamma}_{k+1} \bar{g}_{n+k+1} + h^2 \gamma_k g_{n+k}.$$

This modification not only reduces the computational cost associated with ESDMMs but also increases the stability angle α in the $A(\alpha)$ -stability property. MESDMMs have also been parallelized—referred to as PMESDMMs—to enable their efficient implementation on parallel computers [15].

A general formula introduced in [18] generates the stability functions for the SDBDF, ESDMMs, MESDMMs, and PMESDMMs. This formula helps to understand how modifying the structure of a method can effectively enhance its stability properties.

5 Conclusion

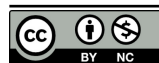
LMMs, as an efficient and flexible class of numerical methods for solving ODEs, face a significant challenge known as Dahlquist's second barrier, which limits their ability to solve stiff systems with high accuracy. This paper has investigated three effective strategies that overcome this barrier: Advanced step-point methods, adaptive methods and second derivative methods. These strategies can be applied individually or in combination to enhance LMMs. By analyzing their formulation and impact on stability, this study provides valuable insights for future research aimed at designing new and more robust algorithms. It is worth noting that other techniques, such as hybrid methods employing off-step points, also exist; however, the strategies discussed here represent general frameworks that generate entire classes of methods. Moreover, the first derivative methods, including LMMs, and second derivative methods, including SDMMs (and their modifications), are formulated within the general linear methods (GLMs) [3, 21] and second derivative GLMs (SGLMs) [1, 2] frameworks, respectively. Therefore, the strategies presented in this paper can be naturally extended to these more general frameworks.

References

- [1] Abdi, A. and Hojjati, G. *An extension of general linear methods*, Numer. Algorithms, 57 (2011), 149–167.
- [2] Butcher, J.C., and Hojjati, G. *Second derivative methods with RK stability*, Numer. Algorithms, 40 (2005), 415–429.
- [3] Butcher, J.C. *Numerical methods for ordinary differential equations*, Wiley, Chichester, 2016.
- [4] Cash, J.R. *On the integration of stiff systems of ODEs using extended backward differentiation formulas*, Numer. Math. 34(2) (1980), 235–246.
- [5] Cash, J.R. *Second derivative extended backward differentiation formulas for the numerical integration of stiff systems*, SIAM J. Numer. Anal. 18(2) (1981), 21–36.
- [6] Cash, J.R. *The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae*, Comput. Math. Appl. 9 (1983), 645–660.
- [7] Cong, N.H. and Thuy, N.T. *Stability of Two-Step-by-Two-Step IRK Methods Based on Gauss-Legendre Collocation Points and an Application*, Vietnam J. Math. 40(1) (2012), 115–126.
- [8] Curtiss, C.F. and Hirschfelder, J.O. *Integration of stiff equations*, Proceedings of the National Academy of Sciences, 38 (1952), 235–243.
- [9] Dahlquist, G. *A special stability problem for linear multistep methods*, BIT Numer. Math. 3 (1963) 27–43.
- [10] Enright, W.H. *Second derivative multistep methods for stiff ordinary differential equations*, SIAM J. Numer. Anal. 11 (1974), 321–331.
- [11] Fazeli, S., Hojjati, G., and Shahmorad, S. *Super implicit multistep collocation methods for nonlinear Volterra integral equations*, Math. Comput. Model. 55 (2012) 590–607.

- [12] Fredebeul, C. *A-BDF: A generalization of the backward differentiation formulae*, SIAM J. Numer. Anal. 35(5) (1998), 1917–1938.
- [13] Gear, C.W. *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, 1967.
- [14] Hairer, E. and Wanner, G. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, Springer, Berlin, 2010.
- [15] Hojjati, G. *A class of parallel methods with superfuture points technique for the numerical solution of stiff systems*, J. Mod. Meth. Numer. Math. 6 (2015), 57–63.
- [16] Hojjati, G., Rahimi Ardabili, M.Y., and Hosseini, S.M. *A-EBDF: an adaptive method for numerical solution of stiff systems of ODEs*, Math. Comput. Simul. 66 (2004), 33–41.
- [17] Hojjati, G., Rahimi Ardabili, M.Y. and Hosseini, S.M. *New second derivative multistep methods for stiff systems*, Appl. Math. Model. 30 (2006), 466–476.
- [18] Hojjati, G. and Taheri Koltape, L. *On the stability functions of second derivative implicit advanced-step point methods*, J. Math. Model. 10 (2022), 203–212.
- [19] Hindmarsh, A.C. *ODEPACK, a systematized collection of ODE solvers*, Scientific Computing, (1983), 55–64.
- [20] Iserles, A. *A first course in the numerical analysis of differential equations*, Cambridge University Press, 1996.
- [21] Jackiewicz, Z. *General Linear Methods for Ordinary Differential Equations*, Wiley, New Jersey, 2009.
- [22] Psihoyios, G. *Advanced step-point methods for the solution of initial value problems*, Ph.D. Thesis, University of London, Imperial College, 1995.
- [23] Psihoyios, G. *A general formula for the stability functions of a group of implicit advanced step-point (IAS) methods*, Math. Comput. Model. 46 (2007), 214–224.

- [24] Shampine, L.F. and Reichelt, M.W. *The MATLAB ODE suite*, SIAM J. Sci. Comput. 18(1) (1997), 1–22.
- [25] Skeel, R.D., Kong, A.K. *Blended linear multistep methods*, ACM TOMS 3 (1977), 326–343.



Portfolio optimization: A mean-variance approach for non-Markovian regime-switching markets

R. Keykhai

Abstract

This paper develops a novel multi-period mean-variance portfolio optimization framework for non-Markovian regime-switching markets, where state transition probabilities exhibit strong path-dependence. We propose an innovative dynamic programming solution that extends classical frameworks by incorporating path-dependent value functions through a rigorously derived modified Bellman equation. The solution involves constructing an auxiliary optimization problem using Lagrangian methods, with closed-form optimal strategies derived via matrix calculus. Analytically, we demonstrate that classical Markovian solutions emerge as special cases when path-dependence is removed. Numerical examples further demonstrate that our model could generate significantly lower-risk portfolios than Markovian alternatives by adaptively adjusting positions based on market history.

AMS subject classifications (2020): Primary 91G10; Secondary 90C39.

Received 5 April 2025; revised 14 August 2025; accepted 15 September 2025

Reza Keykhai

Department of Mathematics, Khansar Campus, University of Isfahan, Isfahan, Iran.

e-mail: r.keykhai@khc.ui.ac.ir

How to cite this article

Keykhai, R., Portfolio optimization: A mean-variance approach for non-Markovian regime-switching markets. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1658–1687.

<https://doi.org/10.22067/ijnao.2025.92882.1625>

Keywords: Portfolio optimization; Non-Markovian regime-switching; Path-dependence; Dynamic programming; Mean-variance analysis.

1 Introduction

The multi-period mean-variance portfolio optimization problem generalizes the static model introduced by Markowitz [16], where investors aim to minimize terminal wealth variance (risk) while targeting a fixed expected terminal wealth (reward). In this dynamic setting, investors sequentially rebalance their portfolios at discrete time intervals over the investment horizon according to an optimal strategy. While dynamic programming is the standard approach for solving stochastic optimization problems, the nonlinearity of the variance operator in the mean-variance framework renders direct application infeasible. However, Li and Ng [13] circumvented this challenge by embedding the original problem into an auxiliary problem solvable via dynamic programming, yielding explicit solutions. A critical limitation of their approach, however, lies in the assumption of independent asset returns across periods—a condition starkly contradicted by empirical financial market data, where return dependencies are well-documented. To address this, recent work by Cakmak and Ozekici [2] and others has extended the framework to regime-switching markets, where asset returns depend on the market state, modeled as a Markov chain. This formulation captures intertemporal return dependencies while preserving analytical tractability. Regime-switching models have become widely adopted for portfolio selection, asset allocation, and utility maximization in multi-period settings [3, 4, 6, 8, 9, 11, 21, 22, 25]. Hidden Markov chain approaches have been developed in

[1, 5, 27, 28, 29], offering alternative frameworks for modeling unobservable market regimes.

The Markovian assumption for market state processes in prior literature inherently ignores the historical path and evolution of market dynamics. This simplification is economically unrealistic, as market history and memory invariably influence current market behavior and, consequently, investor decisions. Although incorporating path-dependence significantly increases model complexity and poses analytical challenges, some recent works have started

to explore non-Markovian frameworks. In particular, several studies have introduced models where asset returns depend on the historical trajectory of the market rather than solely on the current state. Specifically, these models generalize state-dependent coefficients by conditioning them on the filtration generated by a Markov chain

[7, 12, 17, 18, 19, 20]. However, it should be noted that the majority of these studies have focused exclusively on continuous-time settings, with the exception of [12], which examines a discrete-time, multi-period framework. A fundamental limitation persists that these models still assume the underlying market state follows a Markov process (i.e., transition probabilities remain path-independent). This contradicts empirical evidence where market history exerts persistent effects—exemplified by momentum (trend persistence) or regime shifts after prolonged bull/bear markets. Such phenomena demonstrate that market states are inherently non-Markovian. In addition to regime-switching models, several studies have proposed non-Markovian approaches that do not rely on market states, but instead capture temporal dependence by modeling serial correlation among asset returns directly

[10, 23, 24, 26]. While these approaches account for time dependence, they do not explicitly model market states or regime transitions.

To the best of our knowledge, the mean-variance portfolio optimization problem has not been studied in regime-switching markets where the state process is non-Markovian. In this work, we examine this scenario where asset returns depend on current market states while transition probabilities exhibit path-dependence rather than following Markovian properties. Models such as hidden Markov models and semi-Markov processes also incorporate memory effects, typically by introducing latent variables or duration-based transitions. In contrast, our model allows transition probabilities to depend explicitly on the observed history of market states, thereby providing a more direct form of path dependence. Unlike the embedding technique developed by Li and Ng [13], we employ the alternative Lagrangian multipliers method introduced by Li, Zhou, and Lim [14] to solve this problem. Our solution methodology proceeds in two key stages: First, we construct and solve an auxiliary optimization problem through the method of Lagrangian multipliers; then, we recover the solution to the original problem by applying

Lagrangian duality theorem. To incorporate path-dependence, we derive an extended Bellman equation based on path-dependent value functions. The computational complexity arising from path-dependent parameters requires matrix calculus to derive explicit forms for both the optimal strategy and the mean-variance efficient frontier. Importantly, our model generalizes classical Markovian frameworks, which emerge as special cases when path-dependence is eliminated.

This paper is organized as follows: Section 2 presents our non-Markovian regime-switching market model including all assumptions, problem formulation, and mathematical preliminaries. Section 3 introduces the auxiliary problem and solves it using dynamic programming, ultimately solving the original problem through Lagrangian duality. The special case of Markovian regime-switching markets is examined in Section 4. A numerical illustration comparing our proposed model with classical approaches appears in Section 5. Finally, Section 6 concludes with key findings and suggestions for future research.

2 Model framework: Assumptions and formulation

We consider a discrete-time, multi-period investment horizon with T periods, where the financial market consists of $N + 1$ assets. Among these assets, one is risk-free, and the remaining N are risky assets. Let $\{\Theta_n\}_{n=0}^T$ denote the sequence of market states (or regimes) over the investment horizon, where $\Theta_n \in \{1, 2, \dots, M\}$ represents the regime of the market at time n . The market exhibits regime-switching dynamics, but the transitions between states are not governed by a Markov process. Instead, the transition probabilities depend on the entire past trajectory of the market states, encapsulating a path-dependent behavior. The return of the i th asset at time n , denoted by $R_n^i(\Theta_n)$, depends on the market state Θ_n , where $i = 0, 1, \dots, N$, with $i = 0$ representing the risk-free asset. The transition probabilities between market states depend not only on the current state Θ_{n-1} but also on the past trajectory of the market, represented by $(\Theta_0, \Theta_1, \dots, \Theta_{n-1}) = (\theta_0, \theta_1, \dots, \theta_{n-1})$. The probability of transitioning from state θ_{n-1} at time $n - 1$ to state θ_n at time n under the history of market states $(\theta_0, \theta_1, \dots, \theta_{n-1})$ is defined as

$$Q_n(\theta_0, \dots, \theta_{n-1}, \theta_n) = P(\Theta_n = \theta_n \mid \Theta_{n-1} = \theta_{n-1}, \dots, \Theta_1 = \theta_1, \Theta_0 = \theta_0),$$

where $P(\cdot)$ is the probability measure and

$$\sum_{\theta_n=1}^M Q_n(\theta_0, \dots, \theta_{n-1}, \theta_n) = 1.$$

This non-Markovian model allows for a richer and more realistic representation of market dynamics, accommodating situations where the probability of state transitions is influenced by prior trends, volatility clusters, or other historical features.

At each time $n \in \{0, 1, \dots, T-1\}$, the investor allocates their wealth among the $N+1$ assets in the market. Let $\pi_n = (\pi_n^1, \pi_n^2, \dots, \pi_n^N)' \in \mathbb{R}^N$ (where $'$ denotes transpose) denote the portfolio assigned to the N risky assets at time n , where π_n^i represents the wealth allocated to the i -th risky asset. The remaining wealth

$$W_n - \sum_{i=1}^N \pi_n^i$$

is allocated to the risk-free asset, where W_n denote the total investor's wealth at time n . Here, $\pi = \{\pi_0, \pi_1, \dots, \pi_{T-1}\}$ denotes the overall investment strategy over the entire investment horizon. The evolution of the investor's wealth is driven by the returns of the assets. Assuming that the returns are expressed relative to the risk-free asset, the excess return of the i -th risky asset at time n in regime Θ_n is given by

$$R_n^{e,i}(\Theta_n) = R_n^i(\Theta_n) - R_n^0(\Theta_n).$$

The wealth dynamics over time can then be expressed as

$$W_{n+1} = (W_n - \sum_{i=1}^N \pi_n^i) R_n^0(\Theta_n) + \sum_{i=1}^N \pi_n^i R_n^i(\Theta_n) = W_n R_n^0(\Theta_n) + \pi_n' R_n^e(\Theta_n),$$

where $R_n^e(\Theta_n) = (R_n^{e,1}(\Theta_n), \dots, R_n^{e,N}(\Theta_n))'$.

We assume the following regarding the returns of the assets and market states. Let $R_n(\Theta_n) = (R_n^0(\Theta_n), \dots, R_n^N(\Theta_n))'$ represent the vector of asset returns under the market state Θ_n . For different time points $m \neq n$, given the market states $\Theta_n = \theta_n$ and $\Theta_m = \theta_m$, the random vectors $R_n(\theta_n)$ and $R_m(\theta_m)$ are independent. Additionally, the future market state Θ_{n+1} is assumed to be independent of the given current wealth $W_n = w_n$. Also, we assume that the covariance matrix corresponding to the returns of the assets for a given market state is positive definite. Finally, we assume frictionless trading and exclude transaction costs from the model for analytical tractability.

The objective of the investor in this framework is to construct a portfolio strategy π^* that minimizes the variance of the terminal wealth W_T while achieving a predefined level of expected terminal wealth, E_T . The multi-period Markowitz's mean-variance optimization problem can be formulated as

$$P(MV) : \begin{cases} \min_{\pi} \text{Var}_0 [W_T] \\ \text{s.t. } \mathbb{E}_0 [W_T] = E_T, \\ W_{n+1} = W_n R_n^0(\Theta_n) + \pi'_n R_n^e(\Theta_n), \end{cases}$$

where \mathbb{E}_0 and Var_0 represent the expectation and variance operators under the initial market condition $\Theta_0 = \theta_0$.

In this framework, the path-dependence of the market regimes Θ_n introduces additional complexity. The optimal strategy π^* is inherently influenced by the entire history of market states $(\Theta_0, \Theta_1, \dots, \Theta_n)$, reflecting the non-Markovian nature of the regime-switching dynamics. To analyze and solve this problem, we employ dynamic programming principles, incorporating the path-dependent transition probabilities and the regime-dependent asset returns into the optimization framework.

Before addressing the solution to the optimization problem, we introduce some certain matrix notations that will play a fundamental role in simplifying the subsequent computations.

For the given market state $\Theta_n = \theta_n$, we define

$$h_n(\theta_n) = \bar{R}_n^e(\theta_n)' V_n(\theta_n)^{-1} \bar{R}_n^e(\theta_n),$$

$$\begin{aligned} g_n(\theta_n) &= R_n^0(\theta_n) (1 - h_n(\theta_n)), \\ f_n(\theta_n) &= R_n^0(\theta_n)^2 (1 - h_n(\theta_n)), \end{aligned}$$

where

$$\begin{aligned} V_n(\theta_n) &= \mathbb{E} [R_n^e(\theta_n) R_n^e(\theta_n)'], \\ \bar{R}_n^e(\theta_n) &= \mathbb{E} [R_n^e(\theta_n)]. \end{aligned}$$

Assuming the positive definiteness of the covariance matrices, we state the following lemma. For the proof, please refer to [2, Lemmas 1 and 2]. Note that the following lemma guarantees the invertibility of the matrix $V_n(\theta_n)$ in the above notations.

Lemma 1. The matrix $V_n(\theta_n)$ is positive definite. Additionally, the scalars $f_n(\theta_n)$ and $g_n(\theta_n)$ are strictly positive, and $h_n(\theta_n)$ satisfies $0 < h_n(\theta_n) < 1$.

Let $C_n \in \mathbb{R}^M$ be a column vector, and let B_n be a tensor of order $(n+1)$ with shape $M \times M \times \cdots \times M$, where M is the number of market states. We denote $(Q_n \bullet B_n)$ as an $M \times M \times \cdots \times M$ tensor of order $(n+1)$, and $\overline{(Q_n \bullet B_n)}$ and $\overline{(Q_{C_n} \bullet B_n)}$ as $M \times M \times \cdots \times M$ tensors of order n defined as follows:

$$\begin{aligned} (Q_n \bullet B_n)(\theta_0, \dots, \theta_n) &= Q_n(\theta_0, \dots, \theta_n) B_n(\theta_0, \dots, \theta_n), \\ \overline{(Q_n \bullet B_n)}(\theta_0, \dots, \theta_{n-1}) &= \sum_{\theta_n=1}^M Q_n(\theta_0, \dots, \theta_{n-1}, \theta_n) B_n(\theta_0, \dots, \theta_{n-1}, \theta_n), \\ \overline{(Q_{C_n} \bullet B_n)}(\theta_0, \dots, \theta_{n-1}) &= \sum_{\theta_n=1}^M Q_n(\theta_0, \dots, \theta_n) C_n(\theta_n) B_n(\theta_0, \dots, \theta_n). \end{aligned}$$

Using these notations, for $1 \leq n < k$ ($k \in \mathbb{N}$), we define

$$\overline{\prod_{j=k-n}^{k-1} Q_{C_j} \bullet (Q_k \bullet B_k)} = \overline{(Q_{C_{k-n}} \bullet (\dots \bullet (Q_{C_{k-1}} \bullet \overline{(Q_k \bullet B_k)}) \dots))}, \quad (1)$$

as an $M \times M \times \cdots \times M$ tensor of order $(k-n)$. For convenience, we set

$$\overline{\prod_{\emptyset} Q_{C_j} \bullet (Q_k \bullet B_k)} = \overline{(Q_k \bullet B_k)}.$$

Furthermore, we use the conventions

$$\sum_{\emptyset}(\cdot) = \mathbf{0}, \quad \prod_{\emptyset}(\cdot) = I,$$

where I denotes the identity matrix.

By applying mathematical induction, we can derive the following lemmas. See also [12, Lemmas 2 and 3].

Lemma 2. For $n \geq 0$,

$$\begin{aligned} & \overline{\prod_{j=k-n}^{k-1} Q_{C_j} \bullet (Q_k \bullet B_k)}(\theta_0, \dots, \theta_{k-n-1}) \\ &= \sum_{\theta_{k-n}=1}^M \dots \sum_{\theta_{k-1}=1}^M \sum_{\theta_k=1}^M Q_{k-n}(\theta_0, \dots, \theta_{k-n}) C_{k-n}(\theta_{k-n}) \dots Q_{k-1}(\theta_0, \dots, \theta_{k-1}) \\ & \quad \times C_{k-1}(\theta_{k-1}) Q_k(\theta_0, \dots, \theta_k) B_k(\theta_0, \dots, \theta_k). \end{aligned}$$

Lemma 3. Let $\{C_n\}_{n=0}^{T-1}$ be a sequence of M -column vectors and let $\{B_n\}_{n=0}^T$ be a sequence of $M \times M \times \dots \times M$ tensors of order $(n+1)$. Define the sequence $\{A_n\}_{n=0}^T$ of $M \times M \times \dots \times M$ tensors of order $(n+1)$ recursively as follows:

$$\begin{aligned} A_n(\theta_0, \dots, \theta_n) &= B_n(\theta_0, \dots, \theta_n) + C_n(\theta_n) \overline{(Q_{n+1} \bullet A_{n+1})}(\theta_0, \dots, \theta_n), \\ A_T(\theta_0, \dots, \theta_T) &= B_T(\theta_0, \dots, \theta_T). \end{aligned}$$

Then,

$$\begin{aligned} A_n(\theta_0, \dots, \theta_n) &= B_n(\theta_0, \dots, \theta_n) \\ & \quad + C_n(\theta_n) \sum_{k=n+1}^T \overline{\prod_{j=n+1}^{k-1} Q_{C_j} \bullet (Q_k \bullet B_k)}(\theta_0, \dots, \theta_n). \end{aligned}$$

3 Dynamic programming formulation

The dynamic programming method serves as a powerful tool for solving stochastic optimization problems, particularly in multi-stage decision-making scenarios. This approach facilitates breaking down complex problems into smaller, more manageable subproblems. However, when applied to the mean-variance portfolio selection problem, dynamic programming faces challenges

due to the nonseparability of variance in the dynamic programming framework. To address this issue, Li and Ng [13] introduced an embedding technique that redefines the problem, allowing the variance to be addressed indirectly. By constructing an auxiliary problem, they derived solutions for the classical mean-variance problem. While effective, their approach can be computationally intricate and is not always intuitive. In subsequent work, Li, Zhou, and Lim [14] proposed a more practical method by employing the Lagrange duality technique to derive optimal solutions. This approach reduces computational complexity and simplifies the solution process. In this study, we adopt a similar method to solve the problem $P(MV)$.

First, we reformulate $P(MV)$ using the variance definition, resulting in the following formulation:

$$P(MV) : \begin{cases} \min_{\pi} \mathbb{E}_0 [(W_T - E_T)^2] \\ \text{s.t. } \mathbb{E}_0 [W_T - E_T] = 0, \\ W_{n+1} = W_n R_n^0(\Theta_n) + \pi'_n R_n^e(\Theta_n). \end{cases}$$

By introducing the Lagrange multiplier $2\lambda \in \mathbb{R}$, the constrained problem can be transformed into the following unconstrained formulation:

$$\tilde{P}(MV) : \begin{cases} \min_{\pi} \mathbb{E}_0 [(W_T - E_T)^2] + 2\lambda \mathbb{E}_0 [W_T - E_T] \\ \text{s.t. } W_{n+1} = W_n R_n^0(\Theta_n) + \pi'_n R_n^e(\Theta_n). \end{cases}$$

Introducing the substitutions $d_1 = 2(\lambda - E_T)$ and $d_0 = E_T^2 - 2\lambda E_T$, we obtain

$$\tilde{P}(MV) : \begin{cases} \min_{\pi} \mathbb{E}_0 [W_T^2 + d_1 W_T + d_0] \\ \text{s.t. } W_{n+1} = W_n R_n^0(\Theta_n) + \pi'_n R_n^e(\Theta_n). \end{cases}$$

3.1 Solution to problem $\tilde{P}(MV)$

To determine the optimal solution for problem $\tilde{P}(MV)$, the approach involves minimizing the expected cost function of the terminal wealth, that is,

$$\min_{\pi} \mathbb{E}_0 [g(W_T)],$$

using the dynamic programming approach, where the cost function is defined as

$$g(W_T) = W_T^2 + d_1 W_T + d_0.$$

Define $J_n(\theta_0, \dots, \theta_n; w_n; \pi_n)$ as the expected cost incurred when employing the investment policy π_n at time n , followed by optimal strategies from time $n+1$ to T . This is conditional on the market path $(\theta_0, \dots, \theta_n)$ and the wealth w_n available at time n . Accordingly,

$$v_n(\theta_0, \dots, \theta_n; w_n) = \min_{\pi_n} J_n(\theta_0, \dots, \theta_n; w_n; \pi_n),$$

represents the optimal expected cost under the given market path $(\theta_0, \dots, \theta_n)$ and the available wealth w_n at stage n . Using the dynamic programming principle, the relationship between J_n and v_{n+1} is given by

$$\begin{aligned} & J_n(\theta_0, \dots, \theta_n; w_n; \pi_n) \\ &= \mathbb{E}[v_{n+1}(\theta_0, \dots, \theta_n, \Theta_{n+1}; W_{n+1}(\pi_n)) \mid \Theta_0 = \theta_0, \dots, \Theta_n = \theta_n, W_n = w_n], \end{aligned}$$

where $W_{n+1}(\pi_n)$ is the wealth at time $n+1$ resulting from applying policy π_n . The dynamic programming equation (DPE) for this problem can thus be expressed as

$$\begin{aligned} & v_n(\theta_0, \dots, \theta_n; w_n) \\ &= \min_{\pi_n} \mathbb{E}[v_{n+1}(\theta_0, \dots, \theta_n, \Theta_{n+1}; W_{n+1}(\pi_n)) \mid \Theta_0 = \theta_0, \dots, \Theta_n = \theta_n, W_n = w_n]. \end{aligned}$$

Rewriting this, the DPE becomes

$$\begin{aligned} v_n(\theta_0, \dots, \theta_n; w_n) = \min_{\pi_n} & \left\{ \sum_{\theta_{n+1}=1}^M Q_{n+1}(\theta_0, \dots, \theta_n, \theta_{n+1}) \right. \\ & \left. \times \mathbb{E}[v_{n+1}(\theta_0, \dots, \theta_{n+1}; w_n R_n^0(\theta_n) + \pi'_n R_n^e(\theta_n))] \right\}, \end{aligned} \quad (2)$$

with the boundary condition

$$v_T(\theta_0, \dots, \theta_T; w_T) = w_T^2 + d_1 w_T + d_0. \quad (3)$$

The DPE is solved recursively, starting from the terminal condition at T and proceeding backward to $n = 0$, to determine the optimal strategy.

The following theorem provides the main result of our analysis, offering an explicit solution to problem $\tilde{P}(MV)$.

Theorem 1. For $n = 0, 1, \dots, T-1$, $v_n(\theta_0, \dots, \theta_n; w_n)$ is given by

$$v_n(\theta_0, \dots, \theta_n; w_n) = a_n(\theta_0, \dots, \theta_n)w_n^2 + b_n(\theta_0, \dots, \theta_n)w_n + c_n(\theta_0, \dots, \theta_n), \quad (4)$$

under the optimal policy

$$\begin{aligned} & \pi_n^*(\theta_0, \dots, \theta_n; w_n) \\ &= - \left[w_n R_n^0(\theta_n) + \frac{\overline{\prod_{j=n+1}^{T-1} Q_{g_j} \bullet (Q_T \bullet \mathbf{d}_T)}(\theta_0, \dots, \theta_n)}{2 \overline{\prod_{j=n+1}^{T-1} Q_{f_j} \bullet (Q_T \bullet \mathbf{1}_T)}(\theta_0, \dots, \theta_n)} \right] V_n(\theta_n)^{-1} \bar{R}_n^e(\theta_n), \end{aligned} \quad (5)$$

where

$$\begin{aligned} a_n(\theta_0, \dots, \theta_n) &= f_n(\theta_n) \overline{\prod_{j=n+1}^{T-1} Q_{f_j} \bullet (Q_T \bullet \mathbf{1}_T)}(\theta_0, \dots, \theta_n), \\ b_n(\theta_0, \dots, \theta_n) &= g_n(\theta_n) \overline{\prod_{j=n+1}^{T-1} Q_{g_j} \bullet (Q_T \bullet \mathbf{d}_T)}(\theta_0, \dots, \theta_n), \\ c_n(\theta_0, \dots, \theta_n) &= e_n(\theta_0, \dots, \theta_n) + \sum_{k=n+1}^T \overline{\prod_{j=n+1}^{k-1} Q_{1_j} \bullet (Q_k \bullet e_k)}(\theta_0, \dots, \theta_n), \\ e_n(\theta_0, \dots, \theta_n) &= - \frac{\left[\overline{\prod_{j=n+1}^{T-1} Q_{g_j} \bullet (Q_T \bullet \mathbf{d}_T)}(\theta_0, \dots, \theta_n) \right]^2}{4 \overline{\prod_{j=n+1}^{T-1} Q_{f_j} \bullet (Q_T \bullet \mathbf{1}_T)}(\theta_0, \dots, \theta_n)} h_n(\theta_n), \end{aligned}$$

$e_T(\theta_0, \dots, \theta_T) = d_0$, $\mathbf{d}_T(\theta_0, \dots, \theta_T) = d_1$, $\mathbf{1}_T(\theta_0, \dots, \theta_T) = 1$ and $\mathbf{1}_j(\theta_j) = 1$ ($j = 1, 2, \dots, T-1$).

Remark 1. In the quadratic value function, the coefficient a_n captures the impact of risk (variance), b_n relates to expected return, and c_n reflects the accumulated path-dependent effect independent of current wealth.

Proof. By mathematical induction, we first establish (4) under the following recursive relationships:

$$\begin{aligned}
a_n(\theta_0, \dots, \theta_n) &= f_n(\theta_n) \overline{(Q_{n+1} \bullet a_{n+1})}(\theta_0, \dots, \theta_n) > 0, \quad a_T(\theta_0, \dots, \theta_T) = 1, \\
b_n(\theta_0, \dots, \theta_n) &= g_n(\theta_n) \overline{(Q_{n+1} \bullet b_{n+1})}(\theta_0, \dots, \theta_n), \quad b_T(\theta_0, \dots, \theta_T) = d_1, \\
c_n(\theta_0, \dots, \theta_n) &= e_n(\theta_0, \dots, \theta_n) + \overline{(Q_{n+1} \bullet c_{n+1})}(\theta_0, \dots, \theta_n), \quad c_T(\theta_0, \dots, \theta_T) = d_0,
\end{aligned}$$

where

$$e_n(\theta_0, \dots, \theta_n) = -\frac{\left[\overline{(Q_{n+1} \bullet b_{n+1})}(\theta_0, \dots, \theta_n)\right]^2}{4\overline{(Q_{n+1} \bullet a_{n+1})}(\theta_0, \dots, \theta_n)} h_n(\theta_n).$$

For $n = T$, the boundary condition (3) provides the terminal values for a_T , b_T , and c_T . Now, let $n = T - 1$. Given an arbitrary market path $(\theta_0, \dots, \theta_{T-1})$ and the available wealth w_{T-1} , (2) leads to

$$\begin{aligned}
&v_{T-1}(\theta_0, \dots, \theta_{T-1}; w_{T-1}) \\
&= \min_{\pi_{T-1}} \mathbb{E} \left\{ \sum_{\theta_T=1}^M Q_T(\theta_0, \dots, \theta_T) \right. \\
&\quad \times v_T(\theta_0, \dots, \theta_T; w_{T-1} R_{T-1}^0(\theta_{T-1}) + \pi'_{T-1} R_{T-1}^e(\theta_{T-1})) \Big\} \\
&= \min_{\pi_{T-1}} \mathbb{E} \left\{ \sum_{\theta_T=1}^M Q_T(\theta_0, \dots, \theta_T) \right. \\
&\quad \times a_T(\theta_0, \dots, \theta_T) [w_{T-1} R_{T-1}^0(\theta_{T-1}) + \pi'_{T-1} R_{T-1}^e(\theta_{T-1})]^2 \\
&\quad + \sum_{\theta_T=1}^M Q_T(\theta_0, \dots, \theta_T) \\
&\quad \times b_T(\theta_0, \dots, \theta_T) [w_{T-1} R_{T-1}^0(\theta_{T-1}) + \pi'_{T-1} R_{T-1}^e(\theta_{T-1})] \\
&\quad \left. + \sum_{\theta_T=1}^M Q_T(\theta_0, \dots, \theta_T) c_T(\theta_0, \dots, \theta_T) \right\} \\
&= \min_{\pi_{T-1}} \mathbb{E} \left\{ \overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1}) [w_{T-1} R_{T-1}^0(\theta_{T-1}) + \pi'_{T-1} R_{T-1}^e(\theta_{T-1})]^2 \right. \\
&\quad + \overline{(Q_T \bullet b_T)}(\theta_0, \dots, \theta_{T-1}) [w_{T-1} R_{T-1}^0(\theta_{T-1}) + \pi'_{T-1} R_{T-1}^e(\theta_{T-1})] \\
&\quad \left. + \overline{(Q_T \bullet c_T)}(\theta_0, \dots, \theta_{T-1}) \right\} \\
&= \overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1}) w_{T-1}^2 R_{T-1}^0(\theta_{T-1})^2 \\
&\quad + \overline{(Q_T \bullet b_T)}(\theta_0, \dots, \theta_{T-1}) w_{T-1} R_{T-1}^0(\theta_{T-1}) + \overline{(Q_T \bullet c_T)}(\theta_0, \dots, \theta_{T-1})
\end{aligned}$$

$$\begin{aligned}
& + \min_{\pi_{T-1}} \left\{ \overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1}) \pi'_{T-1} V_{T-1}(\theta_{T-1}) \pi_{T-1} \right. \\
& + \left[2 \overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1}) w_{T-1} R_{T-1}^0(\theta_{T-1}) + \overline{(Q_T \bullet b_T)}(\theta_0, \dots, \theta_{T-1}) \right] \\
& \left. \times \pi'_{T-1} \bar{R}_{T-1}^e(\theta_{T-1}) \right\}. \tag{6}
\end{aligned}$$

Since Lemma 1 establishes that $V_{T-1}(\theta_{T-1})$ is positive definite and the positivity of a_T ensures that $\overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1})$ remains positive, it follows that the Hessian matrix of the objective function in (6) is positive definite. Thus, setting the gradient to zero provides the necessary and sufficient optimality condition

$$\begin{aligned}
& \overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1}) V_{T-1}(\theta_{T-1}) \pi_{T-1} \\
& + \left[\overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1}) w_{T-1} R_{T-1}^0(\theta_{T-1}) + \frac{1}{2} \overline{(Q_T \bullet b_T)}(\theta_0, \dots, \theta_{T-1}) \right] \\
& \times \bar{R}_{T-1}^e(\theta_{T-1}) = \mathbf{0},
\end{aligned}$$

which leads to the optimal policy

$$\begin{aligned}
& \pi_{T-1}^*(\theta_0, \dots, \theta_{T-1}; w_{T-1}) = \\
& - \left[w_{T-1} R_{T-1}^0(\theta_{T-1}) + \frac{\overline{(Q_T \bullet b_T)}(\theta_0, \dots, \theta_{T-1})}{2 \overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1})} \right] V_{T-1}(\theta_{T-1})^{-1} \bar{R}_{T-1}^e(\theta_{T-1}).
\end{aligned}$$

By substituting this optimal policy in (6), we derive

$$\begin{aligned}
& v_{T-1}(\theta_0, \dots, \theta_{T-1}; w_{T-1}) \\
& = a_{T-1}(\theta_0, \dots, \theta_{T-1}) w_{T-1}^2 + b_{T-1}(\theta_0, \dots, \theta_{T-1}) w_{T-1} + c_{T-1}(\theta_0, \dots, \theta_{T-1}),
\end{aligned}$$

where

$$\begin{aligned}
& a_{T-1}(\theta_0, \dots, \theta_{T-1}) = f_{T-1}(\theta_{T-1}) \overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1}) > 0, \\
& b_{T-1}(\theta_0, \dots, \theta_{T-1}) = g_{T-1}(\theta_{T-1}) \overline{(Q_T \bullet b_T)}(\theta_0, \dots, \theta_{T-1}), \\
& c_{T-1}(\theta_0, \dots, \theta_{T-1}) = - \frac{\left[\overline{(Q_T \bullet b_T)}(\theta_0, \dots, \theta_{T-1}) \right]^2}{4 \overline{(Q_T \bullet a_T)}(\theta_0, \dots, \theta_{T-1})} h_{T-1}(\theta_{T-1}) \\
& \quad + \overline{(Q_T \bullet c_T)}(\theta_0, \dots, \theta_{T-1}).
\end{aligned}$$

Observe that the positivity of a_{T-1} is a direct consequence of Lemma 1 and the positivity of a_T .

Now, suppose that (4) holds for $n+1$. We will establish its validity for n , considering the market path $(\theta_0, \dots, \theta_n)$ and the corresponding wealth level w_n . Utilizing the induction hypothesis and equation (2), we derive

$$\begin{aligned}
 & v_n(\theta_0, \dots, \theta_n; w_n) \\
 &= \min_{\pi_n} \mathbb{E} \left\{ \sum_{\theta_{n+1}=1}^M Q_{n+1}(\theta_0, \dots, \theta_{n+1}) v_{n+1}(\theta_0, \dots, \theta_{n+1}; w_n R_n^0(\theta_n) + \pi'_n R_n^e(\theta_n)) \right\} \\
 &= \min_{\pi_n} \mathbb{E} \left\{ \sum_{\theta_{n+1}=1}^M Q_{n+1}(\theta_0, \dots, \theta_{n+1}) a_{n+1}(\theta_0, \dots, \theta_{n+1}) [w_n R_n^0(\theta_n) + \pi'_n R_n^e(\theta_n)]^2 \right. \\
 &\quad + \sum_{\theta_{n+1}=1}^M Q_{n+1}(\theta_0, \dots, \theta_{n+1}) b_{n+1}(\theta_0, \dots, \theta_{n+1}) [w_n R_n^0(\theta_n) + \pi'_n R_n^e(\theta_n)] \\
 &\quad \left. + \sum_{\theta_{n+1}=1}^M Q_{n+1}(\theta_0, \dots, \theta_{n+1}) c_{n+1}(\theta_0, \dots, \theta_{n+1}) \right\} \\
 &= \min_{\pi_n} \mathbb{E} \left\{ \overline{(Q_{n+1} \bullet a_{n+1})}(\theta_0, \dots, \theta_n) [w_n R_n^0(\theta_n) + \pi'_n R_n^e(\theta_n)]^2 \right. \\
 &\quad + \overline{(Q_{n+1} \bullet b_{n+1})}(\theta_0, \dots, \theta_n) [w_n R_n^0(\theta_n) + \pi'_n R_n^e(\theta_n)] \\
 &\quad \left. + \overline{(Q_{n+1} \bullet c_{n+1})}(\theta_0, \dots, \theta_n) \right\} \\
 &= \overline{(Q_{n+1} \bullet a_{n+1})}(\theta_0, \dots, \theta_n) w_n^2 R_n^0(\theta_n)^2 + \overline{(Q_{n+1} \bullet b_{n+1})}(\theta_0, \dots, \theta_n) w_n R_n^0(\theta_n) \\
 &\quad + \overline{(Q_{n+1} \bullet c_{n+1})}(\theta_0, \dots, \theta_n) \\
 &\quad + \min_{\pi_n} \left\{ \overline{(Q_{n+1} \bullet a_{n+1})}(\theta_0, \dots, \theta_n) \pi'_n V_n(\theta_n) \pi_n \right. \\
 &\quad + \left[2 \overline{(Q_{n+1} \bullet a_{n+1})}(\theta_0, \dots, \theta_n) w_n R_n^0(\theta_n) + \overline{(Q_{n+1} \bullet b_{n+1})}(\theta_0, \dots, \theta_n) \right] \\
 &\quad \left. \times \pi'_n \bar{R}_n^e(\theta_n) \right\}.
 \end{aligned} \tag{7}$$

The minimization problem in (7) shares the same structural form as that in (6). Following a similar reasoning, the optimal policy is derived as

$$\begin{aligned}
 & \pi_n^*(\theta_0, \dots, \theta_n; w_n) \\
 &= - \left[w_n R_n^0(\theta_n) + \frac{\overline{(Q_{n+1} \bullet b_{n+1})}(\theta_0, \dots, \theta_n)}{2 \overline{(Q_{n+1} \bullet a_{n+1})}(\theta_0, \dots, \theta_n)} \right] V_n(\theta_n)^{-1} \bar{R}_n^e(\theta_n).
 \end{aligned}$$

By substituting this optimal policy, (7) simplifies to

$$v_n(\theta_0, \dots, \theta_n; w_n) = a_n(\theta_0, \dots, \theta_n)w_n^2 + b_n(\theta_0, \dots, \theta_n)w_n + c_n(\theta_0, \dots, \theta_n),$$

where

$$\begin{aligned} a_n(\theta_0, \dots, \theta_n) &= f_n(\theta_n) \overline{(Q_{n+1} \bullet a_{n+1})}(\theta_0, \dots, \theta_n), \\ b_n(\theta_0, \dots, \theta_n) &= g_n(\theta_n) \overline{(Q_{n+1} \bullet b_{n+1})}(\theta_0, \dots, \theta_n), \\ c_n(\theta_0, \dots, \theta_n) &= - \frac{((Q_{n+1} \bullet b_{n+1})(\theta_0, \dots, \theta_n))^2}{4(Q_{n+1} \bullet a_{n+1})(\theta_0, \dots, \theta_n)} h_n(\theta_n) \\ &\quad + \overline{(Q_{n+1} \bullet c_{n+1})}(\theta_0, \dots, \theta_n). \end{aligned}$$

Once again, we confirm that $a_n(\theta_0, \dots, \theta_n) > 0$.

The claims of the theorem now follow from Lemma 3, applied to the recursive definitions of a_n , b_n , and c_n . To derive a_n , we define $B_n(\theta_0, \dots, \theta_n) = 0$ and set the terminal condition as $B_T(\theta_0, \dots, \theta_T) = a_T(\theta_0, \dots, \theta_T) = 1$. Similarly, for b_n , we impose $B_n(\theta_0, \dots, \theta_n) = 0$ with the final condition $B_T(\theta_0, \dots, \theta_T) = b_T(\theta_0, \dots, \theta_T) = d_1$. Using these results, we reformulate e_n and the optimal strategy π_n^* . For c_n , we assume $C_n(\theta_n) = 1$. \square

3.2 Solution to problem $P(MV)$

To derive the solution for problem $P(MV)$ under the initial conditions $\Theta_0 = \theta_0$ and $W_0 = w_0$, we reformulate the optimal value function of $\tilde{P}(MV)$, given by

$$v_0(\theta_0; w_0) = a_0(\theta_0)w_0^2 + b_0(\theta_0)w_0 + c_0(\theta_0)$$

in terms of the Lagrange multiplier λ . To achieve this, we apply Lemma 2 to re-express the coefficients derived in Theorem 1, incorporating their dependence on λ . This yields

$$\begin{aligned} c_0(\theta_0) &= e_0(\theta_0) + \sum_{k=1}^{T-1} \overline{\prod_{j=1}^{k-1} Q_{1_j} \bullet (Q_k \bullet e_k)}(\theta_0) + \overline{\prod_{j=1}^{T-1} Q_{1_j} \bullet (Q_T \bullet e_T)}(\theta_0) \\ &= -(d_1^2/4)c_0^*(\theta_0) + d_0, \end{aligned}$$

where

$$c_0^*(\theta_0) = e_0^*(\theta_0) + \sum_{k=1}^{T-1} \overline{\prod_{j=1}^{k-1} Q_{1_j} \bullet (Q_k \bullet e_k^*)(\theta_0)},$$

$$e_n^*(\theta_0, \dots, \theta_n) = \frac{\left[\overline{\prod_{j=n+1}^{T-1} Q_{g_j} \bullet (Q_T \bullet \mathbf{1}_T)(\theta_0, \dots, \theta_n)} \right]^2}{\overline{\prod_{j=n+1}^{T-1} Q_{f_j} \bullet (Q_T \bullet \mathbf{1}_T)(\theta_0, \dots, \theta_n)}} h_n(\theta_n).$$

Moreover, we obtain $b_0(\theta_0) = d_1 b_0^*(\theta_0)$, where

$$b_0^*(\theta_0) = g_0(\theta_0) \overline{\prod_{j=1}^{T-1} Q_{g_j} \bullet (Q_T \bullet \mathbf{1}_T)(\theta_0)}.$$

Thus, the function simplifies to

$$\begin{aligned} v_0(\theta_0; w_0) &= a_0(\theta_0)w_0^2 + b_0(\theta_0)w_0 + c_0(\theta_0) \\ &= a_0(\theta_0)w_0^2 + d_1 b_0^*(\theta_0)w_0 - \frac{d_1^2}{4} c_0^*(\theta_0) + d_0 \\ &= a_0(\theta_0)w_0^2 + 2(\lambda - E_T) b_0^*(\theta_0)w_0 - (\lambda - E_T)^2 c_0^*(\theta_0) + E_T^2 - 2\lambda E_T. \end{aligned} \quad (8)$$

Since $v_0(\theta_0; w_0)$ depends on λ , we define

$$L(\lambda) := v_0(\theta_0; w_0).$$

By the Lagrange duality theorem (see [15]), maximizing (8) over $\lambda \in \mathbb{R}$ provides the optimal value for problem $P(MV)$, denoted by $\mathbb{V}ar_0^*(E_T)$, that is,

$$\mathbb{V}ar_0^*(E_T) = \max_{\lambda \in \mathbb{R}} L(\lambda).$$

Lemma 1 guarantees that $c_0^*(\theta_0)$ remains strictly positive. Consequently, the function $L(\lambda)$ attains its maximum at

$$\lambda^* = \frac{b_0^*(\theta_0)w_0 - E_T}{c_0^*(\theta_0)} + E_T.$$

To derive the optimal portfolio strategy for $P(MV)$, we substitute

$$d_1 = 2(\lambda^* - E_T) = \frac{2(b_0^*(\theta_0)w_0 - E_T)}{c_0^*(\theta_0)}$$

into (5). Additionally, replacing λ^* in (8) yields the minimum variance associated with problem $P(MV)$. These results lead to the following theorem, which provides the solution to the main problem $P(MV)$.

Theorem 2. The optimal variance (or risk) corresponding to problem $P(MV)$ is given by

$$\mathbb{V}\text{ar}_0^*(E_T) = \frac{1 - c_0^*(\theta_0)}{c_0^*(\theta_0)} \left(E_T - \frac{b_0^*(\theta_0)w_0}{1 - c_0^*(\theta_0)} \right)^2 + \left(a_0(\theta_0) - \frac{b_0^*(\theta_0)^2}{1 - c_0^*(\theta_0)} \right) w_0^2. \quad (9)$$

This follows from the optimal portfolio strategy given by

$$\begin{aligned} & \pi_n^*(\theta_0, \dots, \theta_n; w_n) \\ &= - \left[w_n R_n^0(\theta_n) + d^* \frac{\overline{\prod_{j=n+1}^{T-1} Q_{g_j} \bullet (Q_T \bullet \mathbf{1}_T)(\theta_0, \dots, \theta_n)}}{\overline{\prod_{j=n+1}^{T-1} Q_{f_j} \bullet (Q_T \bullet \mathbf{1}_T)(\theta_0, \dots, \theta_n)}} \right] V_n(\theta_n)^{-1} \bar{R}_n^e(\theta_n), \end{aligned} \quad (10)$$

where

$$d^* = \frac{b_0^*(\theta_0)w_0 - E_T}{c_0^*(\theta_0)}. \quad (11)$$

Remark 2. The optimal policy presented in (10) clearly reflects the influence of regime dynamics on asset allocation. Specifically, the allocation decision is shaped by three distinct components: Return parameters such as $R_n^0(\theta_n)$, $V_n(\theta_n)$, and $\bar{R}_n^e(\theta_n)$, which depend only on the current market state; the scalar d^* , which encapsulates initial market conditions and investor targets; and a path-dependent term involving transition tensors, which captures the effect of historical regime evolution. This structure highlights how the policy simultaneously accounts for both the present market regime and the trajectory of past regimes in determining optimal investment actions.

The primary objective in portfolio selection is to determine *efficient* portfolio strategies. A portfolio strategy π^* is deemed efficient if no alternative strategy π exists that yields the same expected terminal wealth with lower risk, or the same risk with a higher expected terminal wealth. An efficient point refers to the ordered pair in the Mean-Variance plane associated with

an efficient portfolio strategy. The efficient frontier is the collection of all such efficient points. It is important to note that while an efficient point corresponds to the solution of a Mean-Variance problem, not the solution of a Mean-Variance problem necessarily represents an efficient point. Specifically, if an optimal portfolio strategy falls on the lower branch of the parabola (9), meaning

$$E_T < \frac{b_0^*(\theta_0)w_0}{1 - c_0^*(\theta_0)},$$

then there exists another optimal strategy with the same variance but a greater expected terminal wealth. Therefore, in the Mean-Variance plane, the efficient frontier corresponds to the upper branch of the parabola (9), where

$$E_T \geq \frac{b_0^*(\theta_0)w_0}{1 - c_0^*(\theta_0)}.$$

This discussion leads to the following theorem, which concludes this section.

Corollary 1. The Mean-Variance efficient frontier is given by (9) for

$$E_T \geq \frac{b_0^*(\theta_0)w_0}{1 - c_0^*(\theta_0)}.$$

The global minimum risk portfolio strategy corresponds to the ordered pair

$$\left(\frac{b_0^*(\theta_0)w_0}{1 - c_0^*(\theta_0)}, \left(a_0(\theta_0) - \frac{b_0^*(\theta_0)^2}{1 - c_0^*(\theta_0)} \right) w_0^2 \right)$$

in the Mean-Variance plane and can be determined using (10) with

$$d^* = \frac{b_0^*(\theta_0)w_0}{c_0^*(\theta_0) - 1}.$$

Remark 3. While our model allows for path-dependent transition probabilities, its computational complexity remains manageable in practice. In real-world applications, these transition probabilities typically depend only on a short and finite memory of recent market states—such as the last two or three time periods—rather than the entire historical path. This finite-memory assumption not only aligns with empirical observations in financial markets, but also significantly reduces the number of relevant paths that need to be considered. Furthermore, the use of tensor-based representations and recur-

sive backward computation contributes to numerical tractability and enables efficient implementation in multi-period settings. If higher-dimensional path dependence is required in specific applications, then the model can also accommodate various approximation methods, such as Monte Carlo simulation, dimension reduction techniques, or path truncation strategies.

4 Markovian regime-switching markets

In this section, we examine the Mean-Variance portfolio selection problem within a standard Markovian regime-switching market, where the transition probability satisfies the Markov property:

$$P(\Theta_n = \theta_n \mid \Theta_{n-1} = \theta_{n-1}, \dots, \Theta_1 = \theta_1, \Theta_0 = \theta_0) = P(\Theta_n = \theta_n \mid \Theta_{n-1} = \theta_{n-1}).$$

This assumption leads to the simplification

$$Q_n(\theta_0, \dots, \theta_{n-1}, \theta_n) = Q_n(\theta_{n-1}, \theta_n),$$

which implies that Q_n can be considered as a standard $M \times M$ matrix.

This classical framework has been previously analyzed by Cakmak and Ozekici [2]. We demonstrate that the findings in [2] emerge as special cases of our generalized results. A key distinction from their work is that we do not impose the assumption of time-homogeneity on the Markov process, and we allow asset returns to be influenced by both the market state and the specific time period.

To simplify the calculations, we adopt a notation consistent with the conventions introduced in Section 2. For an $M \times M$ matrix A and the M -column vector $\mathbf{1} = (1, \dots, 1)'$ we define $\overline{A} = A\mathbf{1}$. Then, for any $M \times M$ matrix B and an M -column vector C the following identities hold:

$$\overline{AB} = AB\mathbf{1} = A\overline{B}, \quad \overline{AC} = AC.$$

Below, we present our results under the assumption of Markovian transition probabilities. To achieve this, we express key parameters and coefficients using Lemma 2 alongside the notations introduced earlier. For instance, we

obtain

$$\begin{aligned}
 & \overline{\prod_{j=n+1}^{T-1} Q_{g_j} \bullet (Q_T \bullet \mathbf{1}_T)(\theta_0, \dots, \theta_n)} \\
 &= \sum_{\theta_{n+1}=1}^M \dots \sum_{\theta_{T-1}=1}^M \sum_{\theta_T=1}^M Q_{n+1}(\theta_0, \dots, \theta_{n+1}) g_{n+1}(\theta_{n+1}) \dots Q_{T-1}(\theta_0, \dots, \theta_{T-1}) \\
 &\quad \times g_{T-1}(\theta_{T-1}) Q_T(\theta_0, \dots, \theta_T) \mathbf{1}_T(\theta_0, \dots, \theta_T) \\
 &= \sum_{\theta_{n+1}=1}^M \dots \sum_{\theta_{T-1}=1}^M \sum_{\theta_T=1}^M Q_{n+1}(\theta_n, \theta_{n+1}) g_{n+1}(\theta_{n+1}) \dots Q_{T-1}(\theta_{T-2}, \theta_{T-1}) \\
 &\quad \times g_{T-1}(\theta_{T-1}) Q_T(\theta_{T-1}, \theta_T) \\
 &= \sum_{\theta_{n+1}=1}^M \dots \sum_{\theta_{T-1}=1}^M Q_{g_{n+1}}(\theta_n, \theta_{n+1}) \dots Q_{g_{T-1}}(\theta_{T-2}, \theta_{T-1}) \sum_{\theta_T=1}^M Q_T(\theta_{T-1}, \theta_T) \\
 &= \sum_{\theta_{n+1}=1}^M \dots \sum_{\theta_{T-1}=1}^M Q_{g_{n+1}}(\theta_n, \theta_{n+1}) \dots Q_{g_{T-1}}(\theta_{T-2}, \theta_{T-1}) \mathbf{1}(\theta_{T-1}) \\
 &= \left(\left(\prod_{j=n+1}^{T-1} Q_{g_j} \right) \mathbf{1} \right) (\theta_n) \\
 &= \overline{\left(\prod_{j=n+1}^{T-1} Q_{g_j} \right) (\theta_n)}.
 \end{aligned} \tag{12}$$

Here, the notation \prod represents standard matrix multiplication. Following a similar procedure, we obtain

$$\overline{\prod_{j=n+1}^{T-1} Q_{f_j} \bullet (Q_T \bullet \mathbf{1}_T)(\theta_0, \dots, \theta_n)} = \overline{\left(\prod_{j=n+1}^{T-1} Q_{f_j} \right) (\theta_n)}.$$

It follows that these parameters depend solely on θ_n . Consequently, e_n^* simplifies to

$$e_n^*(\theta_n) = \frac{\left[\overline{\left(\prod_{j=n+1}^{T-1} Q_{g_j} \right) (\theta_n)} \right]^2}{\overline{\left(\prod_{j=n+1}^{T-1} Q_{f_j} \right) (\theta_n)}} h_n(\theta_n).$$

Applying the same approach as in (12), we obtain

$$\overline{\prod_{j=n+1}^{k-1} Q_{1_j} \bullet (Q_k \bullet e_k^*)}(\theta_0, \dots, \theta_n) = \left(\left(\prod_{j=n+1}^k Q_j \right) e_k^* \right) (\theta_n).$$

Finally, the key parameters are expressed as

$$\begin{aligned} a_0(\theta_0) &= f_0(\theta_0) \left[\overline{\left(\prod_{j=1}^{T-1} Q_{f_j} \right)}(\theta_0) \right], \\ b_0^*(\theta_0) &= g_0(\theta_0) \left[\overline{\left(\prod_{j=1}^{T-1} Q_{g_j} \right)}(\theta_0) \right], \\ c_0^*(\theta_0) &= e_0^*(\theta_0) + \sum_{k=1}^{T-1} \left(\overline{\left(\prod_{j=1}^k Q_j \right) e_k^*} \right) (\theta_0) = \sum_{k=0}^{T-1} \left(\overline{\left(\prod_{j=1}^k Q_j \right) e_k^*} \right) (\theta_0). \end{aligned}$$

By replacing these expressions in (10) and (11), the optimal portfolio strategy simplifies to

$$\begin{aligned} \pi_n^*(\theta_0, \dots, \theta_n; w_n) &= \pi_n^*(\theta_n; w_n) = \\ &= \left[w_n R_n^0(\theta_n) + d^* \frac{\overline{\left(\prod_{j=n+1}^{T-1} Q_{g_j} \right)}(\theta_n)}{\overline{\left(\prod_{j=n+1}^{T-1} Q_{f_j} \right)}(\theta_n)} \right] V_n(\theta_n)^{-1} \bar{R}_n^e(\theta_n). \end{aligned}$$

This confirms that the optimal portfolios depend only on θ_n .

In a more constrained setting, Cakmak and Ozekici [2] studied a market with risky assets and a riskless asset, assuming that asset returns depend only on the market state, not on the time period, within a time-homogeneous Markov chain. In other words, their parameters are time-independent. A straightforward manipulation shows that, under time-independent parameters, the results presented in [2, Corollary 5] match our results in Theorem 2 and Corollary 1.

5 Numerical illustration

Consider a regime-switching market model with two states: A bull state (State 1) and a bear state (State 2). The model includes two assets: A risky asset, whose returns follow a log-normal distribution, and a risk-free asset,

whose returns are deterministic and vary with the market state. Specifically, if R^1 denotes the return of the risky asset, then $\ln R^1$ follows a normal distribution with mean μ and variance σ^2 , where μ and σ^2 depend solely on the prevailing market state. In the bull state, the risky asset exhibits positive rate of returns, reflecting favorable market conditions. In contrast, during the bear state, the risky asset experiences negative rate of returns, reflecting adverse market conditions and heightened volatility. The risk-free asset provides stable but slightly reduced returns (denoted by R^0) in the bear state compared to the bull state. The parameters for both market states are summarized in Table 1.

Table 1: Market parameters in bull and bear states

| θ_n | $\mu(\theta_n)$ | $\sigma^2(\theta_n)$ | $\mathbb{E}[R_n^1(\theta_n)]$ | $\mathbb{V}\text{ar}[R_n^1(\theta_n)]$ | $R_n^0(\theta_n)$ |
|------------|-------------------------------|----------------------|-------------------------------|--|-------------------|
| 1 | 0.020 | 0.015 | 1.0279 | 0.016 | 1.005 |
| 2 | -0.030 | 0.045 | 0.9925 | 0.0453 | 1.003 |
| θ_n | $\mathbb{E}[R_n^e(\theta_n)]$ | $V_n(\theta_n)$ | $h_n(\theta_n)$ | $g_n(\theta_n)$ | $f_n(\theta_n)$ |
| 1 | 0.0229 | 0.0165 | 0.0317 | 0.9731 | 0.978 |
| 2 | -0.0105 | 0.0454 | 0.0024 | 1.0006 | 1.0036 |

Let the investment horizon be $T = 3$. The transition matrices, as shown in Table 2, are constructed for each time period, conditional on the historical path, ensuring that the model accurately reflects the dynamic nature of market regimes, where

$$\begin{aligned}
 P_0(\theta_0, \theta_1) &= Q_1(\theta_0, \theta_1), \\
 P_1^{(\theta_0)}(\theta_1, \theta_2) &= Q_2(\theta_0, \theta_1, \theta_2), \\
 P_2^{(\theta_0, \theta_1)}(\theta_2, \theta_3) &= Q_3(\theta_0, \theta_1, \theta_2, \theta_3).
 \end{aligned}$$

For example, if at time $n = 1$, the market is currently in the bull state ($\Theta_1 = 1$) and has been in the bull state in the previous period ($\Theta_0 = 1$), then the probability of remaining in the bull state is 80%, while the probability of transitioning to the bear state is 20% (see $P_1^{(1)}(1, 1)$ and $P_1^{(1)}(1, 2)$ in Table 2). However, if at time $n = 2$ the market is currently in the bull state ($\Theta_2 = 1$) and has been in the bull state for the previous two periods ($\Theta_0 = 1, \Theta_1 = 1$), then the probability of remaining in the bull state is 90%, while the probability of transitioning to the bear state is 10% (see

$P_2^{(1,1)}(1,1)$ and $P_2^{(1,1)}(1,2)$ in Table 2). Here, $P_2^{(1,1)}(1,1) > P_1^{(1)}(1,1)$ but $P_2^{(1,1)}(1,2) < P_1^{(1)}(1,2)$.

Table 2: Path-dependent transition matrices

| Time | Path | Transition Matrix |
|---------|------------------------------|--|
| $n = 0$ | — | $P_0 = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$ |
| $n = 1$ | $\Theta_0 = 1$ | $P_1^{(1)} = \begin{bmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{bmatrix}$ |
| $n = 1$ | $\Theta_0 = 2$ | $P_1^{(2)} = \begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}$ |
| $n = 2$ | $\Theta_0 = 1, \Theta_1 = 1$ | $P_2^{(1,1)} = \begin{bmatrix} 0.9 & 0.1 \\ 0.6 & 0.4 \end{bmatrix}$ |
| $n = 2$ | $\Theta_0 = 1, \Theta_1 = 2$ | $P_2^{(1,2)} = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$ |
| $n = 2$ | $\Theta_0 = 2, \Theta_1 = 1$ | $P_2^{(2,1)} = \begin{bmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{bmatrix}$ |
| $n = 2$ | $\Theta_0 = 2, \Theta_1 = 2$ | $P_2^{(2,2)} = \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix}$ |

Let the initial wealth be 100, and assume that the market starts in state 1 (bull state). Values of

$$\hat{g}_n(\theta_0, \dots, \theta_n) := \overline{\prod_{j=n+1}^2 Q_{g_j} \bullet (Q_3 \bullet \mathbf{1}_3)}(\theta_0, \dots, \theta_n),$$

$$\hat{f}_n(\theta_0, \dots, \theta_n) := \overline{\prod_{j=n+1}^2 Q_{f_j} \bullet (Q_3 \bullet \mathbf{1}_3)}(\theta_0, \dots, \theta_n)$$

are given in Table 3. Moreover, we obtain the following initial parameters:

$$a_0(1) = 0.95, \quad b_0^*(1) = 0.937, \quad c_0^*(1) = 0.076.$$

In the following, we compare the optimal investment strategies under a path-dependent model and a Markovian model to achieve a fixed expected terminal wealth of $E_3 = 105$ for some different market paths. M-V efficient frontiers are also compared under two different models. For the Markovian model, we set P_0 as the transition matrix.

Table 3: Values of \hat{g}_n and \hat{f}_n .

| Time | Path | $\hat{g}_n(\theta_0, \dots, \theta_n)$ | $\hat{f}_n(\theta_0, \dots, \theta_n)$ |
|---------|--|--|--|
| $n = 0$ | $\Theta_0 = 1$ | 0.9628 | 0.9713 |
| $n = 1$ | $\Theta_0 = 1, \Theta_1 = 1$ | 0.9786 | 0.9831 |
| $n = 1$ | $\Theta_0 = 1, \Theta_1 = 2$ | 0.9868 | 0.9908 |
| $n = 2$ | $\Theta_0 = 1, \Theta_1 = 1, \Theta_2 = 1$ | 1 | 1 |
| $n = 2$ | $\Theta_0 = 1, \Theta_1 = 1, \Theta_2 = 2$ | 1 | 1 |
| $n = 2$ | $\Theta_0 = 1, \Theta_1 = 2, \Theta_2 = 1$ | 1 | 1 |
| $n = 2$ | $\Theta_0 = 1, \Theta_1 = 2, \Theta_2 = 2$ | 1 | 1 |

For the first market path $(1, 1, 1)$ (see Figure 1) with corresponding wealth levels $(100, 101, 102)$, the optimal investment in the risky asset under the path-dependent model is $(65.5, 64.97, 64.52)$, while under the Markovian model, it is $(68.39, 67.87, 67.39)$. The higher investment in the risky asset under the Markovian model stems from its lower expected return compared to the path-dependent model. Specifically, the path-dependent model incorporates momentum effects, increasing the probability of remaining in the bull state and thus raising the expected return. This allows for a more conservative investment strategy to achieve the fixed expected terminal wealth. In contrast, the Markovian model, which ignores historical paths, yields a lower expected return, necessitating a more aggressive (higher) investment strategy to meet the same target.

For the second market path $(1, 2, 2)$ (see Figure 1) with corresponding wealth levels $(100, 101, 99)$, the optimal investment in the risky asset under the path-dependent model is $(65.5, -10.85, -11.45)$, while under the Markovian model, it is $(68.39, -11.34, -11.93)$. The negative investments in the risky asset under unfavorable market conditions (e.g., state 2, bear state) reflect risk-averse behavior, where investors reduce exposure or take short positions to mitigate potential losses. However, the magnitude of these short positions differs between the two models due to their distinct assumptions. In the path-dependent model, the expected decline in the value of the risky asset is greater, as the model accounts for the persistence of unfavorable market conditions, leading to a higher likelihood of continued losses. Consequently, a smaller short position is required to achieve the fixed expected terminal

wealth, as the model anticipates and adjusts for the larger expected decline. In contrast, the Markovian model predicts a smaller expected decline in the value of the risky asset, as it ignores historical paths. Thus, a larger short position is necessary to achieve the same target, as the model underestimates the persistence of unfavorable conditions and associated risks. This highlights how the path-dependent model's incorporation of historical market behavior enables more precise adjustments in investment strategies.

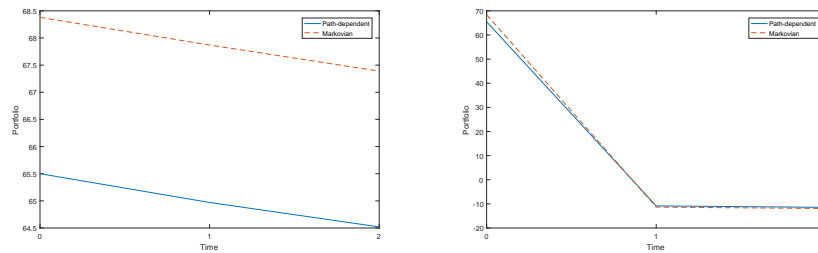


Figure 1: Optimal portfolio strategies for the market paths $(1, 1, 1)$ (left) and $(1, 2, 2)$ (right).

We also compared and plotted the mean-variance efficient frontiers for the path-dependent and Markovian models. The efficient frontier for the path-dependent model lies below that of the Markovian model, indicating that for a fixed level of expected terminal wealth, the optimal investment risk (i.e., the variance of terminal wealth) is lower in the path-dependent model compared to the Markovian model. This can be attributed to the observed behavior in the above two market paths: The absolute value of the investment in the risky asset is generally smaller in the path-dependent model than in the Markovian model. This reduction in exposure to the risky asset likely contributes to lower overall risk. As demonstrated in the optimal strategies examples, the path-dependent model's incorporation of historical market behavior leads to more conservative investment strategies, which in turn reduce risk. This comparison of the efficient frontiers is illustrated in Figure 2. The end point $(101.4, 0.0216)$ represents the global minimum risk portfolio strategy.

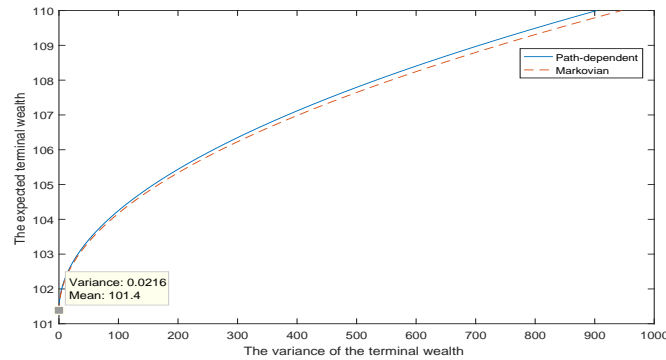


Figure 2: The M-V efficient frontier.

6 Conclusion

In this paper, we have investigated a multi-period mean-variance portfolio optimization problem under a non-Markovian regime-switching model. The asset returns in this market depend on market states that evolve stochastically over time among a finite set of possible states, with transition probabilities that are path-dependent rather than Markovian. To solve this optimization problem, we employed dynamic programming combined with an auxiliary problem approach, necessitated by the non-separability introduced by the variance operator in the dynamic programming framework. The solution methodology combines dynamic programming with Lagrangian multiplier method, utilizing an extended Bellman equation based on path-dependent value functions. The optimal policy parameters are obtained implicitly through a system of path-dependent backward recursive relations. Explicit closed-form solutions are derived using matrix computations, and the optimal strategy for the original problem is recovered via Lagrangian duality theorem.

Our results demonstrated that the optimal investment strategy exhibits path-dependence at each time point. Notably, we showed that the traditional Markovian model emerges as a special case of our framework, where the optimal strategy depends only on the current state. Furthermore, we establish that the efficient frontier—characterizing the relationship between

expected final wealth and optimal risk—is significantly influenced by our modeling assumptions. In particular, the path-dependent model leads to risk reduction compared to the Markovian benchmark, achieved through more conservative allocations to risky assets. This reduction stems from the model’s ability to incorporate historical market behavior, enabling finer adjustments in investment strategies.

For future research, we suggest two promising directions: (1) Extending the model to incorporate path-dependent asset returns, and (2) investigating time-consistent formulations under these path-dependent assumptions. These extensions could provide even more realistic tools for portfolio management in regime-switching environments.

References

- [1] Bian, L. and Zhang, L. *Equilibrium multi-period investment strategy for a DC pension plan with incomplete information: Hidden Markov model*, Comm. Statist. Theory Methods 49 (2025), 1702–1728.
- [2] Cakmak, U. and Ozekici, S. *Portfolio optimization in stochastic markets*, Math. Methods Oper. Res. 63 (2006), 151–168.
- [3] Canakoglu, E. and Ozekici, S. *Portfolio selection in stochastic markets with exponential utility functions*, Ann. Oper. Res. 166 (2009), 281–297.
- [4] Canakoglu, E. and Ozekici, S. *Portfolio selection in stochastic markets with HARA utility functions*, Eur. J. Oper. Res. 201 (2010), 520–536.
- [5] Canakoglu, E. and Ozekici, S. *Portfolio selection with imperfect information: A hidden Markov model*, Appl. Stoch. Models Bus. Ind. 27 (2011), 95–114.
- [6] Celikyurt, U. and Ozekici, S. *Multiperiod portfolio optimization models in stochastic markets using the mean-variance approach*, Eur. J. Oper. Res. 179 (2007), 186–202.

- [7] Chen, T., Liu, R. and Wu, Z. *Continuous-time mean-variance portfolio selection under non-Markovian regime-switching model with random horizon*, J. Syst. Sci. Complex. 36 (2023), 457–479.
- [8] Chen, P. and Yang, H. *Markowitz's mean-variance asset-liability management with regime switching: A multi-period model*, Appl. Math. Finance, 18 (2011), 29–50.
- [9] Costa, O.L.V. and Araujo, M.V. *A generalized multi-period mean-variance portfolio optimization with Markov switching parameters*, Automatica, 44 (2008), 2487–2497.
- [10] Dokuchaev, N. *Discrete time market with serial correlations and optimal myopic strategies*, European J. Oper. Res. 177 (2007), 1090–1104.
- [11] Ge, H., Li, X., Li, X. and Li, Z. *Equilibrium strategy for a multi-period weighted mean-variance portfolio selection in a Markov regime-switching market with uncertain time-horizon and a stochastic cash flow*, Comm. Statist. Theory Methods, 52 (2023), 1797–1832.
- [12] Keykhaei, R. *Portfolio optimization under regime-switching with market path-dependent returns*, J. Math. Model. (2025), 485–496. doi: 10.22124/jmm.2025.28912.2571
- [13] Li, D. and Ng, W.L. *Optimal dynamic portfolio selection: multiperiod mean-variance formulation*, Math. Finance, 10 (2000), 387–406.
- [14] Li, X., Zhou, X.Y. and Lim, A.E.B. *Dynamic mean-variance portfolio selection with no-shorting constraints*, SIAM J. Control Optim. 40 (2002), 1540–1555.
- [15] Luenberger, D.G. *Optimization by vector space methods*, Wiley, New York, 1968.
- [16] Markowitz, H. *Portfolio selection*, J. Finance, 7 (1952), 77–91.
- [17] Shen, Y., Wei, J. and Zhao, Q. *Mean-variance asset-liability management problem under non-Markovian regime-switching mode*, Appl. Math. Optim. 81 (2020), 859–897.

- [18] Sun, Z. *Mean-Variance Asset-Liability Management in a Non-Markovian Regime-Switching Jump-Diffusion Market with Random Horizon*, Appl. Math. Optim. 84 (2021), 319–353.
- [19] Wang, T., Jin, Z. and Wei, J. *Mean-variance portfolio selection under a non-Markovian regime-switching model: Time-consistent solutions*, SIAM J. Control Optim. 57 (2019), 3249–3271.
- [20] Wang, T. and Wei, J. *Mean-variance portfolio selection under a non-Markovian regime-switching model*, J. Comput. Appl. Math. 350 (2019), 442–455.
- [21] Wei, S.Z. and Ye, Z.X. *Multi-period optimization portfolio with bankruptcy control in stochastic market*, Appl. Math. Comput. 186 (2007), 414–425.
- [22] Wu, H. and Chen, H. *Nash equilibrium strategy for a multi-period mean-variance portfolio selection problem with regime switching*, Econ. Model. 46 (2015), 79–90.
- [23] Xiao, H., Zhou, Z., Ren, T., Bai, Y. and Liu, W. *Time-consistent strategies for multi-period mean-variance portfolio optimization with the serially correlated returns*, Comm. Statist. Theory Methods, 49 (2020), 2831–2868.
- [24] Xu, Y. and Li, Z.F. *Dynamic portfolio selection based on serially correlated return-dynamic mean-variance formulation*, Syst. Eng. Theory Pract. 18 (2008), 123–131.
- [25] Yao, H., Li, D. and Wu, H. *Dynamic trading with uncertain exit time and transaction costs in a general Markov market*, Int. Rev. Financial Anal. 84 (2022), 102371.
- [26] Zhang, L. and Li, Z. *Multi-period mean-variance portfolio selection with uncertain time horizon when returns are serially correlated*, Math. Probl. Eng. 2012 (2012), 216891.
- [27] Zhang, L., Li, Z., Xu, Y. and Li, Y. *Multi-period mean variance portfolio selection under incomplete information*, Appl. Stoch. Models Bus. Ind. 32 (2016), 753–774.

- [28] Zhang, L., Zhang, H. and Yao, H. *Optimal investment management for a defined contribution pension fund under imperfect information*, Insur. Math. Econ. 79 (2018), 210–224.
- [29] Zhu, D.M., Lu, J., Ching, W.K. and Siu, T.K. *Discrete-time optimal asset allocation under higher-order hidden Markov model*, Econ. Model. 66 (2017), 223–232.



Approximation of functions in Hölder's class and solution of nonlinear Lane–Emden differential equation by orthonormal Euler wavelets

H.C. Yadav*, A. Yadav and S. Lal

Abstract

In this article, a method has been developed to solve a nonlinear Lane–Emden differential equation based on the orthonormal Euler wavelet series. The orthonormal Euler wavelets are constructed by the dilatation and translation of orthogonal Euler polynomials. The convergence analysis of

*Corresponding author

Received 1 February 2025; revised 11 July 2025; accepted 1 August 2025

Harish Chandra Yadav

Department of Mathematics, School of Basic Sciences, Galgotias University, Greater Noida, India. e-mail: harishchandrayadav20395@gmail.com

Abhilasha Yadav

Department of Mathematics, Institute of Integrated and Honors Studies, Kurukshetra University, Kurukshetra, India. e-mail: yadavabhilasha1942@kuk.ac.in

Shyam Lal

Department of Mathematics, Institute of Science, Banaras Hindu University, Varanasi, India. e-mail: shyam_lal@rediffmail.com

How to cite this article

Yadav, H.C., Yadav, A. and Lal, S., Approximation of functions in Hölder's class and solution of nonlinear Lane–Emden differential equation by orthonormal Euler wavelets. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1688–1709. <https://doi.org/10.22067/ijnao.2025.91960.1593>

the orthonormal Euler wavelet series is studied in the Hölder's class. The orthonormal Euler wavelet approximations of solution functions of the nonlinear Lane–Emden differential equation in Hölder's class are determined by partial sums of their orthonormal Euler wavelet series. In concisely, two approximations $E_{2^{k-1},M}^{(1)}(f)$ and $E_{2^{k-1},M}^{(2)}(f)$ of solution functions of classes $H_2^\alpha[0,1)$ and $H_2^\phi[0,1)$ by $(2^k, M)^{th}$ partial sums of their orthonormal Euler wavelet expansions have been estimated. There are several applications of nonlinear differential equations, which include the nonlinear Lane–Emden differential equations. The solution of the nonlinear Lane–Emden differential equation obtained by the orthonormal Euler wavelets method is compared to its solution obtained by the Euler method and the ODE-45 method. It has been shown that the solutions produced by the orthonormal Euler wavelets are more accurate than those produced by the Euler method and the ODE-45 method. This is a result of the wavelet analysis research article.

AMS subject classifications (2020): Primary 34A34; Secondary 42C40, 65T60, 65L05.

Keywords: Orthonormal Euler wavelet, $H_2^\alpha[0,1)$ and $H_2^\phi[0,1)$ class, Approximation of function and nonlinear Lane–Emden differential equations.

1 Introduction

The wavelet theory has acquired a lot of applications in recent times. Wavelets naturally adapt to irregular domains, simplifying computations and improving stability. There are various wavelet methods proposed for the approximation of functions and numerical solution of differential and integral equations, such as Legendre, Chebyshev, Gegenbauer, Genocchi, Vieta–Lucas, Euler, and sine-cosine wavelets. With the help of the orthogonal basis of those wavelets, it is possible to reduce the numerical problems of differential and integral equations to a system of linear and nonlinear algebraic equations. Many researchers like Chui [4], Debnath [5], Doha, Abd-Elhameed, and Youssri [6], Lal and Kumar [10], Lal and Patel [11], Meyer [14], and so on, are working in the direction of approximation of functions and solution of differential or integral equations. The present work introduces orthonormal

Euler wavelets (OEWs) as a novel basis for solving such equations. The key contributions include the construction of OEWs by combining Euler polynomials with wavelet theory, enabling efficient representation of solutions in Hölder spaces (Polat and Dincel [17]). Since the OEWs are generated by orthonormal Euler polynomials, the orthonormal Euler polynomials have fewer terms than other polynomials for generating their wavelets.

Various natural phenomena are studied and described using differential and integral equations [2, 3, 7, 8, 9, 1]. The nonlinear Lane–Emden differential equation is a fundamental model in astrophysics, describing phenomena such as stellar structure, isothermal gas spheres, and thermionic currents. Traditional numerical methods (e.g., finite difference, Runge–Kutta) often struggle with singularity at the origin and nonlinearity, leading to reduced accuracy and stability. To the best of our knowledge, there is no work related to the approximation of solution functions of the nonlinear Lane–Emden differential equation belonging to Hölder’s class $H_2^\alpha[0, 1)$ and $H_2^\phi[0, 1)$ by the OEW expansion (Titchmarsh [16]). In Hölder’s class $H_2^\alpha[0, 1)$ and $H_2^\phi[0, 1)$, the convergence analysis of the solution function f of the nonlinear Lane–Emden differential equation by the OEW series has been investigated. A method of the collocation has been proposed to find the numerical solution of the nonlinear Lane–Emden differential equations by the OEWs. Unlike other numerical methods, the collocation method easily transforms the differential equations into algebraic equations and can achieve high accuracy with relatively few collocation points. Rigorous convergence analysis in Hölder’s class ensures that the wavelet approximations of solution functions are better and provide good results, demonstrating the method’s effectiveness in handling singular and nonlinear terms.

The objective of this research paper are as follows:

- (i) To define the Hölder’s class $H_2^\alpha[0, 1)$ and $H_2^\phi[0, 1)$ in the interval $[0, 1)$.
- (ii) To define the orthonormal Euler polynomial and the OEW in the interval $[0, 1)$.
- (iii) To derive the approximation of the solution function f of the nonlinear Lane–Emden differential equations belonging to classes $H_2^\alpha[0, 1)$ and $H_2^\phi[0, 1)$.
- (iv) To describe the procedure for calculating the numerical solution of the

nonlinear Lane–Emden differential equations and to provide examples to show the effectiveness of this procedure.

(v) To compare the exact solution of the nonlinear Lane–Emden differential equation using OEWs, the Euler method (EM), and the ODE-45 method.

The remaining parts of this paper are categorized as follows: In Section 2, some definitions and properties of the Hölder's class, the orthonormal Euler polynomial, and the OEW are mentioned. In Section 3, the convergence analysis of the solution function f of the nonlinear Lane–Emden differential equation by OEW series has been investigated. In Section 4, the definition of the OEW approximation of the solution function f of the nonlinear Lane–Emden differential equation and two estimators by OEW approximations and their proofs in $H_2^\alpha[0, 1)$ and $H_2^\phi[0, 1)$ class have been developed. In Section 5, the algorithm for the solution of the nonlinear Lane–Emden differential equation has been developed in the interval $[0, 1)$, which is used to obtain the solution of the nonlinear Lane–Emden differential equation by the OEW. In Section 6, the solutions of the nonlinear Lane–Emden differential equation by OEWs, the EM, and the ODE45 method, and their absolute error have been obtained. Section 7 is designated for the conclusions of this research paper.

2 Definitions and preliminaries

2.1 Function of Hölder's class $H_2^\alpha[0, 1)$

A function f belongs to $H_2^\alpha[0, 1)$, $\alpha \in (0, 1]$, if f is continuous and satisfies the following condition:

$$\left(\int_0^1 (f(x+t) - f(x))^2 dx \right)^{\frac{1}{2}} = O(|t|^\alpha), \quad \text{for all } x, t, x+t \in [0, 1).$$

2.2 Function of Hölder's class $H_2^\phi[0, 1]$

Let $\phi(t)$ be positive monotonic increasing function of t such that $\phi(|t|) \rightarrow 0$ as $t \rightarrow 0$. A function f belongs to $H_2^\phi[0, 1]$ if f is continuous and satisfies the following condition:

$$\left(\int_0^1 (f(x+t) - f(x))^2 dx \right)^{\frac{1}{2}} = O(\phi(|t|)), \quad \text{for all } x, t, x+t \in [0, 1].$$

If $\phi(t) = t^\alpha$, then $H_2^\phi[0, 1]$ coincides with classical Hölder's class $H_2^\alpha[0, 1]$.

2.3 Orthonormal Euler polynomial and OEW

The orthonormal Euler polynomials of order m is denoted by $E_m^{(O)}(t)$ and defined in the interval $[0, 1]$ as

$$E_m^{(O)}(t) = \sqrt{2m+1} \sum_{k=0}^m (-1)^{m-k} \binom{m}{k} \binom{m+k}{k} t^k, \quad m \in \{0, 1, 2, 3, \dots\}. \quad (1)$$

The orthonormality property for Euler polynomials is as follows:

$$\langle E_m^{(O)}, E_n^{(O)} \rangle = \begin{cases} 1, & \text{if } n = m, m = m'; \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

By analyzing the integrals of these polynomials from 0 to t , we obtain

$$\int_0^t E_0^{(O)}(x) dx = \frac{1}{2} E_0^{(O)}(t) + \frac{1}{2\sqrt{3}} E_1^{(O)}(t), \quad (3)$$

and for $m \geq 1$,

$$\int_0^t E_m^{(O)}(x) dx = \frac{E_{m+1}^{(O)}(t)}{2\sqrt{(2m+1)(2m+3)}} - \frac{E_{m-1}^{(O)}(t)}{2\sqrt{(2m+1)(2m-1)}}. \quad (4)$$

Therefore,

$$2\sqrt{2m+1}E_m^{(O)}(t) = \frac{1}{\sqrt{2m+3}}(E_{m+1}^{(O)}(t))' - \frac{1}{\sqrt{2m-1}}(E_{m-1}^{(O)}(t))'. \quad (5)$$

The OEWS denoted by $\psi_{n,m}^{(O)}$, are defined on $[0, 1)$ by

$$\psi_{n,m}^{(O)}(t) = \begin{cases} 2^{\frac{k-1}{2}} E_m^{(O)}(2^{k+2}t - 4n + 2), & \text{if } t \in [\frac{n-1}{2^{k-1}}, \frac{n}{2^{k-1}}); \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where $n = 1, 2, 3, \dots, 2^{k-1}$, $m = 0, 1, 2, \dots, M-1$, m is the order of the orthonormal Euler polynomials, and $k = 1, 2, 3, \dots$ is the level of resolution.

3 Convergence analysis of the OEWS series

In this section, the convergence analysis of the solution function f , of the nonlinear Lane–Emden differential equations, in $L^2[0, 1)$ by the OEWS expansion has been described.

Theorem 1. If $f(t)$ is the exact solution of the nonlinear Lane–Emden differential equation, then its OEWS series $\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} c_{n,m} \psi_{n,m}^{(O)}(t)$ converges uniformly to $f(t)$.

Proof.

$$\begin{aligned} \text{Let } f(t) &= \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} c_{n,m} \psi_{n,m}^{(O)}(t). \\ \text{Then } \langle f, f \rangle &= \left\langle \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} c_{n,m} \psi_{n,m}^{(O)}(t), \sum_{n'=1}^{2^{k-1}} \sum_{m'=0}^{\infty} c_{n',m'} \psi_{n',m'}^{(O)}(t) \right\rangle \\ &= \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} \sum_{n'=1}^{2^{k-1}} \sum_{m'=0}^{\infty} c_{n,m} \overline{c_{n',m'}} \langle \psi_{n,m}^{(O)}, \psi_{n',m'}^{(O)} \rangle \\ &= \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} |c_{n,m}|^2 \|\psi_{n,m}^{(O)}\|_2^2 \\ &= \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} |c_{n,m}|^2, \\ &\quad \{\psi_{n,m}^{(O)}\} \text{ is an orthonormal basis of } L^2[0, 1). \end{aligned}$$

$$\text{Thus, } \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} |c_{n,m}|^2 = \langle f, f \rangle = \int_0^1 |f(t)|^2 dt < \infty, \quad f \in L^2[0, 1).$$

Therefore, the wavelet series $\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{\infty} c_{n,m} \psi_{n,m}^{(O)}(t)$ is convergent and by the Bessel's inequality, $\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} |c_{n,m}|^2 \leq \|f\|_2^2 < \infty$, for all $M > 2$.

For $N > M$ & $k > p$,

$$\begin{aligned} \text{let } (S_{2^{k-1}, M} f)(t) &= \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{n,m}^{(O)}(t). \\ \|(S_{2^{k-1}, N} f) - (S_{2^{p-1}, M} f)\|_2^2 &= \left\| \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{N-1} c_{n,m} \psi_{n,m}^{(O)}(t) - \sum_{n=1}^{2^{p-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{n,m}^{(O)}(t) \right\|_2^2 \\ &= \left\| \sum_{n=2^{(p-1)}+1}^{2^{k-1}} \sum_{m=M}^{N-1} c_{n,m} \psi_{n,m}^{(O)}(t) \right\|_2^2 \\ &= \sum_{n=2^{p-1}+1}^{2^{k-1}} \sum_{m=M}^{N-1} |c_{n,m}|^2 \rightarrow 0 \text{ as } M \rightarrow \infty, N \rightarrow \infty. \end{aligned}$$

Therefore, $\|(S_{2^{k-1}, N} f) - (S_{2^{p-1}, M} f)\|_2^2 \rightarrow 0$ as $M \rightarrow \infty, N \rightarrow \infty$. Hence, $(S_{2^{k-1}, N} f)_{N=0}^{\infty}$ is a Cauchy sequence in $L^2[0, 1)$. Since, $L^2[0, 1)$ is a Banach space, the Cauchy sequence $(S_{2^{k-1}, N} f)_{N=0}^{\infty}$ converges to a function $b(t)$, (say). Here, $b(t) = \lim_{N \rightarrow \infty} (S_{2^{k-1}, N} f) = \lim_{N \rightarrow \infty} \sum_{n=1}^{2^k} \sum_{m=0}^{N-1} c_{n,m} \psi_{n,m}^{(O)}(t)$.

Now, we need to show that $b(t) = f(t)$. For this, consider

$$\begin{aligned} \langle b(t) - f(t), \psi_{n,m}^{(O)}(t) \rangle &= \langle b(t), \psi_{n,m}^{(O)}(t) \rangle - \langle f(t), \psi_{n,m}^{(O)}(t) \rangle \\ &= \lim_{N \rightarrow \infty} \langle (S_{2^{k-1}, N} f), \psi_{n,m}^{(O)}(t) \rangle - c_{n,m} \\ &= c_{n,m} - c_{n,m} = 0. \end{aligned}$$

Therefore, $b(t) = f(t)$ for all $t \in [0, 1)$. Hence, the OEW series $\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{N-1} c_{n,m} \psi_{n,m}^{(O)}(t)$ converges uniformly to $f(t)$ as $N \rightarrow \infty$. \square

4 Approximations and theorems

In this section, approximation and theorems based on the OEW have been established.

4.1 OEW approximation

Since, $\{\psi_{n,m}^{(O)}(t)\}$ forms an orthonormal basis for $L^2[0,1)$, a function $f \in L^2[0,1)$ can be expressed into the OEW series as

$$f(t) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} c_{n,m} \psi_{n,m}^{(O)}(t), \quad c_{n,m} = \langle f, \psi_{n,m}^{(O)} \rangle. \quad (7)$$

The $(2^{k-1}, M)$ th partial sum $(S_{2^{k-1},M}f)(t)$ of the OEW series (7) is given by

$$(S_{2^{k-1},M}f)(t) = \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{n,m}^{(O)}(t) = C^T \psi^{(O)}(t), \quad (8)$$

where $C = [c_{1,0}, c_{1,1}, \dots, c_{1,M-1}; c_{2,0}, \dots, c_{2,M-1}, \dots, c_{2^{k-1},0}; \dots, c_{2^{k-1},M-1}]^T$ and

$$\psi^{(O)}(t) = [\psi_{1,0}^{(O)}(t), \dots, \psi_{1,M-1}^{(O)}(t); \psi_{2,0}^{(O)}(t), \dots; \psi_{2^{k-1},0}^{(O)}(t), \dots, \psi_{2^{k-1},M-1}^{(O)}(t)]^T.$$

The OEW approximation $E_{2^{k-1},M}(f)$ of f by $(2^{k-1}, M)$ th partial sum $(S_{2^{k-1},M}f)$ of the OEW series (7), is defined by

$$E_{2^{k-1},M}(f) = \min_{(S_{2^{k-1},M}f)} \|f - (S_{2^{k-1},M}f)\|_2. \quad (9)$$

Here, $E_{2^{k-1},M}(f)$ is said to be the best approximation of f by $(2^{k-1}, M)$ th partial sum $(S_{2^{k-1},M}f)$, if $E_{2^{k-1},M}(f) \rightarrow 0$, as $k \rightarrow \infty$, $M \rightarrow \infty$ (Zygmund [18]).

4.2 Theorems

In this paper, the following theorems based on the OEW in the Hölder's class have been developed.

Theorem 2. If $f'' \in H_2^\alpha[0, 1]$ class, then the OEW expansion of the solution function f of the nonlinear Lane–Emden differential equation is $f(t) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} c_{n,m} \psi_{n,m}^{(O)}(t)$ having $(2^{k-1}, M)^{\text{th}}$ partial sums,

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

then, the OEW approximation of f by $(S_{2^{k-1}, M} f)$, under $\|\cdot\|_2$, for $M > 2$ is given by

$$E_{2^{k-1}, M}^{(1)}(f) = \min \|f - \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{n,m}^{(O)}(t)\|_2 = O\left(\frac{1}{2^{(k-1)(\alpha+2)}(2M-3)^{\frac{3}{2}}}\right).$$

Proof. Consider

$$\begin{aligned} f(t) &= \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} c_{n,m} \psi_{n,m}^{(O)}(t). \\ c_{n,m} &= \left\langle f, \psi_{n,m}^{(O)} \right\rangle \\ &= \int_{\frac{n-1}{2^{k-1}}}^{\frac{n}{2^{k-1}}} f(t) \psi_{n,m}^{(O)}(t) dt \\ &= 2^{\frac{k-1}{2}} \int_{\frac{n-1}{2^{k-1}}}^{\frac{n}{2^{k-1}}} f(t) E_m^{(O)}(2^{k-1}t - n + 1) dt \\ &= 2^{\frac{k-1}{2}} \int_0^1 f\left(\frac{u+n-1}{2^{k-1}}\right) E_m^{(O)}(u) \frac{du}{2^{k-1}}, \quad 2^{k-1}t - n + 1 = u \\ &= \frac{1}{2^{\frac{k+1}{2}} \sqrt{2m+1}} \int_0^1 f\left(\frac{u+n-1}{2^{k-1}}\right) d\left(\frac{E_{m+1}^{(O)}(t)}{\sqrt{2m+3}} - \frac{E_{m-1}^{(O)}(t)}{\sqrt{2m-1}}\right) \quad (10) \\ &\quad (\text{by (5)}) \end{aligned}$$

Integrating (10) by parts, we have

$$\begin{aligned}
c_{n,m} &= -\frac{1}{2^{\frac{3k-1}{2}}\sqrt{2m+1}} \int_0^1 f' \left(\frac{u+n-1}{2^{k-1}} \right) \left(\frac{E_{m+1}^{(O)}(t)}{\sqrt{2m+3}} - \frac{E_{m-1}^{(O)}(t)}{\sqrt{2m-1}} \right) dt \\
&= -\frac{1}{2^{\frac{3k-1}{2}}\sqrt{2m+1}} \int_0^1 f' \left(\frac{u+n-1}{2^{k-1}} \right) d \left(\frac{E_{m+2}^{(O)}(t)}{2(2m+3)\sqrt{2m+5}} \right. \\
&\quad \left. - \frac{\sqrt{2m+1}E_m^{(O)}(t)}{(2m+3)(2m-1)} + \frac{E_{m-2}^{(O)}(t)}{2(2m-1)\sqrt{2m+1}} \right), \text{ by (5)} \quad (11)
\end{aligned}$$

Integrating (11) by parts, we have

$$\begin{aligned}
c_{n,m} &= \frac{1}{2^{\frac{5k-3}{2}}\sqrt{2m+1}} \int_0^1 f'' \left(\frac{u+n-1}{2^{k-1}} \right) \\
&\quad \times \left(\frac{E_{m+2}^{(O)}(t)}{(2m+3)\sqrt{2m+5}} - \frac{\sqrt{2m+1}E_m^{(O)}(t)}{2(2m+3)(2m-1)} + \frac{E_{m-2}^{(O)}(t)}{2(2m-1)\sqrt{2m+1}} \right) dt \\
&= \frac{1}{2^{\frac{5k-3}{2}}\sqrt{2m+1}} \int_0^1 \left(f'' \left(\frac{u+n-1}{2^{k-1}} \right) - f'' \left(\frac{n-1}{2^{k-1}} \right) \right) \\
&\quad \times \left(\frac{E_{m+2}^{(O)}(t)}{(2m+3)\sqrt{2m+5}} - \frac{\sqrt{2m+1}E_m^{(O)}(t)}{2(2m+3)(2m-1)} + \frac{E_{m-2}^{(O)}(t)}{2(2m-1)\sqrt{2m+1}} \right) dt \\
&\quad + \frac{1}{2^{\frac{5k-3}{2}}\sqrt{2m+1}} f'' \left(\frac{n-1}{2^{k-1}} \right) \int_0^1 \left(\frac{E_{m+2}^{(O)}(t)}{(2m+3)\sqrt{2m+5}} \right. \\
&\quad \left. - \frac{\sqrt{2m+1}E_m^{(O)}(t)}{2(2m+3)(2m-1)} + \frac{E_{m-2}^{(O)}(t)}{2(2m-1)\sqrt{2m+1}} \right) dt \\
&\leq \frac{1}{2^{\frac{5k-3}{2}}\sqrt{2m+1}} \left(\int_0^1 \left(f'' \left(\frac{u+n-1}{2^{k-1}} \right) - f'' \left(\frac{n-1}{2^{k-1}} \right) \right)^2 \right)^{\frac{1}{2}} \\
&\quad \times \left(\int_0^1 \left(\frac{E_{m+2}^{(O)}(t)}{(2m+3)\sqrt{2m+5}} - \frac{\sqrt{2m+1}E_m^{(O)}(t)}{2(2m+3)(2m-1)} \right. \right. \\
&\quad \left. \left. + \frac{E_{m-2}^{(O)}(t)}{2(2m-1)\sqrt{2m+1}} \right)^2 dt \right)^{\frac{1}{2}} \\
&\leq \frac{r}{2^{\frac{5k-3}{2}}\sqrt{2m+1}} \left(\frac{1}{2^{k-1}} \right)^\alpha \left(\frac{1}{(2m+3)^2(2m+5)} \right. \\
&\quad \left. - \frac{2m+1}{4(2m+3)^2(2m-1)^2} + \frac{1}{4(2m-1)^2(2m+1)} \right)^{\frac{1}{2}} \\
&\quad (\text{by (2) and } f'' \in H_2^\alpha[0,1), \text{ } r \text{ be a positive constant})
\end{aligned}$$

$$|c_{n,m}| \leq \frac{r}{2^{(k-1)(\alpha+\frac{5}{2})}(2m-3)^2}. \quad (12)$$

$$\begin{aligned} \text{Then, } (E_{2^{k-1},M}^{(1)}(f))^2 &= \|f(t) - (S_{2^{k-1},M}f)(t)\|_2^2 = \sum_{n=1}^{2^{k-1}} \sum_{m=M}^{\infty} |c_{n,m}|^2 \\ &\leq \sum_{n=1}^{2^{k-1}} \sum_{m=M}^{\infty} \left(\frac{r}{2^{(k-1)(\alpha+\frac{5}{2})}(2m-3)^2} \right)^2 \\ &= \frac{r^2 2^{k-1}}{2^{(k-1)(2\alpha+5)}} \sum_{m=M}^{\infty} \frac{1}{(2m-3)^4} \\ &\leq \frac{r^2 2^{k-1}}{2^{(k-1)(2\alpha+5)}} \left(\frac{1}{(2M-3)^4} + \int_M^{\infty} \frac{dm}{(2m-3)^4} \right) \\ &\quad (\text{by Cauchy's integral test}) \\ &\leq \frac{r^2}{2^{(k-1)(2\alpha+4)}} \left(\frac{7}{6(2M-3)^3} \right). \\ E_{2^{k-1},M}^{(1)}(f) &\leq \frac{r\sqrt{7}}{2^{(k-1)(\alpha+2)}\sqrt{6}(2M-3)^{\frac{3}{2}}}. \\ \text{Therefore, } E_{2^{k-1},M}^{(1)}(f) &= O\left(\frac{1}{2^{(k-1)(\alpha+2)}(2M-3)^{\frac{3}{2}}}\right), \quad M > 2. \end{aligned}$$

□

Theorem 3. If $f'' \in H_2^\phi[0, 1)$ class, such that $\phi(|t|) \rightarrow 0$ as $t \rightarrow 0$, then the OEW approximation of f by $(S_{2^{k-1},M}f)$ satisfies

$$\begin{aligned} E_{2^{k-1},M}^{(2)}(f) &= \min \|f - \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c'_{n,m} \psi_{n,m}^{(O)}(t)\|_2 \\ &= O\left(\frac{1}{2^{(k-1)}(2M-3)^{\frac{3}{2}}} \phi\left(\frac{1}{2^{k-1}}\right)\right), \quad M > 2. \end{aligned}$$

Proof. Following the proof of Theorem 2 and for $f'' \in H^\phi[0, 1)$ class, we have

$$\begin{aligned} c'_{n,m} &\leq \frac{1}{2^{\frac{5k-3}{2}}\sqrt{2m+1}} \left(\int_0^1 \left(f''\left(\frac{u+n-1}{2^{k-1}}\right) \right. \right. \\ &\quad \left. \left. - f''\left(\frac{n-1}{2^{k-1}}\right) \right)^2 \right)^{\frac{1}{2}} \left(\int_0^1 \left(\frac{E_{m+2}^{(O)}(t)}{(2m+3)\sqrt{2m+5}} \right) \right. \end{aligned}$$

$$\begin{aligned}
& - \frac{\sqrt{2m+1}E_m^{(O)}(t)}{2(2m+3)(2m-1)} + \frac{E_{m-2}^{(O)}(t)}{2(2m-1)\sqrt{2m+1}} \Big)^2 dt \Big)^{\frac{1}{2}} \\
& \leq \frac{q}{2^{\frac{5k-5}{2}}(2m-3)^2} \phi\left(\frac{1}{2^{k-1}}\right) \\
& \quad (\text{by (2) and } f'' \in H_2^\phi[0, 1))
\end{aligned} \tag{13}$$

$$\begin{aligned}
\text{Then, } (E_{2^{k-1},M}^{(2)}(f))^2 &= \sum_{n=1}^{2^{k-1}} \sum_{m=M}^{\infty} |c'_{n,m}|^2 \\
&\leq \sum_{n=1}^{2^{k-1}} \sum_{m=M}^{\infty} \left(\frac{q}{2^{\frac{5k-5}{2}}(2m-3)^2} \phi\left(\frac{1}{2^{k-1}}\right) \right)^2 \\
&= \frac{q^2 2^{k-1}}{2^{5(k-1)}} \phi^2\left(\frac{1}{2^{k-1}}\right) \sum_{m=M}^{\infty} \frac{1}{(2m-3)^4} \\
&\leq \frac{q^2 2^{k-1}}{2^{5(k+1)}} \phi^2\left(\frac{1}{2^{k-1}}\right) \left(\frac{7}{6(2M-3)^3}\right).
\end{aligned}$$

$$\text{Therefore, } E_{2^k,M}^{(2)}(f) = O\left(\frac{1}{2^{(k-1)}(2M-3)^{\frac{3}{2}}} \phi\left(\frac{1}{2^{k-1}}\right)\right), \quad M > 2.$$

This completes the proof of the Theorem 3. \square

5 Algorithm for the nonlinear Lane–Emden differential equation

This section contains the procedure for solving the nonlinear Lane–Emden differential equations by the OEW. The five basis functions of the OEW for $k = 1$ and $M = 5$ are as follows:

$$\left. \begin{aligned} \psi_{1,0}^{(O)}(t) &= 1 \\ \psi_{1,1}^{(O)}(t) &= \sqrt{3}(2t-1) \\ \psi_{1,2}^{(O)}(t) &= \sqrt{5}(6t^2-6t+1) \\ \psi_{1,3}^{(O)}(t) &= \sqrt{7}(20t^3-30t^2+12t-1) \\ \psi_{1,4}^{(O)}(t) &= \sqrt{9}(70t^4-140t^3+90t^2-20t+1) \end{aligned} \right\}, \quad t \in [0, 1). \tag{14}$$

Let $y(t)$ be the solution of the nonlinear Lane–Emden differential equation:

$$y'' + \frac{2}{t}y' + y^\gamma = 0, \gamma \geq 2, t \in [0, 1], y(0) = 1, y'(0) = 0 \text{ (Mukherjee et al. [15])}. \quad (15)$$

$$\text{Then, } y(t) = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} c_{n,m} \psi_{n,m}^{(O)}(t) \quad (16)$$

and $(2^{k-1}, M)^{th}$ partial sum of series (16) is

$$y(t) = (S_{2^{k-1}, M} f)(t) = \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{n,m}^{(O)}(t) = C^T \psi^{(O)}(t). \quad (17)$$

By initial conditions (15), (17) reduces to

$$y(0) = \sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{n,m}^{(O)}(0) = 1, y'(0) = \frac{d}{dt} \left(\sum_{n=1}^{2^{k-1}} \sum_{m=0}^{M-1} c_{n,m} \psi_{n,m}^{(O)}(t) \right)_{t=0} = 0.$$

In (17), C^T contains $2^{k-1}M$ unknown coefficients. Hence, excluding initial conditions of (15), $2^{k-1}M - 2$ extra conditions are needed for the solution of the nonlinear Lane–Emden differential equation (15). For determining the values of $2^{k-1}M$ unknown coefficients $c_{n,m}$, collocation points $t_i = \frac{i-1}{2^{k-1}M}$, $i = 2, \dots, 2^{k-1}M$, are substituted in (17) to obtain $2^{k-1}M - 2$ system of algebraic equations. Hence, the values of unknown coefficients $c_{n,m}$ are obtained by solving these $2^{k-1}M$ system of algebraic equations. This algorithm is also applicable to the solution of higher-order linear and nonlinear differential equations (Lal and Yadav [12], Yogit, Scindia, and Kumar [13]).

6 Results and discussion

The applicability of the suggested approach for numerical solution of the nonlinear Lane–Emden differential equation and its error analysis has been covered in this section. The solutions obtained are also compared in the suggested way, the EM, and the ODE-45 method with their exact solutions for $\gamma = 5$.

Example 1. Consider the nonlinear Lane–Emden differential equation for $\gamma = 2$

$$y'' + \frac{2}{t}y' + y^2 = 0, y(0) = 1, y'(0) = 0. \quad (18)$$

By the algorithm of the OEW approach described in section 5, the nonlinear Lane–Emden differential equations have been solved. For the approximate solution of (18), take $M = 5$ and $k = 1$. Then $y(t)$ will be

$$\begin{aligned} y(t) &= \sum_{m=0}^4 c_{1,m} \psi_{1,m}^{(O)}(t) \\ &= c_{1,0} + \sqrt{3}(2t-1)c_{1,1} + \sqrt{5}(6t^2-6t+1)c_{1,2} + \sqrt{7}(20t^3-30t^2 \\ &\quad + 12t-1)c_{1,3} + \sqrt{9}(70t^4-140t^3+90t^2-20t+1)c_{1,4}, \quad t \in [0, 1]. \end{aligned} \quad (19)$$

For calculating the unknown values $c_{1,0}$, $c_{1,1}$, $c_{1,2}$, $c_{1,3}$, and $c_{1,4}$, we collocate (18) and (19) at $t = 0.25, 0.5, 0.75$ and using the initial condition in (19), system of five equations is obtained. Solving this system of equations, the values of the unknowns are as follows:

$$\begin{aligned} c_{1,0} &= 0.947558069231224, \quad c_{1,1} = -0.044519188708113, \\ c_{1,2} &= -0.010463291917928, \quad c_{1,3} = 0.000547078923993, \\ c_{1,4} &= 0.000058833597300. \end{aligned} \quad (20)$$

Putting the values of $c_{1,0}$, $c_{1,1}$, $c_{1,2}$, $c_{1,3}$, and $c_{1,4}$ from (20) into (19), we obtain

$$\begin{aligned} y(t) &= 0.947558069231224 - 0.044519188708113\sqrt{3}(2t-1) \\ &\quad - 0.010463291917928\sqrt{5}(6t^2-6t+1) + 0.000547078923993\sqrt{7}(20t^3 \\ &\quad - 30t^2 + 12t-1) + 0.0000588335973\sqrt{9}(70t^4-140t^3+90t^2-20t+1). \end{aligned}$$

The approximate solution of the nonlinear Lane–Emden differential equation (18) is obtained by the OEW method for $M = 5, 10$, and 15 in the interval $[0, 1]$. Also, this solution has been compared with the ODE-45 method and the EM is shown in Table 1.

The graph between the ODE-45, EM and OEW of solution of the nonlinear Lane–Emden differential equation (18) is shown in Figure 1.

Example 2. Consider the nonlinear Lane–Emden differential equation for $\gamma = 4$ as

$$y'' + \frac{2}{t}y' + y^4 = 0, \quad y(0) = 1, \quad y'(0) = 0. \quad (21)$$

Table 1: Comparison between ODE-45, OEW and EM for Example 1

| t | ODE-45 | OEW ($M = 5$) | OEW ($M = 10$) | OEW ($M = 15$) | EM ($M = 10$) | EM ($M = 15$) |
|-----|--------------|--------------------|---------------------|---------------------|--------------------|--------------------|
| 0.1 | 0.9983349952 | 0.9983262964 | 0.9983349986 | 0.9983349985 | 0.9985010028 | 0.9984456377 |
| 0.2 | 0.9933599136 | 0.9933369662 | 0.9933599072 | 0.9933599071 | 0.9936874525 | 0.9935781792 |
| 0.3 | 0.9851339505 | 0.9851019189 | 0.9851339470 | 0.9851339469 | 0.9856141179 | 0.9854538721 |
| 0.4 | 0.9737541183 | 0.9737207165 | 0.9737541164 | 0.9737541163 | 0.9743738441 | 0.9741669514 |
| 0.5 | 0.9593527169 | 0.9593225730 | 0.9593527158 | 0.9593527158 | 0.9600952381 | 0.9598472632 |
| 0.6 | 0.9420940363 | 0.9420663544 | 0.9420940358 | 0.9420940352 | 0.9429394773 | 0.9426570314 |
| 0.7 | 0.9221703488 | 0.9221405789 | 0.9221703486 | 0.9221703485 | 0.9230963905 | 0.9227869095 |
| 0.8 | 0.8997973703 | 0.8997634170 | 0.8997973703 | 0.8997973702 | 0.9007799757 | 0.9004514781 |
| 0.9 | 0.8752093703 | 0.8751826912 | 0.8752093704 | 0.8752093703 | 0.8762235317 | 0.8758843691 |

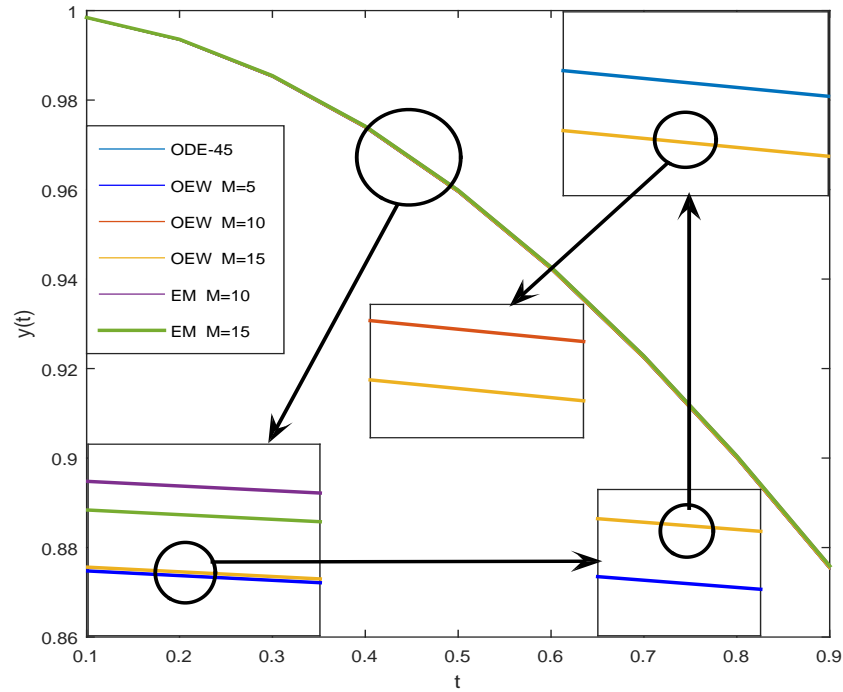


Figure 1: The graphs between ODE-45, OEW and EM for Example 1

The approximate solution of the nonlinear Lane-Emden differential equation (21) obtained by OEW is given in Table 2, for $M = 5, 10$, and 15 in the interval $[0, 1)$. Also, a comparison among this solution, the ODE-45, and EM is shown in Table 2.

The graph between ODE-45, EM and OEW solutions of the nonlinear Lane-Emden differential equation (21) is shown in Figure 2.

Table 2: Comparison between ODE-45, OEW and EM for Example 2

| t | ODE-45 | OEW ($M = 5$) | OEW ($M = 10$) | OEW ($M = 15$) | EM ($M = 10$) | EM ($M = 15$) |
|-----|--------------|--------------------|---------------------|---------------------|--------------------|--------------------|
| 0.1 | 0.9983366618 | 0.9983025751 | 0.9983366543 | 0.9983366595 | 0.9985020041 | 0.9984468288 |
| 0.2 | 0.9933862156 | 0.9932972736 | 0.9933862071 | 0.9933862135 | 0.9937080741 | 0.9936006083 |
| 0.3 | 0.9852648948 | 0.9851421997 | 0.9852648879 | 0.9852648944 | 0.9857260484 | 0.9855719731 |
| 0.4 | 0.9741584084 | 0.9740323168 | 0.9741584026 | 0.9741584089 | 0.9747346018 | 0.9745419648 |
| 0.5 | 0.9603109012 | 0.9601994473 | 0.9603108961 | 0.9603109023 | 0.9609727347 | 0.9607513270 |
| 0.6 | 0.9440112896 | 0.9439122728 | 0.9440112841 | 0.9440112908 | 0.9447263563 | 0.9444869990 |
| 0.7 | 0.9255783519 | 0.9254763341 | 0.9255783457 | 0.9255783526 | 0.9263133809 | 0.9260672071 |
| 0.8 | 0.9053459236 | 0.9052340309 | 0.9053459167 | 0.9053459238 | 0.9060687187 | 0.9058265206 |
| 0.9 | 0.8836493241 | 0.8835646224 | 0.8836493175 | 0.8836493239 | 0.8843303463 | 0.8841020405 |

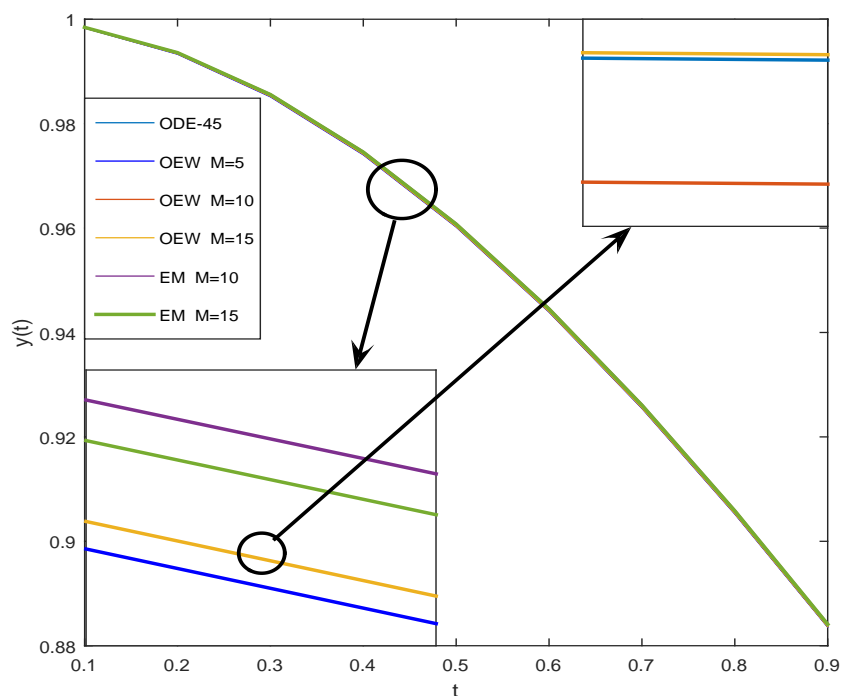


Figure 2: The graphs between ODE-45, OEW and EM for Example 2

Example 3. Consider the nonlinear Lane–Emden differential equation for $\gamma = 5$ as

$$y'' + \frac{2}{t}y' + y^5 = 0, \quad y(0) = 1, \quad y'(0) = 0. \quad (22)$$

The exact solution of (22) is $y(t) = \frac{1}{\sqrt{1+t^2/3}}$.
Consider $y''(x+t) - y''(x) = \beta_2|t|$.

By Lagrange's mean value theorem, $|y'''(c_1)| \leq \beta_1$, $c_1 \in (0, 1)$.

$$\left(\int_0^1 |y''(x+t) - y''(x)|^2 dx \right)^{\frac{1}{2}} \leq \beta_1 \left(\int_0^1 |t|^2 dx \right)^{\frac{1}{2}} = O(|t|^\alpha), \quad \alpha \in (0, 1].$$

Hence, $y''(t) \in H_2^\alpha[0, 1]$.

The approximate solution of the nonlinear Lane–Emden differential equation (22) obtained by OEW for $M = 10$, and 15 in the interval $[0, 1]$ is given in Table 3. This solution has also been compared to the exact solution (ES), the solution obtained by the ODE-45 and EM is shown in Table 3.

Table 3: Comparison between ES, ODE-45, OEW and EM for Example 3

| t | ES | ODE-45 | OEW ($M = 10$) | OEW ($M = 15$) | EM ($M = 10$) | EM ($M = 15$) |
|-----|--------------|--------------|---------------------|---------------------|--------------------|--------------------|
| 0.1 | 0.9983374884 | 0.9983374920 | 0.9983374705 | 0.9983374884 | 0.9985025042 | 0.9984474235 |
| 0.2 | 0.9933992677 | 0.9933992691 | 0.9933992459 | 0.9933992677 | 0.9937183239 | 0.9936117508 |
| 0.3 | 0.9853292781 | 0.9853292778 | 0.9853292561 | 0.9853292781 | 0.9857812146 | 0.9856301345 |
| 0.4 | 0.9743547036 | 0.9743547022 | 0.9743546823 | 0.9743547036 | 0.9749103063 | 0.9747244246 |
| 0.5 | 0.9607689228 | 0.9607689208 | 0.9607689022 | 0.9607689228 | 0.9613937542 | 0.9611845569 |
| 0.6 | 0.9449111825 | 0.9449111809 | 0.9449111607 | 0.9449111825 | 0.9455686216 | 0.9453483586 |
| 0.7 | 0.9271455408 | 0.9271455401 | 0.9271455187 | 0.9271455408 | 0.9277996158 | 0.9275803477 |
| 0.8 | 0.9078412990 | 0.9078412992 | 0.9078412772 | 0.9078412990 | 0.9084590067 | 0.9082518193 |
| 0.9 | 0.8873565094 | 0.8873565100 | 0.8873564897 | 0.8873565094 | 0.8879094562 | 0.8877239019 |

The graph between exact solution, ODE-45, EM and OEW solutions of the nonlinear Lane–Emden differential equation (22) is shown in Figure 3.

By Table 3 and Figure 3, it is clear that the exact and the OEW solutions of the nonlinear Lane–Emden equation (22) coincide almost everywhere for $M = 15$.

6.1 Absolute error

The absolute error in the approximate solution of the nonlinear Lane–Emden differential equation (22) by the ODE-45 OEW method and the EM is given in Table 4. The absolute error is negligible in the solution obtained by the OEW method for $M = 15$.

The graphs of the absolute error in the solution of the nonlinear Lane–Emden differential equation (22) by the OEW method for $M = 10, 15$ and ODE-45 method are shown in Figure 4.

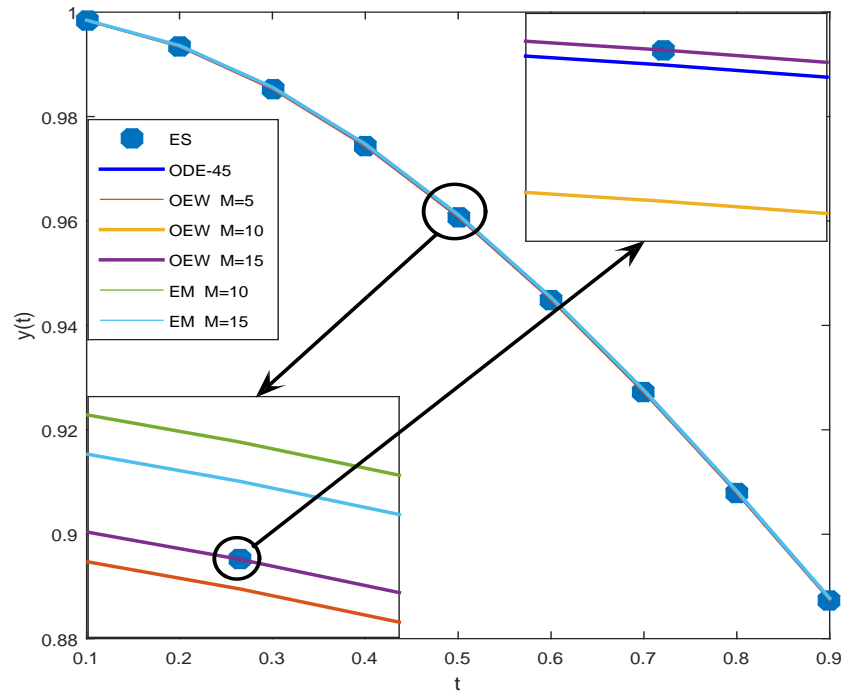


Figure 3: The graphs between ES, ODE-45, OEW and EM for Example 3

Table 4: Absolute error between ES, ODE-45, OEW and EM of Example 3

| t | ODE-45 ($\times 10^{-8}$) | OEW $M=5$ ($\times 10^{-3}$) | OEW $M=10$ ($\times 10^{-7}$) | OEW $M=15$ ($\times 10^{-11}$) | EM $M=10$ ($\times 10^{-3}$) | EM $M=15$ ($\times 10^{-3}$) |
|-----|--------------------------------|-----------------------------------|------------------------------------|-------------------------------------|-----------------------------------|-----------------------------------|
| 0.1 | 0.3610796328 | 0.0495610203 | 0.1794897497 | 0.2246203223 | 0.1650157999 | 0.1099351000 |
| 0.2 | 0.1358930190 | 0.1286525500 | 0.2188676739 | 0.2629008122 | 0.3190562000 | 0.2124830999 |
| 0.3 | 0.0307614156 | 0.1764263294 | 0.2203716298 | 0.2814748434 | 0.4519364999 | 0.3008563999 |
| 0.4 | 0.0257128496 | 0.1800378634 | 0.2136527910 | 0.2774225293 | 0.5556026999 | 0.3697209999 |
| 0.5 | 0.1931504844 | 0.1575967430 | 0.2060257175 | 0.2659761300 | 0.6248313999 | 0.4156341000 |
| 0.6 | 0.1528948967 | 0.1378408315 | 0.2174169055 | 0.2488009798 | 0.6574391000 | 0.4371761000 |
| 0.7 | 0.0634760466 | 0.1390788755 | 0.2205584404 | 0.2272737553 | 0.6540750000 | 0.4348068999 |
| 0.8 | 0.0205243488 | 0.1496226164 | 0.2179250213 | 0.2095879025 | 0.6177077000 | 0.4105202999 |
| 0.9 | 0.0616085071 | 0.1112946509 | 0.1962926043 | 0.1538325022 | 0.5529467999 | 0.3673924999 |

6.2 Physical interpretation

The solution of the Lane–Emden equation represents the dimensionless density profile of a polytropic stellar model. These solutions predict the stability conditions for polytropic stars, influencing theories of stellar evolution. The Lane–Emden equation has a singularity at $t = 0$, but wavelets resolve this

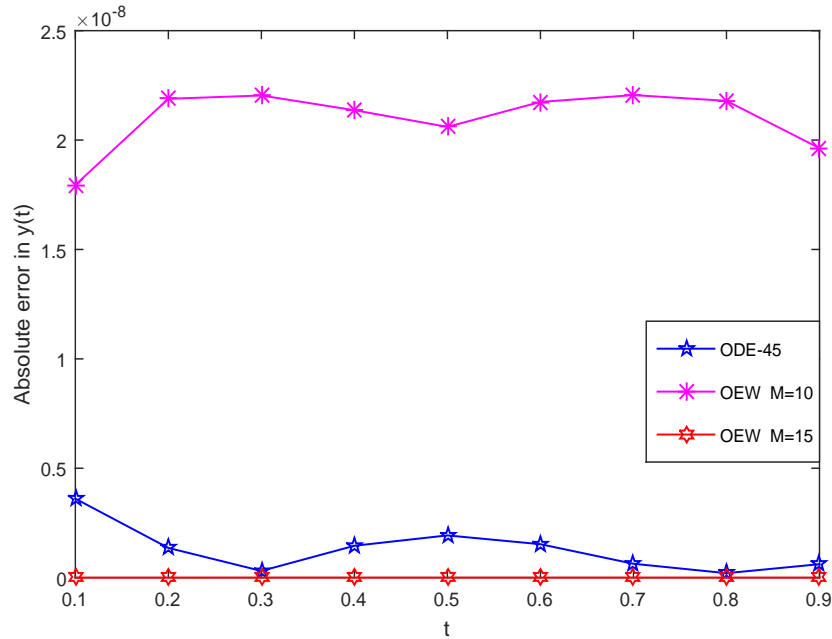


Figure 4: The graphs of the absolute error between ES, ODE-45, OEW and EM for Example 3

singularity by adjusting local resolution, leading to more stable and physically meaningful solutions as shown in the above examples.

7 Conclusion

1. The OEW approximation of solution functions of the nonlinear Lane–Emden differential equation of Theorems 2 and 3 is given by

$$E_{2^{k-1},M}^{(1)}(f) = O\left(\frac{1}{2^{(k-1)(\alpha+2)}(2M-3)^{\frac{3}{2}}}\right), \quad M > 2. \quad (E_{2^{k-1},M}^{(1)}(f)) \rightarrow 0 \text{ as } k \rightarrow \infty \text{ and } M \rightarrow \infty;$$

$$E_{2^{k-1},M}^{(2)}(f) = O\left(\frac{1}{2^{(k-1)(2M-3)^{\frac{3}{2}}}}\phi\left(\frac{1}{2^{k-1}}\right)\right), \quad M > 2. \quad (E_{2^{k-1},M}^{(2)}(f)) \rightarrow 0 \text{ as } k \rightarrow \infty \text{ and } M \rightarrow \infty.$$

Therefore, the approximations $E_{2^{k-1},M}^{(1)}(f)$ and $E_{2^{k-1},M}^{(2)}(f)$ of the solution functions of the nonlinear Lane–Emden differential equation belonging to the classes $H_2^\alpha[0,1)$ and $H_2^\phi[0,1)$ are the best possible in wavelet analysis.

2. By Tables 1, 2, and 3 and Figures 1, 2, and 3, it is shown that the OEW solutions for $M = 15$ of the nonlinear Lane–Emden differential equation coincide almost everywhere in the interval $[0, 1)$.
3. It is clear from Table 4 and Figure 4 that the OEW approach for $M = 15$ has significantly lower absolute error than the ODE-45 method and EM.
4. The examples provided illustrate the applicability and precision of the algorithm in Section 6 for solving the nonlinear Lane–Emden differential equation.
5. Future research directions using this approach can be generalized to solve the higher-order nonlinear differential equations and extended to multi-dimensional linear Partial differential equations.

Acknowledgements

The authors thank to the reviewers for their valuable suggestions which improve this manuscript.

References

- [1] Alsalami, Z. *Modeling of Optimal Fully Connected Deep Neural Network based Sentiment Analysis on Social Networking Data*, J. Smart Internet Things. 2023(2) (2023), 114–132.
- [2] Al-Shetwi, A. and Sujod, M. *Modeling and simulation of photovoltaic module with enhanced perturb and observe mppt algorithm using MATLAB/Simulink*, ARPN J. Eng. Appl. Sci. 11 (2016), 12033–12038.
- [3] Bouchaala, F., Ali, M., Matsushima, J., Jouini, M., Mohamed, A. and Nizamudin, S. *Experimental study of seismic wave attenuation in carbonate rocks*, SPE Journal, 29 (2024), 1–15.
- [4] Chui, C.K. *An introduction to Wavelets (Wavelet Analysis and its Applications)*, Academic Press Cambridge, 1992.
- [5] Debnath, L. *Wavelet transforms and their applications*, Birkhäuser, Boston, 2002.

- [6] Doha, E.H., Abd-Elhameed, W.M. and Youssri, Y.H. *New ultraspherical wavelets collocation method for solving 2nd-order initial and boundary value problems*, J. Egypt. Math. Soc. 24(2) (2016), 319–327.
- [7] Kharnoob, M.M., Carbajal, N.C., Chenet Zuta, M.E., Ali, E., Abdullaev, S.S., Alawadi, A.H.R., Zearah, S.A., Alsalamy, A. and Saxena, A. *Thermoelastic damping in asymmetric vibrations of nonlocal circular plate resonators with Moore-Gibson-Thompson heat conduction*, Proc. Inst. Mech. Eng. Pt. C J. Mechan. Eng. Sci. 238(24) (2024), 11264–11281.
- [8] Kharnoob, M.M., Carbajal, N.C., Chenet Zuta, M.E., Ali, E., Abdullaev, S.S., Alawadi, A.H.R., Zearah, S.A., Alsalamy, A. and Saxena, A. *Analysis of thermoelastic damping in a microbeam following a modified strain gradient theory and the Moore-Gibson-Thompson heat equation*, Mech Time-Depend Mat. 28 (2024), 2367–2393.
- [9] Kharnoob, M.M., Hasan, F.F., Sharma, M.K., Zearah, S.A., Alsalamy, A., Alawadi, A.H.R. and Thabit, D. *Dynamics of spinning axially graded porous nanoscale beams with rectangular cross-section incorporating rotary inertia effects*, J. Vib. Control. 30 (2023), 5358–5374.
- [10] Lal, S. and Kumar, S. *CAS wavelet approximation of functions of Hölder's class $H^\alpha[0, 1)$ and Solution of Fredholm Integral Equations*, Ratio Math. 39 (2020), 187–212.
- [11] Lal, S. and Patel, N. *Chebyshev wavelet approximation of functions having first derivative of Hölder's class*, São Paulo J. Math. Sci. 16 (2022), 1355–1381.
- [12] Lal, S. and Yadav, H.C. *Approximation of functions belonging to Hölder's class and solution of Lane–Emden differential equation using Gegenbauer wavelets*, Filomat, 37(12) (2022), 4029–4045.
- [13] Mahatekar, Y., Scindia, P.S. and Kumar, P. *A new numerical method to solve fractional differential equations in terms of Caputo-Fabrizio derivatives*, Phys. Scr. 98(2) (2023) 024001..

- [14] Meyer, Y. and Roques, S. *Wavelets their past and their future*, *Progress in Wavelet Analysis and Applications* (Toulouse,1992), Frontiers, Gif-sur-Yvette, 1993.
- [15] Mukherjee, S., Roy, B. and Chatterjee, P.K. *Solution of Lane–Emden equation by differential transform method*, *Int. J. Nonlinear Sci.* 12(4) (2011), 478–484.
- [16] Titchmarsh, E.C. *The theory of functions*, (2nd edn.), Oxford University Press, Oxford, 1939.
- [17] Tural Polat, S.N. and Turan Dincel, A. *Euler wavelet method as a numerical approach for the solution of nonlinear systems of fractional differential equations*, *Fractal Fract.* 7(3) (2023), 246.
- [18] Zygmund, A. *Trigonometric series*, Cambridge University Press, Cambridge, 1959.



Mathematical modeling of an optimal control problem for combined chemotherapy and anti-angiogenic cancer treatment protocols

Y.A. Mahaman Nouri* and S. Bisso

Abstract

We formulate and analyze an optimal control problem for combined chemotherapy and anti-angiogenic therapy. The model couples tumor burden, vascular support, and a logistic surrogate for healthy tissue that encodes homeostasis and drug-induced depletion. The cost functional balances tumor reduction and drug sparing with toxicity mitigation: Beyond terminal terms and quadratic control regularization, it includes a trajectory reward for healthy tissue and a smooth, differentiable below-threshold

*Corresponding author

Received 13 April 2025; revised 15 August 2025; accepted 26 September 2025

Yahaya Alassane Mahaman Nouri

Department of Fundamental Sciences, National School of Engineering and Energy Sciences, University of Agadez, Niger. e-mail: alessanenouri@yahoo.fr

Saley Bisso

Department of Mathematics and Computer Science, Faculty of Science and Technology, Abdou Moumouni University, Niger. e-mail: bsaley@yahoo.fr

How to cite this article

Mahaman Nouri, Y.A. and Bisso, S., Mathematical modeling of an optimal control problem for combined chemotherapy and anti-angiogenic cancer treatment protocols. *Iran. J. Numer. Anal. Optim.*, 2025; 15(4): 1710–1729. <https://doi.org/10.22067/ijnao.2025.92904.1628>

penalty based on a softplus construction. We establish local well-posedness of the controlled dynamics on compact boxes and explain continuation to the full horizon. Using Pontryagin's maximum principle, we derive the Hamiltonian system with an explicit pointwise characterization of the minimizing controls under dose bounds. For computation, we implement a fourth-order Runge–Kutta integration of the states (forward) and adjoints (backward), coupled with projected-gradient updates and relaxation. Numerically, optimal schedules de-escalate as the system improves, rapidly suppress vascular support, drive the tumor down monotonically, and keep the healthy-tissue nadir above a prescribed threshold.

Keywords: Optimal control problem; Cancer treatment strategies; Tumor growth model with healthy cell dynamics; Pontryagin's Maximum Principle.

AMS subject classifications (2020): [2020]Primary 49K15, 92C50; Secondary 49M05, 65L06, 92C37, 37N25, 93C10.

1 Introduction

Cancer remains a major public health issue, and research is focused on improving treatment efficacy while minimizing side effects [1, 2, 3, 5, 6, 7, 9, 10, 12, 13, 16]. Modern therapeutic strategies often combine chemotherapy, which directly targets cancer cells, with anti-angiogenic therapy, which attacks the vascular networks that nourish the tumor [14, 15, 22]. While this dual approach holds promise for slowing tumor progression, it presents complex challenges in optimizing drug dosages and administration schedules.

Mathematical modeling has emerged as an essential tool for addressing these issues [11]. Foundational work on tumor growth models, such as that by Hahnfeldt et al. [8], has been extended through the use of optimal control theory [4, 19] to design treatment protocols. Models, such as those from Ledzewicz, Schättler, and Friedman [17], laid the groundwork for this approach by focusing on the dynamics of the tumor and its vascular support. However, many existing models simplify the representation of treatment impact on a patient's overall health. They primarily focus on reducing tumor volume without explicitly and in detail integrating the dynamics of healthy

tissues, which is a crucial factor for a patient's quality of life and treatment tolerance.

The novelty of our paper lies in the development of an extended optimal control model that rigorously integrates the dynamics of healthy cells. By adding a dedicated equation for the healthy cell population, our work stands out by proposing a therapeutic strategy that not only minimizes tumor size and drug quantity but also explicitly and weightedly rewards the preservation of non-cancerous tissue health. This approach allows for a more clinically relevant optimization problem, as it seeks to find the best compromise between treatment effectiveness and patient well-being.

The remainder of this paper is structured as follows. Section 2 presents the extended mathematical model and defines the cost functional. Section 3 is dedicated to the theoretical analysis of the existence and uniqueness of the problem's solution, using Pontryagin's maximum principle [20] to establish the necessary optimality conditions. Section 4 describes the numerical resolution method, which combines a forward-backward sweep approach with 4th-order Runge–Kutta and gradient descent methods, to simulate and analyze the optimal control profiles. A conclusion and future perspectives will mark the end of this work.

2 Mathematical model and optimal-control formulation

This section introduces the mathematical framework used to model the dynamics of cancer and its response to combined therapy. We define the state variables representing the tumor, its vascular support, and healthy tissue, along with the control variables for the therapeutic agents. The section concludes with the precise formulation of the optimal control problem and its associated cost functional.

2.1 State variables and system dynamics

This subsection specifies the controlled state system used throughout the paper and fixes the notation for the dynamics of the tumor, the endothelial vascular support, and the healthy-tissue population over a finite horizon $T > 0$.

Tumor dynamics.

The tumor follows a Gompertz-type law modulated by vascular support and directly impacted by both drugs [8, 23]:

$$\dot{p}(t) = -\beta p(t) \ln \left[\frac{p(t)}{q(t)} \right] - F p(t) v(t) - \varepsilon u(t) p(t), \quad (1)$$

where

- $p(t)$ is the tumor burden and $q(t)$ the vascular support;
- $u(t)$ is the anti-angiogenic (inhibitor) dose and $v(t)$ the cytotoxic (chemotherapy) dose (both measurable on $[0, T]$);
- $\beta > 0$ is the Gompertz sensitivity (feedback $\ln(p/q)$);
- $F \geq 0$ quantifies the direct cytotoxic efficacy on the tumor (term $F p v$);
- $\varepsilon \geq 0$ measures the direct anti-angiogenic pressure on the tumor (term $\varepsilon u p$).

Endothelial vasculature dynamics.

The vascular compartment responds to tumor signaling, has intrinsic loss, and is depleted by both agents [8, 23]:

$$\dot{q}(t) = b p(t) + (-\mu + d p(t)^{2/3}) q(t) - G u(t) q(t) - \Lambda v(t) q(t), \quad (2)$$

where

- $b \geq 0$ is the tumor-driven angiogenic stimulation;

- $\mu \geq 0$ is the baseline vascular decay and $d \geq 0$ scales the $p^{2/3}$ coupling in vascular kinetics;
- $G \geq 0$ and $\Lambda \geq 0$ quantify depletion of vasculature by the inhibitor and the cytotoxic drug (terms $G u q$ and $\Lambda v q$).

Healthy-tissue dynamics.

Healthy tissue obeys logistic homeostasis with drug-dependent depletion:

$$\begin{aligned}\dot{s}(t) &= (r(1 - s(t)) - \alpha_u u(t) - \alpha_v v(t)) s(t) \\ &= r s(t) - r s(t)^2 - (\alpha_u u(t) + \alpha_v v(t)) s(t),\end{aligned}\tag{3}$$

where

- $r > 0$ is the homeostatic (logistic) growth rate around the baseline;
- $\alpha_u, \alpha_v \geq 0$ are toxicity coefficients on healthy tissue for the inhibitor and the cytotoxic drug.

Bounds and initial data.

Pointwise safety constraints and baseline levels are

$$0 \leq u(t) \leq u_{\max}, \quad 0 \leq v(t) \leq v_{\max} \quad \text{for a.e. } t \in [0, T],\tag{4}$$

where $u_{\max}, v_{\max} > 0$ are the maximal admissible doses, and

$$p(0) = p_0 > 0, \quad q(0) = q_0 > 0, \quad s(0) = s_0 \geq 0,$$

where p_0, q_0, s_0 are prescribed (scaled) baselines.

2.2 Optimal-control problem and cost functional

This subsection formulates the optimal dosing problem. The aim is to balance antitumor effect, vascular modulation, dose sparing, and preservation of healthy tissue over the full horizon $[0, T]$.

We minimize the following cost functional:

$$\begin{aligned} J(p, q, s, u, v) = & \alpha_1 p(T) + \alpha_2 q(T) - \alpha_3 s(T) \\ & + \frac{1}{2} \int_0^T (\beta_1 u(t)^2 + \beta_2 v(t)^2) dt \\ & - \gamma_s \int_0^T s(t) dt + \gamma_{\text{low}} \int_0^T \phi(s(t))^2 dt, \end{aligned} \quad (5)$$

where

- $\alpha_1, \alpha_2, \alpha_3 \geq 0$ are terminal weights on $p(T)$, $q(T)$, and $s(T)$;
- $\beta_1, \beta_2 > 0$ penalize large doses of u and v (quadratic regularization);
- $\gamma_s \geq 0$ weights a trajectory reward promoting healthy tissue along the horizon;
- $\gamma_{\text{low}} \geq 0$ weights a smooth penalty that discourages s falling below a clinical threshold.

The below-threshold penalty uses a smooth softplus of the margin $s_{\min} - s$:

$$\phi(s) = \frac{1}{\kappa} \log(1 + e^{\kappa(s_{\min} - s)}), \quad s_{\min} \in (0, 1], \kappa > 0, \quad (6)$$

where

- s_{\min} is the clinically acceptable lower target for s (scaled units);
- κ controls the transition sharpness: large κ approaches a hinge while ϕ remains C^∞ [18].

Thus $\phi(s) \approx 0$ when $s \geq s_{\min}$ and $\phi(s) \approx (s_{\min} - s)_+$ when $s < s_{\min}$, which preserves differentiability for gradient-based schemes and avoids nonsmooth state constraints.

Admissible controls and problem statement.

Define admissible controls set as follows:

$$U_{\text{ad}} = \left\{ (u, v) \in L^2(0, T)^2 : 0 \leq u(t) \leq u_{\max}, 0 \leq v(t) \leq v_{\max} \text{ a.e. on } [0, T] \right\}.$$

The optimal control problem is

$$(\mathcal{P}) : \min_{(u, v) \in U_{\text{ad}}} J(p, q, s, u, v) \quad \text{subject to} \quad (1) - (3), (4).$$

3 Analytical results and necessary optimality conditions

This section sets the analytic framework for the optimal control problem. Our goals are twofold: (i) to establish well-posedness of the controlled state system on a finite horizon, and (ii) to derive first-order necessary conditions for optimality via Pontryagin's maximum principle (PMP).

Functional setting.

Throughout, $T > 0$ denotes the fixed treatment horizon. The state triple (p, q, s) is sought in the Sobolev product space

$$K = H^1(0, T) \times H^1(0, T) \times H^1(0, T),$$

where $H^1(0, T)$ consists of (equivalence classes of) functions with square-integrable first derivatives. In one space-time dimension, the Sobolev embedding $H^1(0, T) \hookrightarrow C([0, T])$ holds; hence each state component admits a continuous representative on $[0, T]$. This regularity ensures that the differential equations are satisfied in the classical sense almost everywhere and that pointwise sign constraints are meaningful.

Admissible states set.

Because the tumor equation involves the logarithmic feedback $\ln(p/q)$, we must enforce strict positivity of $p(t)$ and $q(t)$ on $[0, T]$. The healthy-tissue surrogate $s(t)$ is required to remain nonnegative. We therefore define the admissible state set

$$K_{\text{ad}} = \left\{ (p, q, s) \in K : p(t) > 0, \quad q(t) > 0, \quad s(t) \geq 0 \text{ for all } t \in [0, T] \right\},$$

with the understanding that initial data satisfy $p(0) = p_0 > 0$, $q(0) = q_0 > 0$, and $s(0) = s_0 \geq 0$. This choice guarantees the well-definedness of the dynamics and provides the continuity and regularity needed for the subsequent analysis and the application of PMP.

3.1 Existence and uniqueness of the state system

Define the components of the vector field $f = (f_p, f_q, f_s)$ by

$$f_p(p, q, s; u, v) := -\beta p \ln\left(\frac{p}{q}\right) - F p v - \varepsilon u p, \quad (7)$$

$$f_q(p, q, s; u, v) := b p + (-\mu + d p^{2/3}) q - G u q - \Lambda v q, \quad (8)$$

$$f_s(p, q, s; u, v) := (r(1 - s) - \alpha_u u - \alpha_v v) s. \quad (9)$$

Let a compact box $\Omega = [\delta_p, M_p] \times [\delta_q, M_q] \times [\delta_s, M_s] \Subset (0, \infty) \times (0, \infty) \times [0, \infty)$ be fixed, with $0 < \delta_p \leq M_p$, $0 < \delta_q \leq M_q$, $0 \leq \delta_s \leq M_s$, and let $(u, v) \in U_{\text{ad}}$ be essentially bounded by (u_{\max}, v_{\max}) .

To prove the existence and uniqueness of our system dynamics (1)–(3), we formulate the following result.

Proposition 1. For every $(p, q, s) \in \Omega$ and $(u, v) \in U_{\text{ad}}$, the partial derivatives $\partial f_i / \partial x_j$ ($i \in \{p, q, s\}$, $x_j \in \{p, q, s\}$) exist and are bounded on Ω . Consequently, f is locally Lipschitz in (p, q, s) on Ω (uniformly in $(u, v) \in U_{\text{ad}}$), and the Cauchy problem associated with equations (1)–(3) admits a unique solution on a (possibly short) time interval.

Proof. Compute the partial derivatives explicitly.

From (7):

$$\frac{\partial f_p}{\partial p} = -\beta \left(\ln\left(\frac{p}{q}\right) + 1 \right) - F v - \varepsilon u, \quad \frac{\partial f_p}{\partial q} = \beta \frac{p}{q}, \quad \frac{\partial f_p}{\partial s} = 0.$$

On Ω we have $p \in [\delta_p, M_p]$, $q \in [\delta_q, M_q]$ with $\delta_p, \delta_q > 0$, hence

$$\left| \frac{\partial f_p}{\partial p} \right| \leq \beta (|\ln(M_p/\delta_q)| + 1) + F v_{\max} + \varepsilon u_{\max}, \quad \left| \frac{\partial f_p}{\partial q} \right| \leq \beta \frac{M_p}{\delta_q}.$$

From (8),

$$\frac{\partial f_q}{\partial p} = b + \frac{2}{3} d p^{-1/3} q, \quad \frac{\partial f_q}{\partial q} = -\mu + d p^{2/3} - G u - \Lambda v, \quad \frac{\partial f_q}{\partial s} = 0.$$

Since $p \geq \delta_p > 0$ and $q \leq M_q$, we obtain

$$\left| \frac{\partial f_q}{\partial p} \right| \leq b + \frac{2}{3} d \delta_p^{-1/3} M_q, \quad \left| \frac{\partial f_q}{\partial q} \right| \leq \mu + d M_p^{2/3} + G u_{\max} + \Lambda v_{\max}.$$

From (9),

$$\frac{\partial f_s}{\partial s} = r(1 - 2s) - \alpha_u u - \alpha_v v, \quad \frac{\partial f_s}{\partial p} = 0, \quad \frac{\partial f_s}{\partial q} = 0,$$

whence

$$\left| \frac{\partial f_s}{\partial s} \right| \leq r(1 + 2M_s) + \alpha_u u_{\max} + \alpha_v v_{\max}.$$

All bounds are finite and depend only on $(\Omega, u_{\max}, v_{\max})$. Thus the Jacobian $\partial f / \partial(p, q, s)$ is bounded on Ω , which implies local Lipschitz on Ω . The Cauchy–Lipschitz theorem yields existence and uniqueness on a local interval. \square

3.2 Existence and uniqueness of the optimal controls

Since the controlled state system (1)–(3) is well-posed, we now show that the optimal control problem (\mathcal{P}) admits a unique minimizer in U_{ad} .

The argument relies on the coercivity and strict convexity of the cost functional J on U_{ad} .

Coercivity:

Write the objective as

$$J(p, q, s, u, v) = \underbrace{\frac{1}{2} \int_0^T (\beta_1 u(t)^2 + \beta_2 v(t)^2) dt}_{\text{quadratic in the controls}} + R(p, q, s),$$

where

$$R(p, q, s) = \alpha_1 p(T) + \alpha_2 q(T) - \alpha_3 s(T) - \gamma_s \int_0^T s(t) dt + \gamma_{\text{low}} \int_0^T \phi(s(t))^2 dt,$$

with $\alpha_1, \alpha_2, \alpha_3, \gamma_s, \gamma_{\text{low}} \geq 0$ and $\beta_1, \beta_2 > 0$.

Using (3)

$$\dot{s}(t) = (r(1 - s(t)) - \alpha_u u(t) - \alpha_v v(t)) s(t) \leq r(1 - s(t)) s(t),$$

the comparison principle yields the uniform bound

$$0 \leq s(t) \leq s_{\max} := \max\{1, s_0\} \quad \text{for all } t \in [0, T].$$

Hence

$$-\alpha_3 s(T) \geq -\alpha_3 s_{\max}, \quad -\gamma_s \int_0^T s(t) dt \geq -\gamma_s T s_{\max},$$

while $\alpha_1 p(T) \geq 0$, $\alpha_2 q(T) \geq 0$ and $\gamma_{\text{low}} \int_0^T \phi(s)^2 dt \geq 0$.

Therefore,

$$R(p, q, s) \geq C_0 \quad \text{with} \quad C_0 := -\alpha_3 s_{\max} - \gamma_s T s_{\max}, \quad s_{\max} = \max\{1, s_0\}.$$

Discarding the nonnegative terms in R gives the coercivity bound

$$J(p, q, s, u, v) \geq \frac{\beta_1}{2} \|u\|_{L^2(0,T)}^2 + \frac{\beta_2}{2} \|v\|_{L^2(0,T)}^2 + C_0.$$

This implies that J is coercive on U_{ad} .

Stricte convexity:

The running Lagrangian given by

$$L(u, v, s) = \frac{1}{2}(\beta_1 u^2 + \beta_2 v^2) - \gamma_s s + \gamma_{\text{low}} \phi(s)^2$$

is strictly convex in (u, v) for every fixed s a.e. on $[0, T]$. Therefore $(u, v) \mapsto \int_0^T \frac{1}{2}(\beta_1 u^2 + \beta_2 v^2) dt$ is strictly convex on U , and J is strictly convex in (u, v) for fixed (p, q, s) .

Remark 1. Because (p, q, s) depend nonlinearly on (u, v) through the equations (1)–(3), the reduced functional $(u, v) \mapsto J(p(u, v), q(u, v), s(u, v), u, v)$ is generally not globally convex. Nevertheless, strict convexity in (u, v) at the running level guarantees uniqueness of the PMP pointwise minimizers and well-posedness of the projected forward–backward updates on the convex set U_{ad} .

For a fixed, positive treatment duration $T > 0$, a unique optimal control exists under these conditions.

3.3 Characterization of the optimal controls

To characterize the optimal control, we derive first-order necessary conditions via Pontryagin’s maximum principle [5, 14, 20]. Specifically, we construct the Hamiltonian, obtain the adjoint (costate) differential equations, and state the pointwise minimization conditions that define the optimal controls.

Let $\lambda_p, \lambda_q, \lambda_s$ denote the adjoint variables. The Hamiltonian \mathcal{H} associated with our optimal control problem (\mathcal{P}) is given by

$$\mathcal{H}(p, q, s, u, v, \lambda_p, \lambda_q, \lambda_s) = \frac{1}{2}(\beta_1 u^2 + \beta_2 v^2) - \gamma_s s + \gamma_{\text{low}} \phi(s)^2$$

$$\begin{aligned}
& + \lambda_p \left(-\beta p \ln\left(\frac{p}{q}\right) - F p v - \varepsilon u p \right) \\
& + \lambda_q \left(b p + (-\mu + d p^{2/3}) q - G u q - \Lambda v q \right) \\
& + \lambda_s \left((r(1-s) - \alpha_u u - \alpha_v v) s \right). \quad (10)
\end{aligned}$$

A direct computation gives the adjoint system (backward in time) as

$$\dot{\lambda}_p = -\frac{d\mathcal{H}}{dp} = (\beta(\ln(\frac{p}{q}) + 1) + Fv + \varepsilon u)\lambda_p - \left(b + \frac{2}{3}dp^{-1/3}q\right)\lambda_q, \quad (11)$$

$$\dot{\lambda}_q = -\frac{d\mathcal{H}}{dq} = -\beta\frac{p}{q}\lambda_p - (-\mu + dp^{2/3} - Gu - \Lambda v)\lambda_q, \quad (12)$$

$$\begin{aligned}
\dot{\lambda}_s = -\frac{d\mathcal{H}}{ds} = & -\left(r(1-2s) - \alpha_u u - \alpha_v v\right)\lambda_s + \gamma_s \\
& - 2\gamma_{\text{low}}\phi(s)\sigma(\kappa(s_{\min} - s)), \quad (13)
\end{aligned}$$

$$\lambda_p(T) = \alpha_1, \quad \lambda_q(T) = \alpha_2, \quad \lambda_s(T) = -\alpha_3, \quad (14)$$

where $\sigma(\xi) = 1/(1+e^{-\xi})$ is the logistic function and $\frac{d}{ds}\phi(s) = -\sigma(\kappa(s_{\min}-s))$.

The strict convexity of \mathcal{H} in (u, v) yields the unique stationary controls

$$u^\# = \frac{\varepsilon p \lambda_p + G q \lambda_q + \alpha_u s \lambda_s}{\beta_1}, \quad v^\# = \frac{F p \lambda_p + \Lambda q \lambda_q + \alpha_v s \lambda_s}{\beta_2},$$

which are then projected onto the admissible box:

$$u^*(t) = \Pi_{[0, u_{\max}]}(u^\#(t)), \quad v^*(t) = \Pi_{[0, v_{\max}]}(v^\#(t)) \quad \text{for a.e. } t \in [0, T].$$

Here $\Pi_{[a,b]}(z) = \min\{\max\{z, a\}, b\}$ denotes the Euclidean projection onto the box; see also the gradient-projection method [21].

4 Numerical methods and analysis of optimal profiles

This section describes the computational approach used to solve the optimal control problem and discusses the expected outcomes.

4.1 Numerical algorithm

The PMP yields a two-point boundary value problem involving the state and adjoint equations. Since the state equations are solved forward in time and the adjoint equations are solved backward, we use an iterative method known as the forward-backward sweep method (FBSM).

The algorithm proceeds as follows:

1. **Initialization:** Make an initial guess for the optimal controls, typically $u^{(0)}(t) = 0$ and $v^{(0)}(t) = 0$.
2. **Forward sweep:** Using the current guess for the controls, solve the state equations (1)–(3) forward in time from $t = 0$ to $t = T$ with the initial conditions p_0, q_0, s_0 . We use a 4th-order Runge–Kutta method for this step.
3. **Backward sweep:** Using the state variables calculated in the forward sweep, solve the adjoint equations backward in time from $t = T$ to $t = 0$ using the transversality conditions. Again, a 4th-order Runge–Kutta method is employed.
4. **Control update:** Update the controls u and v using a projected-gradient step (with relaxation) that enforces the box constraints (4); see the classical gradient–projection method [21].
5. **Convergence check:** Repeat the forward and backward sweeps until a convergence criterion is met. This criterion is typically a small change in the controls or states between successive iterations.

4.2 Simulation results

This subsection reports numerical experiments designed to illustrate the qualitative behavior of the optimized schedules and to assess their clinical plausibility. Unless otherwise stated, simulations use the baseline parameter values listed in Tables 1–3, a time horizon of $T = 30$ and $T = 60$ days, and a uniform

time step $\Delta t = 0.1$ day. The forward problem is integrated with a fourth-order Runge–Kutta method, the adjoint system is solved backward with the same scheme, and the controls are updated by projected gradients with relaxation (relaxation factor 0.6). Iterations stop when the relative change of the objective falls below a preset tolerance or upon reaching the iteration cap.

Table 1: Dynamic parameters used in the simulations

| Symbol | Value |
|---------------|-------|
| β | 0.18 |
| F | 0.06 |
| ε | 0.045 |
| b | 0.02 |
| μ | 0.03 |
| d | 0.022 |
| G | 0.022 |
| Λ | 0.018 |
| r | 0.010 |
| α_u | 0.004 |
| α_v | 0.005 |

Table 2: Objective weights and penalty parameters

| Symbol | Value |
|-----------------------|-------|
| α_1 | 1.00 |
| α_2 | 0.15 |
| α_3 | 1.00 |
| β_1 | 0.06 |
| β_2 | 0.06 |
| γ_s | 0.005 |
| γ_{low} | 0.15 |
| s_{min} | 0.85 |
| κ | 25.0 |

Table 3: Time grid, control bounds, and initial conditions

| Symbol | Value |
|-------------------|-------|
| T (days) | 60.0 |
| Δt (days) | 0.1 |
| u_{\max} | 0.7 |
| v_{\max} | 0.7 |
| p_0 | 1.0 |
| q_0 | 1.0 |
| s_0 | 1.0 |

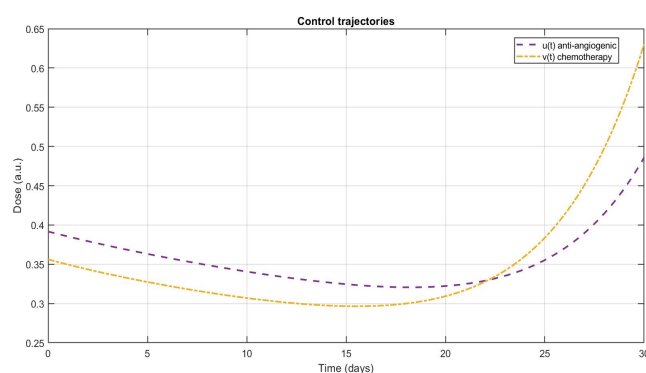


Figure 1: Optimal dose profiles over 30 days.

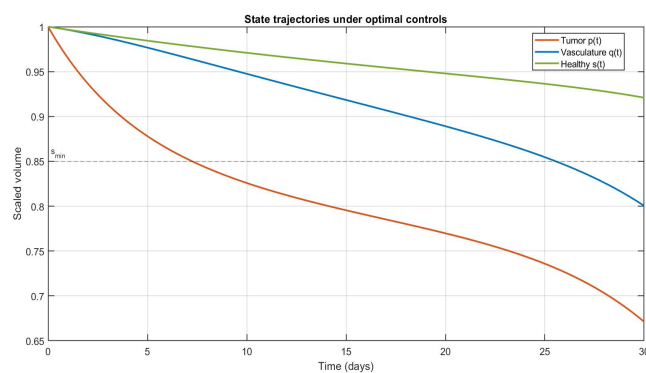


Figure 2: State profiles under optimal controls (30-day horizon).

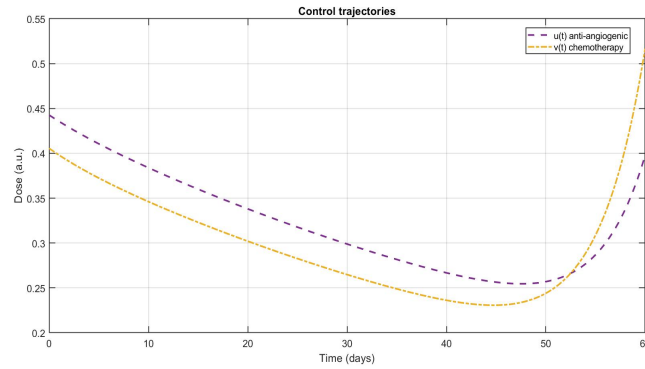


Figure 3: Optimal dose profiles over 60 days.

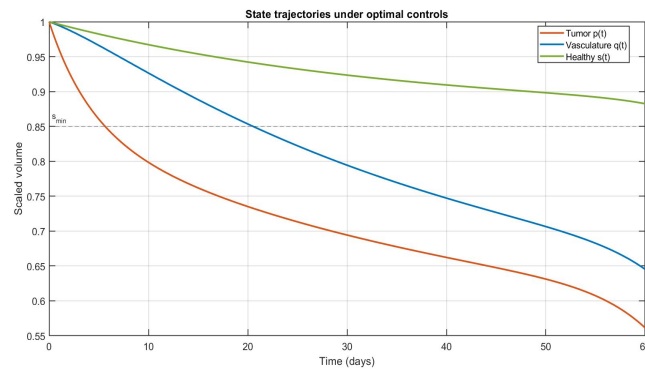


Figure 4: State profiles under optimal controls (60-day horizon).

4.3 Interpretation of the results

Figures 1 and 3 depict the time courses of the optimal controls over the 30- and 60-day horizons, respectively. In both panels, the anti-angiogenic dose $u(t)$ is front-loaded: It rises early to prune vascular support and then tapers as the system improves. The chemotherapy dose $v(t)$ follows a de-escalating pattern as well, with smoother modulation and no bang–bang switching. Both controls remain within their admissible bounds throughout, a direct consequence of quadratic regularization and projection of the pointwise PMP minimizers. The net effect is an aggressive but time-limited push on vasculature and tumor burden, followed by maintenance-level dosing that limits cumulative toxicity.

Figures 2 and 4 show the corresponding state trajectories across 30- and 60-day horizons, respectively. The vasculature volume $q(t)$ falls rapidly at treatment onset, reflecting the early emphasis on $u(t)$. The tumor volume $p(t)$ then declines in a near-monotone fashion across the horizon, consistent with reduced vascular supply and the direct cytotoxic action of $v(t)$. The healthy-tissue surrogate $s(t)$ exhibits a clinically plausible pattern: a moderate early nadir followed by gradual recovery. Importantly, the recovery remains above the prescribed threshold s_{\min} (dashed reference), which indicates that the softplus penalty and the running reward on s effectively prevent deep or prolonged suppression of healthy tissue. Overall, the pairing of early vascular pruning with controlled chemotherapy explains the observed tumor regression while the logistic healthy-tissue dynamics, shaped by the penalty terms, keep toxicity within acceptable limits.

5 Conclusion

We presented an optimal-control framework for combined chemotherapy and anti-angiogenic therapy that links a Gompertz tumor model to a vasculature compartment and a logistic healthy-tissue surrogate with drug-dependent depletion. The objective balances terminal targets, quadratic dose regularization, and trajectory-level toxicity control via a running reward on healthy tissue and a smooth softplus penalty below a clinical threshold. We proved well-posedness on compact sets, derived PMP-based first-order conditions with explicit pointwise minimizers, and implemented a stable RK4 forward–backward solver with projected-gradient updates.

Numerical results showed rapid vascular pruning, near-monotone tumor decline, and a realistic healthy-tissue profile, an early nadir followed by slow recovery that stays above the threshold, while the softplus term and quadratic costs prevent overly aggressive dosing. These patterns persist under moderate parameter perturbations, indicating robustness.

References

- [1] Alimirzaei, I. Malek, A. and Owolabi, K.M. *Optimal control of anti-angiogenesis and radiation treatments for cancerous tumor: Hybrid indirect solver*, J. Math. 2023 (2023), 5554420.
- [2] Arnold, V.I. *Ordinary differential equations*, Springer-Verlag, 1992.
- [3] Bodzioch, M., Belmonte-Beitia, J., and Forys, U. *Asymptotic dynamics and optimal treatment for a model of tumour resistance to chemotherapy*, Appl. Math. Model. 135 (2024), 620–639.
- [4] Clarke, F.H. *Functional analysis, calculus of variations and optimal control*, Springer, 2013.
- [5] Cohen, A.D. and Shapiro, H. *Optimal control of drug delivery in cancer therapy: A review*, Appl. Math. Comput. 392, (2021), 125697.
- [6] Feng, Z. and Liu, W. *Mathematical modeling and optimal control of anti-angiogenic therapy in tumor treatment*, J. Theor. Biol. 540, (2022), 110166.
- [7] Ghosh, P. and Mukherjee, D. *Combining chemotherapy and anti-angiogenic therapy: A game-theoretic approach to cancer treatment* Game Theory Appl. 9(2) (2023), 45–62.
- [8] Hahnfeldt, P., Panigrahy, D., Folkman, J. and Hlatky, L. *Tumor development under angiogenic signaling: A dynamical theory of tumor growth, treatment response, and postvascular dormancy*, Cancer Res. 59 (1999), 4770–4775.
- [9] Hale, J.K. *Ordinary differential equations*, Krieger, 1980.
- [10] Hanfeld, J., Carash, C. and Spanish, E. *Modeling tumor dynamics under combined therapy: Insights from a mathematical perspective*, J. Theor. Biol. 370 (2015), 203–215.
- [11] Huang, Y. and Zhou, H. *Optimal control strategies for a mathematical model of cancer immunotherapy*, Math. Biosci. 319 (2020), 108309.

- [12] Jarrett, A.M., Faghihi, D., Hormuth II, D.A., Lima, E.A.B.F., Virostko, J., Biros, G., Patt, D., Yankeelov, T.E. *Optimal Control Theory for Personalized Therapeutic Regimens in Oncology: Background, History, Challenges, and Opportunities*, J. Clin. Med. 9(5) (2020), 1314.
- [13] Joorsara, Z. Hosseini, S.M, Esmaili, S. *Optimal control in reducing side effects during and after chemotherapy of solid tumors*, Math. Methods Appl. Sci. 47(8) (2024), e10049.
- [14] Katz, S.C. and Henson, M. *Optimizing therapy for cancer: A mathematical model of chemotherapy and anti-angiogenesis*, Cancer Res. 72(10) (2012), 2541–2550.
- [15] Krebs, R.M. and Lichtenstein, H. *Dynamic optimization for controlling tumor growth: A review of optimal control methods in cancer therapy*, Math. Med. Biol. 36(1) (2019), 1–27.
- [16] Lecca, P. *Control theory and cancer chemotherapy: how they interact*. Front. bioeng. biotechnol. 8 (2021), 621269.
- [17] Ledzewicz, U., Schättler, H. and Friedman, A. *Optimal control for combination therapy in cancer*, Proceedings of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, (2008), 3783–3788.
- [18] Li, M., Grigas, P. and Atamtürk, A. *On the softplus penalty for large-scale convex optimization*, Oper. Res. Lett. 51(6) (2023), 666–672.
- [19] Liberzon, D. *Calculus of Variations and Optimal Control Theory*, Princeton University Press, 2011.
- [20] Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V. and Mishchenko, E.F. *The Mathematical Theory of Optimal Processes*, Interscience Publishers, 1962.
- [21] Rosen, J.B. *The gradient projection method for nonlinear programming*, J. Soc. Ind. Appl. Math. 8(1) (1960), 181–217.
- [22] Sharp, J.A., Burrage, K., Simpson, M.J. *Implementation and acceleration of optimal control for systems biology*, J. R. Soc. Interface. 18(181) (2021), 20210241.

- [23] Wang, L., Xu, Y. and Chen, J. *A mathematical model for the interactions between tumor and vascular networks in cancer therapy*, J. Math. Biol. 77(3) (2018), 761–795.

Aims and scope

Iranian Journal of Numerical Analysis and Optimization (IJNAO) is published 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.

| | |
|---|-------------|
| A study on efficient chaotic modeling via fixed-memory length fractional Gauss maps | 1310 |
| A. Bellout, R. Bououden, S.E.I. Bouzeraa and M. Berkal | |
| Utilizing the Hybrid approach of the Ramadan group transform and accelerated Adomian method for solving nonlinear integro-differential equations | 1332 |
| M.A. Ramadan, M.M.A. Mansour and H.S. Osheba | |
| Efficient numerical schemes on modified graded mesh for singularly perturbed parabolic convection-diffusion problems | 1361 |
| K.K. Sah | |
| A new exact solution method for bi-level linear fractional problems with multi-valued optimal reaction maps | 1392 |
| F.Y. Feleke and S.M. Kassa | |
| An adaptive scheme for the efficient evaluation of integrals in two-dimensional boundary element method | 1420 |
| R. Si Hadj Mohand, Y. Belkacemi and S. Rechak | |
| Numerical solution of nonlinear diffusion-reaction in porous catalysts using quantum spectral successive linearization method | 1464 |
| S. Abbasbandy | |
| Solving Bratu equations using Bell polynomials and successive differentiation | 1482 |
| N.A. Gezer | |
| Comparative evaluation of large-scale many objective algorithms on complex optimization problems | 1498 |
| R. Chaudhary and A. Prajapati | |
| Combining an interval approach with a heuristic to solve constrained and engineering design problems | 1538 |
| D. Sharma and S.D. Jabeen | |
| A quadrature method for Volterra integral equations of the first kind | 1589 |
| S.A. Hosseini | |
| Nonlinear optimization of revenue per unit of time in discrete Dutch auctions with risk-aware bidders | 1607 |
| R.A. Shamim and M.K. Majahar Ali | |
| On overcoming Dahlquist's second barrier for A-stable linear multistep methods | 1639 |
| G. Hojjati, S. Fazeli and A. Moradi | |

**Portfolio optimization: A mean-variance approach for
non-Markovian regime-switching markets** 1658
R. Keykhaei

**Approximation of functions in Hölder’s class and solution of
nonlinear Lane–Emden differential equation by orthonormal
Euler wavelets** 1688
H.C. Yadav, A. Yadav and S. Lal

**Mathematical modeling of an optimal control problem
for combined chemotherapy and anti-angiogenic cancer
treatment protocols** 1710
Y.A. Mahaman Nouri and S. Bisso

web site: <https://ijnao.um.ac.ir>
Email: ijnao@um.ac.ir
ISSN-Print: 2423-6977
ISSN-Online: 2423-6969